A semi-parametric Bayesian approach, iSBA, for differential expression analysis of RNA-seq data

Ran Bi [1], Peng Liu [1*]

1 Department of Statistics, Iowa State University, Ames, Iowa, USA

* pliu@iastate.edu

Abstract

RNA sequencing (RNA-seq) technologies have been popularly applied to study gene expression in recent years. Identifying differentially expressed (DE) genes across treatments is one of the major steps in RNA-seq data analysis. Most differential expression analysis methods rely on parametric assumptions, and it is not guaranteed that these assumptions are appropriate for real data analysis. In this paper, we develop a semi-parametric Bayesian approach for differential expression analysis. More specifically, we model the RNA-seq count data with a Poisson-Gamma mixture model, and propose a Bayesian mixture modeling procedure with a Dirichlet process as the prior model for the distribution of fold changes between the two treatment means. We develop Markov chain Monte Carlo (MCMC) posterior simulation using Metropolis Hastings algorithm to generate posterior samples for differential expression analysis while controlling false discovery rate. Simulation results demonstrate that our proposed method outperforms other popular methods used for detecting DE genes.

Introduction

During the past decade, RNA sequencing (RNA-seq) technologies have revolutionized transcriptomic studies. In a typical RNA-seq experiment, messenger RNA (mRNA) molecules are extracted from samples, fragmented, and converted to a library of complementary DNA (cDNA) fragments. The cDNA fragments are then amplified and sequenced on a high-throughput platform, such as HiSeq by Illumina or SOLiD by Applied Biosystems. Millions of DNA fragment sequences, called reads, are obtained for each sample and mapped to a reference genome. The number of reads aligned to a given gene measures the expression level for that gene. Thus, RNA-seq generates discrete count data rather than continuous data serving as measurements of mRNA expression levels.

In the statistical analysis of RNA-seq data, detecting differentially expressed (DE) genes across treatments or conditions is one of the major steps and often the main goal. A gene is considered to be DE if the expression levels change across treatment groups. Otherwise, the gene is said to be equivalently expressed (EE). Generally, negative binomial (NB) distribution is used for modeling RNA-seq count data. Many statistical methods based on the NB distribution have been proposed for detecting DE genes with RNA-seq data, including *edgeR* [1–4], *DESeq* [5] and *DESeq2* [6]. Methods that do not assume NB models typically involves transformation of the count data to continuous scale, such as the *Voom* and *limma* pipeline [7], which models the mean-variance relationship of the log-transformed count data and produces a precision weight for each

observation, then applies the *limma* method based on normal distributions [8] for the detection of DE genes.

The comparison among all the popular methods for RNA-seq data analysis mentioned above has been done through simulation studies [9, 10]. However, the optimality of these existing testing procedures is inadequately studied. Si and Liu (2013) [11] developed an optimal test for RNA-seq data anaysis while controlling FDR, where optimal tests were defined as tests that achieve the maximum of the power averaged across all genes for which null hypotheses are false. Furthermore, Si and Liu (2013) [11] proposed an approximation to the optimal test, where hyper distributions were estimated with mixture distributions, and such a test is called the approximated most average powerful (AMAP) test. In the two-treatment comparison problem, Si and Liu (2013) [11] modeled the gene-specific treatment means by the overall geometric mean expression level across both treatments and the ratio of the two treatment means, i.e., fold change ρ_g. They used a K-component mixture Gamma-Normal (MGN) distribution to model the joint distribution of the overall geometric mean expression level and the logarithm of the fold change. However, there are several limitations of using MGN distribution, such as difficulty in selecting an appropriate number of components K, and challenges in modeling the empirical distribution of all genes by parametric models.

Bayesian nonparametric modeling is a more flexible way for distribution estimation and is often applied to avoid critical dependence on parametric assumptions. The most popular Bayesian nonparametric methods adopt Dirichlet process (DP) mixture modeling, and such modeling framework has been utilized for DE analyses. For instance, [12] chose DP mixtures to model the population of genes under two different conditions and applied to a microarray dataset. Liu *et al.* (2015) [13] used the DP prior for modeling the distribution of fold changes between two treatments, with a mixture of a point mass at one and a Gamma distribution as the base distribution in the DP prior. In the method proposed by Liu *et al.* (2015) [13], one treatment condition was set as the reference condition (i.e., baseline) and they used DP as the prior for the distribution of fold changes of the other condition versus the reference. When they changed the reference treatment group, the declared differential expression status were not exactly the same for all genes.

To address this issue that the model is not invariant to the choice of reference condition, we propose a method using a mixture of three components as the base distribution in the DP prior for the distribution of the fold changes between two treatment conditions. The three components are a point mass at one, a Gamma, and an inverse-Gamma distribution, so that the model becomes invariant no matter which treatment group is set to be the reference. In addition, we model RNA-seq count data via a Poisson-Gamma mixture model, which is equivalent to a NB model. Similar to Liu *et al.* (2015) [13], this paper shows how our mixture modeling procedure can be accommodated to provide meaningful posterior probabilities of simple or composite null hypothesis. Also, we show that the posterior inference can be viewed as an approximation for the optimal test in Si and Liu (2013) [11], thus our approach is an approximated optimal test.

The article is organized as follows. In the Methods section, we describe our proposed Bayesian mixture modeling pipeline and the prior models, then present the MCMC sampling scheme for posterior inference and FDR estimation. In the Results section, we generate several simulation studies based on NB distributions, and compare our proposed method to some popular methods for DE analysis. We also analyze a real dataset using our proposed method. The Discussion section summarizes our results and provides some discussion.

Methods

In this section, we first describe the framework of our mixture modeling, and then introduce the prior models employed in our method.

A Poisson-Gamma Mixture Model

Suppose that an RNA-seq experiment measures G genes. Let Y_{gij} denote the number of reads mapped to gene g from biological replicate j of treatment i, where $g = 1, \ldots, G$, $i = 1, 2$, $j = 1, \ldots, n_i$, and n_i is the number of biological replicates in treatment i. As we mentioned in the introduction section, NB distribution has been popularly applied to such data. In the development of our modeling framework, we use a Poisson-Gamma mixture model parameterization instead of the NB model directly, where the RNA-seq read counts follow a Poisson distribution conditioning on the true expression mean, and the true gene abundances follow a Gamma distribution between replicate RNA samples. Then read count data Y_{gij} can be modeled as below,

$$\begin{aligned} Y_{gij}|\lambda_{gij} &\sim \text{Poisson}(S_{ij}\lambda_{gij}), \\ \lambda_{g1j}|\alpha_g, \beta_g &\sim \text{Gamma}(\alpha_g, \beta_g), \text{ and} \\ \lambda_{g2j}|\alpha_g, \beta_g, \rho_g &\sim \text{Gamma}(\alpha_g, \beta_g\rho_g), \end{aligned} \tag{1}$$

where S_{ij} is a normalization factor that accounts for sequencing depth variation and nuisance technical effects across the replicates, λ_{gij} is the normalized expression mean of jth replicate of ith treatment in gene g, α_g is the shape parameter which stands for the reciprocal of the dispersion parameter for gene g, β_g is the rate parameter for the first treatment, and the product of β_g and ρ_g is the rate parameter for the second treatment. So the marginal expression mean for treatment 1 is α_g/β_g, while for treatment 2 is $\alpha_g/(\beta_g\rho_{gi})$. Therefore, the mean ratio of treatment 1 over treatment 2 is ρ_g, which refers to the fold change between treatment 1 versus treatment 2.

The goal of differential expression analysis is to test

$$H_0^g : \rho_g \in \Delta_0 \quad \text{vs.} \quad H_1^g : \rho_g \in \Delta_1, \tag{2}$$

for each gene g, where Δ_0 represents the null set of values for ρ_g, while Δ_1 represents the alternative set. Δ_0 and Δ_1 are assumed to be a partition of the positive real line $\mathbb{R}^+$ ($\Delta_0 \bigcup \Delta_1 = \mathbb{R}^+$, $\Delta_0 \bigcap \Delta_1 = \varnothing$). The null space Δ_0 can be defined in different ways depending on the biological problems of interest. For example, if we are interested in identifying DE genes across the two treatments, we set $\Delta_0 = \{1\}$. If we are interested in whether the mean expression level in the first treatment is greater than the second treatment, we set $\Delta_0 = (0, 1]$. If we are interested in genes whose expression changes are large enough, for instance, the fold changes are greater than 1.5 [14], we set $\Delta_0 = [1/1.5, 1.5]$.

Prior Specification

Since our main focus is to test the hypothesis about the fold change parameter ρ_g in (2) for each gene, specifying an appropriate prior distribution for ρ_g is very crucial. The empirical distribution of the fold change of all genes could be very irregular and differs between various studies. To provide maximal flexibility, Bayesian nonparametric modeling with DP is a common way for distribution estimation. DP is a stochastic process whose realizations are probability distributions, i.e., each draw from a DP is itself a distribution. The formal definition of DP is as follows. Given a measurable set Ω, a base probability distribution F_0 and a positive real number M called the

concentration parameter, a random probability distribution F is generated by a DP if for any measurable partition $A_1, \ldots, A_k$ of Ω, the distribution of $(F(A_1), \ldots, F(A_k))$ is Dirichlet $D(M \cdot F_0(A_1), \ldots, M \cdot F_0(A_k))$. We denote this by $F \sim DP(M, F_0)$. The parameters F_0 and M play intuitive roles in the definition of the DP. For any measurable subset B of Ω, the base distribution F_0 is the mean of the DP, i.e., $E[F(B)] = F_0(B)$. Besides, the concentration parameter M defines the variance as $Var[F(B)] = F_0(B)(1 - F_0(B))/(M + 1)$. The larger M is, the smaller the variance, and the DP will concentrate more of its mass around the mean.

Throughout our mixture modeling procedure, we use a DP to model the fold change parameters $(\rho_1, \ldots, \rho_G)$. Different from Liu *et al.* (2015) [13], we use a mixture of a point mass at one, a Gamma and an inverse-Gamma distribution as the base distribution in the DP prior for the distribution of the fold change parameters, so that our modeling is invariant to the specification of the reference condition, and we call it *iSBA* (where *SBA* stands for semiparametric Bayesian approach). Details of the proof of reference level invariance are provided in S1 Appendix.

Therefore, the DP prior for gene g, $g = 1, \ldots, G$, can be expressed as

$$\begin{aligned} \rho_g | F &\overset{i.i.d.}{\sim} F, \\ F &\sim DP(M, F_0), \\ F_0 &\sim p_0 \delta_{\{1\}} + \frac{1}{2}(1 - p_0)\text{Gamma}(\alpha_0, \beta_0) \\ &\quad + \frac{1}{2}(1 - p_0)\text{Inv-Gamma}(\alpha_0, \beta_0), \end{aligned} \tag{3}$$

where p_0 is the proportion of EE genes, and $\delta_{\{x\}}$ denotes a point mass at x. In this paper, we set $p_0 = 0.5$ to give no prior preference to either DE or EE. The concentration parameter M in the DP priors is fixed as $M = 1$, which is a common choice used in application [12, 15, 16]. Throughout our paper, the simple null hypothesis of our great interest is $H_0^g : \rho_g = 1$.

Following Liu *et al.* (2015) [13], we use a Gamma distribution as the prior distribution for β_g due to its conjugacy, and an exponential distribution as the prior for α_g in order to reduce the computational complexity of the posterior distribution,

$$\alpha_g \sim \text{Exp}(r), \tag{4}$$

$$\beta_g \sim \text{Gamma}(a_0, b_0), \tag{5}$$

where r, a_0, b_0, in addition to α_0 and β_0 in (3), are hyperparameters. We set $r = 0.01$, $a_0 = 0.1$, $b_0 = 0.1$, $\alpha_0 = 0.1$, $\beta_0 = 0.1$ so that the priors are non-informative and the inference for α_g and β_g mainly relies on the observed data. For computational simplicity, we set the priors for α_g's, β_g's, and ρ_g's to be independent.

Markov Chain Monte Carlo Simulation

Posterior inference based on our proposed model is implemented by using Markov chain Monte Carlo (MCMC) algorithm [17]. MCMC methods are usually employed to generate samples from the posterior distribution by constructing a Markov chain that has the target posterior distribution as its equilibrium distribution. We use an MCMC-based sampling method in our proposed Bayesian mixture models. Gibbs sampling is the most frequently used tool to perform MCMC algorithm for Bayesian hierarchical models when dealing with conjugate priors. However, for addressing non-conjugate priors, the simplest way is by using the Metropolis-Hastings algorithm [18].

The Metropolis-Hastings algorithm simulates samples from a target distribution $\pi(x)$ using a proposal distribution $g(x^*|x)$, and updates the state x as follows. Generate a

candidate state x^* from the distribution $g(x^*|x)$, then compute the acceptance probability

$$a(x^*|x) = min\left[1, \frac{g(x|x^*)\pi(x^*)}{g(x^*|x)\pi(x)}\right].$$

Set the new state x' to x^* with probability $a(x^*|x)$. Otherwise, reject the candidate x^* and let x' be the same as x.

To simplify the use of DP prior, when F is integrated over its prior distribution (3), the sequence of ρ_g's follows a Polya urn scheme [19, 20], that is,

$$\rho_g|\boldsymbol{\rho}_{-g} \sim \frac{1}{G-1+M}\sum_{k\neq g}\delta_{\{\rho_k\}} + \frac{M}{G-1+M}F_0, \tag{6}$$

where $\boldsymbol{\rho}_{-g}$ is the vector of $(\rho_1, \ldots, \rho_G)$ after deleting ρ_g.

Then the most direct approach to sample for our model is to perform Metropolis-Hastings update for each of the ρ_g. However, this algorithm may not be very efficient since it cannot change the ρ_g for more than one gene simultaneously. A change to the ρ_g values occurs only when they are reallocated to new components. Thus it may take long time to converge to the posterior distribution [21]. In order to improve the efficiency of the MCMC algorithm, a modified Metropolis-Hastings updates and partial Gibbs sampling method has been proposed by Neal (2000) [21] (Algorithm 7). Suppose K is the number of distinct values in the vector $(\rho_1, \ldots, \rho_G)$ and the distinct values are denoted as $\rho_1^*, \ldots, \rho_K^*$, respectively. Let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_G)$ be the configuration indicators defined by

$$\xi_g = k \quad \text{if and only if} \quad \rho_g = \rho_k^* = \rho_{\xi_g}^*.$$

Therefore, we reparameterize the prior model for ρ_g's with ρ_k^*'s and ξ_g's as follows,

$$\begin{aligned}
\rho_k^* &\overset{i.i.d.}{\sim} F_0, \\
F_0 &\sim p_0\delta_{\{1\}} + \frac{1}{2}(1-p_0)\text{Gamma}(\alpha_0, \beta_0) \\
&\quad + \frac{1}{2}(1-p_0)\text{Inv-Gamma}(\alpha_0, \beta_0), \\
(\xi_1, \ldots, \xi_G)|M &\sim \text{CRP}(M),
\end{aligned}$$

where the prior models for ρ_k^*'s and ξ_g's are independent and CRP stands for Chinese Restaurant Process. CRP is a random distribution and the full conditional distribution for ξ_g's can be written as

$$\xi_g|\xi_l, M \sim \sum_{k=1}^{K^{(-g)}}\frac{n_k^{(-g)}}{G-1+M}\delta_{\{k\}} + \frac{M}{G-1+M}\delta_{\{K^{(-g)}+1\}},$$

where $K^{(-g)}$ denotes the number of distinct values in the vector $(\rho_1, \ldots, \rho_G)$ after deleting ρ_g, and $n_k^{(-g)}$ denotes the number of $(\rho_1, \ldots, \rho_G)$ who equal ρ_k^* after deleting ρ_g.

The MCMC sampling scheme uses the modified Metropolis-Hastings updates and partial Gibbs sampling method to repeatedly sample the following parameters step by step. The procedure for generating the full conditionals of all parameters and how we apply Metropolis-Hastings algorithm are shown in S2 Appendix.

(1) Draw samples of λ_{gij}'s from their full condition distributions,

$$\begin{aligned}
\lambda_{g1j}|\cdot &\sim \text{Gamma}(Y_{g1j} + \alpha_g, S_{1j} + \beta_g), \\
\lambda_{g2j}|\cdot &\sim \text{Gamma}(Y_{g2j} + \alpha_g, S_{2j} + \beta_g\rho_g).
\end{aligned}$$

(2) Draw samples of β_g's from their full conditional distributions,

$$\beta_g|\cdot \sim \text{Gamma}\Big(\alpha_g(n_1+n_2)+a_0,\ \sum_{j=1}^{n_1}\lambda_{g1j}+\sum_{j=1}^{n_2}\lambda_{g2j}\rho_g+b_0\Big).$$

(3) There is no closed-form full conditional distribution for α_g's. Since the conditional posterior distribution for each gene g is a log-concave function with respect to α_g, we could draw posterior samples based on adaptive rejection sampling method [22].

(4) Obtain posterior samples for ρ_g's by getting the Markov chain for $(\xi_1,\ldots,\xi_G)$ and $(\rho_1^*,\ldots,\rho_K^*)$ as follows:

(i) Update the configuration vector $(\xi_1,\ldots,\xi_G)$.

- For $g=1,\ldots,G$, repeat the following: If $\xi_g=\xi_l$ for some $l\neq g$, let ξ_g^* be a newly created component, with $\rho^*_{\xi_g^*}$ drawn from F_0. Set ξ_g to ξ_g^* with probability

$$a(\xi_g^*,\xi_g)=min\Big[1,\frac{M}{G-1}\cdot e^{-\beta_g\sum_{i=1}^{n_2}\lambda_{g2j}(\rho^*_{\xi_g^*}-\rho^*_{\xi_g})}\Big(\frac{\rho^*_{\xi_g^*}}{\rho^*_{\xi_g}}\Big)^{n_2\alpha_g}\Big].$$

Otherwise, if $\xi_g\neq\xi_l$ for all $l\neq g$, draw ξ_g^* from $\boldsymbol{\xi}_{-g}$, choosing $\xi_g^*=\xi$ with probability $\frac{n_\xi^{(-g)}}{G-1}$. Set the new ξ_g to this ξ_g^* with probability

$$a(\xi_g^*,\xi_g)=min\Big[1,\frac{G-1}{M}\cdot e^{-\beta_g\sum_{j=2}^{n_2}\lambda_{g2j}(\rho^*_{\xi_g^*}-\rho^*_{\xi_g})}\Big(\frac{\rho^*_{\xi_g^*}}{\rho^*_{\xi_g}}\Big)^{n_2\alpha_g}\Big].$$

- For $g=1,\ldots,G$, if $\xi_g\neq\xi_l$ for all $l\neq g$, do nothing. Otherwise, choose a new value for ξ_g from $\{\xi_1,\ldots,\xi_G\}$ with probabilities

$$p(\xi_g=\xi|\boldsymbol{\xi}_{-g},\text{rest})=b\cdot\frac{n_\xi^{(-g)}}{G-1}\prod_{j=1}^{n_2}\frac{\lambda_{g2j}^{\alpha_g-1}e^{-\beta_g\rho_\xi^*\lambda_{g2j}}(\beta_g\rho_\xi^*)^{\alpha_g}}{\Gamma(\alpha_g)},$$

where b is the appropriate normalizing constant.

(ii) Update $(\rho_1^*,\ldots,\rho_K^*)$. For $k=1,\ldots,K$, repeat the following: Draw ρ_k^{**} from F_0. Set the new value of ρ_k^* to ρ_k^{**} with the probability

$$a(\rho_k^{**},\rho_k^*)=min\left[1,e^{\sum_{\{g:\xi_g=k\}}\sum_{j=1}^{n_2}\beta_g\lambda_{g2j}(\rho_k^{**}-\rho_k^*)}\cdot\Big(\frac{\rho_k^{**}}{\rho_k^*}\Big)^{\sum_{\{g:\xi_g=k\}}n_2\alpha_g}\right].$$

Otherwise, let the new ρ_k^* be the same as the old value. If we have duplicated ρ_k^*, delete it and combine ξ_g.

Bayesian FDR Control

In genomic studies, tens of thousands of hypotheses are simultaneously tested, each relating to a gene. Thus multiple testing procedures that control the number of false significant results are commonly used in the analysis. False discovery rate (FDR) [23], defined as the expected proportion of false positives among the rejected hypotheses, has been the common choice of error criterion in RNA-seq data analysis. Within the Bayesian framework, one can estimate the FDR with Bayesian FDR [24, 25] by using posterior probability.

For each gene g, $g = 1, \ldots, G$, the posterior probability that gth null hypothesis is true is denoted by $P(\rho_g \in \Delta_0 | \boldsymbol{Y}_g)$. If we are interested in detecting DE genes, with $\Delta_0 = \{1\}$, $P(\rho_g \in \Delta_0 | \boldsymbol{Y}_g)$ is the posterior probability that gene g is EE. $P(\rho_g \in \Delta_0 | \boldsymbol{Y}_g)$ can be estimated by the proportion of the posterior samples obtained from MCMC for gene g that fall into the null set Δ_0, i.e.,

$$\hat{v}_g = \hat{P}(\rho_g \in \Delta_0 | \boldsymbol{Y}_g) = \frac{1}{N} \sum_{m=1}^{N} I(\rho_g^m \in \Delta_0 | \boldsymbol{Y}_g),$$

where N is the number of posterior samples. We reject H_0^g if the estimated posterior probability $\hat{v}_g$ is smaller than a critical value c^*. The critical value c^* is chosen based on controlling the FDR at a target level γ, for example, 0.05, i.e.,

$$c^* = sup\{c : \widehat{FDR}(c) < \gamma\},$$

where

$$\widehat{FDR}(c) = \frac{\sum_{g=1}^{G} \hat{v}_g I(\hat{v}_g < c)}{\sum_{g=1}^{G} I(\hat{v}_g < c)}.$$

So the Bayesian FDR controlled at level γ can be calculated by

$$\widehat{BFDR}(\gamma) = \frac{\sum_{g=1}^{G} \hat{v}_g I(\hat{v}_g < c^*)}{\sum_{g=1}^{G} I(\hat{v}_g < c^*)}.$$

Results

In this section, we adopt the simulation settings in Liu *et al.* (2015) [13] to assess the performance of our proposed method (*iSBA*), and compare to their semi-parametric Bayesian (*SBA*) method along with other popular methods for differential expression analysis of RNA-seq data, such as *edgeR* [3], *voom* and *limma* pipeline [7], and *DESeq* [5]. To mimic the distributions of real RNA-seq count data, we sampled gene-specific mean and dispersion parameters from the estimated values based on a maize study [26] that compared gene expression between bundle sheath and mesophyll cells of corn plants. Following Liu *et al.* (2015) [13], we conducted the same two sets of simulation studies (A and B). For each simulation study, 32 independent RNA-seq datasets were simulated from NB distributions with given mean and dispersion parameters, each dataset contains 10,000 genes, 2 treatment groups, and n replicates per treatment group, where $n = 3$ or 6. For our proposed method, we generated 5000 posterior samples after 3000 iterations burn-in, to calculate the estimated posterior probabilities. Convergence was checked via Gelman-Rubin criteria [27]. The test performances of different methods are evaluated by averaging the 32 datasets.

Simulation A

We used the maize dataset published by Tausta *et al.* (2014) [26] to estimate the gene-specific mean for one treatment group and the dispersion parameters, and randomly sampled 10,000 pairs of mean and dispersion parameters out of all 27,819 pairs without replacement, which would be used as the true mean expression level for the control group (μ_g) and the true dispersion parameter (ϕ_g) for gene $g = 1, \ldots, 10000$. Given the number of replicates per treatment group, $n = 3$ or 6, the RNA-seq read count data for the control group were generated from $NB(\mu_g, \phi_g)$ for gene g. Then we

randomly selected 5000 out of the 10,000 genes to be EE, whose count data for the treatment group were also drawn from $NB(\mu_g, \phi_g)$. The remaining 5000 genes were simulated to be DE genes, with fold change (ρ_g) set to be 4, 8, 0.25 and 0.125. Thus we had 1250 genes for each ρ_g value, whose count data for the treatment group were drawn from $NB(\mu_g\rho_g, \phi_g)$.

Simulation B

Similar to Simulation A, we generated 10,000 genes from $NB(\mu_g, \phi_g)$, with fold change ρ_g for 5000 DE genes. Instead of setting ρ_g to be 4, 8, 0.25 or 0.125, we simulated ρ_g from a two-component mixture of lognormal distributions,

$$log(\rho_g) \sim 0.5\text{Normal}(log(4), 1) + 0.5\text{Normal}(-log(4), 1).$$

Simulation Results for Testing DE Genes

In order to avoid the impact on test performance with different normalization procedures, we applied the same normalization steps for all the methods under comparison. Specifically, we set all normalization factors to be 1 for both Simulations A and B.

The receiver operating characteristic (ROC) curves that plot the true positive rate (TPR) versus false positive rate (FPR) resulting from Simulations A and B with number of replicates per group $n = 3$ or 6 are shown in Fig 1. These curves were generated based on either the posterior probabilities or $p-$values for each method. For each level of FPR, the TPRs were averaged over the 32 simulated datasets. We plotted the curves over the FPR values in the range of 0 and 0.1 because we are most interested in small FPR values. We also calculated the area under the curve (AUC) values as the percentages of 0.1, which is the total area in the range of FPR < 0.1. The average values and standard deviations of the AUC across the 32 simulated datasets are reported in the legends of Fig 1. Fig 1 shows that our *iSBA* method and the *SBA* method proposed by Liu *et al.* (2015) [13] generated the highest ROC curves and largest AUC values among all tests under all simulation settings, indicating that *iSBA* and SBA methods outperformed other methods in terms of ranking DE genes.

Fig 1. ROC curves resulting from Simulations A and B. For each level of FPR, the TPRs were averaged over the 32 simulated datasets. The percentage reported in the legend is the average AUC for each method, representing the percentage of 0.1, which is the total area in the range of FPR < 0.1, and the percentage in each set of parentheses is the standard deviation of the estimated AUC.

We also checked the false discovery (FD) plot as in Liu *et al.* (2015) [13], which is the plot of the number of false positives versus the number of top ranked genes selected as DE. Genes were ranked based on either posterior probabilities or $p-$values for each method. A better performing method would have a lower FD curve. The FD plots for Simulations A and B with $n = 3$ or 6 are shown in Fig 2. The number of false positives decreased when sample size increased from 3 to 6 for all methods, as expected. Our *iSBA* method and the *SBA* method provided the lowest FD curves under all simulation settings, indicating that our *iSBA* method and the *SBA* method produced less false positives than others, when we declared the same number of DE genes for all methods.

In addition, we evaluated the estimation of FDR based on subsection "Bayesian FDR Control" in Methods Section for our method and *SBA* method. For other non-Bayesian methods, we applied the Benjamini and Hochberg [23] procedure to adjust $p-$values for multiple comparisons. FDR plots for Simulations A and B with $n = 3$ or 6

Fig 2. False discovery curves resulting from Simulations A and B. For each number of top ranked genes selected as DE, the number of false positives were averaged across the 32 simulated datasets. Genes were ranked based on either posterior probabilities or $p-$values.

are presented in Fig 3. Our *iSBA* method controlled FDR well, the *SBA* method performed the second, while other methods provide more conservative results.

Fig 3. Plots of the actual FDR versus the nominal level of FDR resulting from Simulations A and B. The dashed lines correspond to the $Y = X$ line. A well performing method would control the FDR below or close to the dashed line.

Based on results from these simulations, our *iSBA* method and the *SBA* method generated the highest ROC curves and the least false positives, comparing with other popularly applied RNA-seq DE analysis methods. Furthermore, the *iSBA* method controlled FDR the best, hence provided reliable lists of declared DE genes. All in all, our proposed *iSBA* method worked the best or among the best under all simulation settings.

Simulation Results for Testing: $|logFC| \leq log1.5$

In addition to the simple hypothesis testing problem introduced in the last subsection, we could also apply our method to do other types of hypothesis testing, for example, testing whether the fold change falls into a certain interval or not. In practice, biologists often want to detect genes whose fold-changes are big enough and biologically meaningful. This subsection shows the results for testing: $|logFC| \leq log1.5$ for Simulation B.

We applied our *iSBA* method and the *SBA* method directly to do this hypothesis testing problem. For other methods including *edgeR*, *voom* and *limma* pipeline, and *DESeq*, we adopted the two-step procedure described in [11]. More specifically, in the first step, $\rho_g = 1$ was tested for each gene, and a list of DE genes was identified while controlling FDR at a given level. In the second step, among those DE genes declared in the first step, genes with large enough fold changes ($|logFC| > log1.5$) were selected.

The ROC curves for testing $|logFC| \leq log1.5$ for Simulation B are shown in the upper panel of Fig 4. The *iSBA* method and the *SBA* method outperformed all other methods. The lower panel of Fig 4 provides the FDR plots, from which we could notice that our *iSBA* method controlled FDR well in the range of FDR smaller than 0.1.

Fig 4. Results for testing $|logFC| \leq log1.5$ from Simulation B. The upper panel shows the ROC curves. For each level of FPR, the TPRs were averaged over the 32 simulated datasets. The percentage reported in the legend is the average AUC for each method, representing the percentage of 0.1, which is the total area in the range of FPR < 0.1, and the percentage in each set of parentheses is the standard deviation of the estimated AUC. The lower panel plots the actual FDR versus the nominal level of FDR. The dashed lines correspond to the $Y = X$ line.

Simulation Results for Swapping Treatments

As we discussed in the Introduction Section, the semi-parametric Bayesian (*SBA*) method [13] set one treatment group as reference condition. If the choice of a reference condition is not obvious based on the experimental design, the declared differential

expression status may vary depending on which group is set to be baseline. However, the model we proposed is invariant no matter which group is set to be the reference condition. The proportion of genes remaining the same declared differential expression status between two analyses that swapped the treatment and control groups were calculated when controlling FDR at 0.05, with average values and standard deviation of the percentage across the 32 simulated datasets reported in Table 1. It turned out that our *iSBA* method had higher overlap and more consistency in declared differential expression status than *SBA* method when swapping the treatment and control groups, for all simulation settings.

Table 1. Proportion of genes remaining the same declared differential expression status between two analyses that swapped the treatment and control groups for Simulations A and B, when controlling FDR at 0.05. The proportions were averaged across the 32 simulated datasets, and the percentage in each set of parentheses is the standard deviation of the estimated proportion.

Simulation setting	*SBA*	*iSBA*
Simulation A, $n = 3$	91.43% (2.21%)	92.43% (0.33%)
Simulation A, $n = 6$	93.12% (5.55%)	94.87% (0.48%)
Simulation B, $n = 3$	91.42% (4.80%)	93.78% (0.84%)
Simulation B, $n = 6$	93.98% (2.23%)	94.83% (0.39%)

Real Data Analysis

In this subsection, we analyze a real RNA-seq dataset published by Li et al. (2010). The dataset measures the transcript abundance of two cell types, bundle sheath and mesophyll, for different leaf sections. Each cell type has two biological replicates. The objective of the analysis is to detect genes that are DE between cell types or between different leaf sections. We analyzed leaf section 4 to detect DE genes between the two cell types in this section.

After deleting genes that have zero counts for both replicates in either cell type, 28,407 out of 33,743 genes were retained for analysis. We assumed NB models for the count data observed for each gene, and performed our proposed *iSBA* method, together with *SBA* method and *edgeR*. We also controlled FDR as described in subsection "Bayesian FDR Control" for *SBA* and *iSBA*, and applied the Benjamini and Hochberg [23] procedure for *edgeR*.

The numbers of DE genes detected by different methods while controlling FDR at different levels are shown in Fig 5. For example, when we controlled FDR at 0.05, 6040 genes were detected by all three methods. The majority of genes identified by our *iSBA* method were overlapped with *SBA*. 2703 genes were detected by both *iSBA* and *SBA*, but not by *edgeR*, which may due to the conservative control of FDR based on our simulation studies.

Fig 5. The numbers of DE genes between two cell types for leaf section 4. The Venn diagram on the left shows the number of overlapping identified DE genes from our *iSBA* method, *SBA* method, and *edgeR* while controlling FDR at 1%; the Venn diagram on the right shows the corresponding results while controlling FDR at 5%.

The proportions of genes remaining the same declared differential expression status between two analyses that swapped the two treatment groups when controlling FDR at 0.05 for our *iSBA* method is 93.47%, while the *SBA* method is 89.28%.

Discussion

In this paper, we proposed a Bayesian mixture modeling procedure for DE analysis of RNA-seq count data, and employed the MCMC sampling scheme to generate posterior samples for further inference. Simulation results demonstrate that our method outperformed other commonly used methods, such as *edgeR*, *voom* and *limma* pipeline, and *DESeq*, in terms of both ranking DE genes and FDR control.

A common choice of the concentration parameter M in the DP priors that are widely used in application is $M = 1$ [12]. We check the simulation results with different M values (M being 0.2, 0.5, 2, 5, 10 or 20), and the results remain almost the same for various values of M.

In our proposed method, the DP prior we choose guarantees that our modeling is invariant regardless of which treatment group is set to be the reference condition. According to the simulation results on two analyses that swapped the treatment and control groups, it is worth noticing that even for our *iSBA* method, the declared differential expression status are still not 100% the same. Part of the reason is due to the randomness of MCMC, if we run another MCMC using different seed, the overlap between the two MCMCs is about 97%. Since we generated 5000 posterior samples to calculate the estimated posterior probabilities after 3000 iterations burn-in, whether the Markov chains are long enough to get accurate results is also a potential problem. We checked the effective sample size for each gene, genes that had the same declared DE status after swapping treatments had effective sample sizes about 500 or larger, but genes that had different declared DE status overlapped had effective sample sizes around only 100. Effective sample size around 400 can be regarded as large enough, so for those genes with low effective sample size, we may need to run longer MCMC. Based on simulation checking, running the Markov chains longer do increase the percentage of overlapping genes, as expected. For example, for simulation A with $n = 6$, if we doubled the length of chain, the overlap for *iSBA* increased to 95.28%. However, running longer chains is more time consuming, and it only benefits a small proportion of genes while results for most genes would not change. Therefore, it is a tradeoff between efficiency and accuracy, and we will let the users decide which one is more important for a practical application.

As indicated in subsection "Bayesian FDR Control", the estimated posterior probability $\hat{v}_g$ is used as a test statistic and a decision D_g is based on whether $\hat{v}_g$ is small enough. And the AMAP test by Si and Liu (2013) [11] is based on a similar test statistic except that the prior models are different. In fact, the MAP test statistic derived in Si and Liu (2013) [11] can be viewed as the posterior probability of being null given the distribution of gene-specific parameters under the null hypothesis and the distribution of these parameters under the alternative hypothesis. Assuming the distributions of the gene-specific parameters (fold changes and other parameters) follow approximately the prior distribution we use, our estimated posterior probability using MCMC is an AMAP test statistic.

Acknowledgments

The authors would like to thank Emily Goren from Iowa State University for proofreading the manuscript.

References

1. Robinson MD, Smyth GK. Moderated statistical tests for assessing differences in tag abundance. Bioinformatics. 2007;23:2881–2887.

2. Robinson MD, Smyth GK. Small-sample estimation of negative binomial dispersion, with applications to SAGE data. Biostatistics. 2008;9:321–332.

3. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26:139–140.

4. McCarthy DJ, Chen Y, Smyth GK. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Research. 2012;40:4288–4297.

5. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biology. 2010;11:R106.

6. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. Genome Biology. 2014;15(12):550.

7. Law CW, Chen Y, Shi W, Smyth GK. Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol. 2014;15:R29.

8. Smyth GK. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Statistical Applications in Genetics and Molecular Biology. 2004;3:Article 3.

9. Kvam VM, Liu P, Si Y. A Comparison of Statistical Methods for Detecting Differentially Expressed Genes from RNA-Seq Data. American Journal of Botany. 2012;99(2):248–256.

10. Soneson C, Delorenzi M. A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics. 2013;14:91.

11. Si Y, Liu P. An Optimal Test with Maximum Average Power While Controlling FDR with Application to RNA-seq Data. Biometrics. 2013;69:594–605.

12. Do KA, Muller P, Tang F. A Bayesian Mixture Model For Differential Gene. Journal of the Royal Statistical Society, Series C (Applied Statistics). 2005;54:627–644.

13. Liu F, Wang C, Liu P. A Semi-parametric Bayesian Approach for Differential Expression Analysis of RNA-seq Data. J Agric Biol Environ Stat. 2015;20(4):555–576.

14. Peart MJ, Smyth GK, van Laar RK, Bowtell DD, Richon VM, Marks PA, Holloway AJ, Johnstone RW. Identification and functional significance of genes regulated bystructurally different histone deacetylase inhibitors. Proceedings of the National Academy of Sciences of the United States of America. 2005;102:3697–3702.

15. Green PJ, Richardson S. Modelling Heterogeneity With and Without the Dirichlet Process. Scandinavian Journal of Statistics. 2001;28:355–375.

16. Kalli M, Griffin J, Walker S. Slice Sampling Mixture Models. Statistics and Computing. 2011;1:93–105.

17. Tierney L. Markov Chains for Exploring Posterior Distributions. The Annals of Statistics. 1994;22(4):1701–1728.

18. Hastings WK. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika. 1970;57:97–109.

19. Blackwell D, MacQueen BJ. Ferguson Distributions Via Polya Urn Schemes. The Annals of Statistics. 1973;1(2):353–355.

20. Escobar MD. Estimating Normal Means With a Dirichlet Process Prior. Journal of the American Statistical Association. 1994;89:268–277.

21. Neal RM. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. Journal of Computational and Graphical Statistics. 2000;9(2):249–265.

22. Gilks WR. Adaptive Rejection Sampling for Gibbs Sampling. Applied Statistics. 1992;41:337–348.

23. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. B. 1995;57:289–300.

24. Genovese C, Wasserman L. Bayesian and Frequentist Multiple Testing. Bayesian Statistics. 2003;7:145–161.

25. Newton MA, Noueiry A, Sarkar D, Ahlquist P. Detecting Differential Gene Expression with a Semiparametric Hierarchical Mixture Method. Biostatistics. 2004;5:155–176.

26. Tausta SL, Li P, Si Y, Gandotra N, Liu P, Sun Q, Brutnell TP, Nelson T. Developmental dynamics of Kranz cell transcriptional specificity in maize leaf reveals early onset of C4-related processes. Journal of Experimental Botany. 2014;65:3543–3555.

27. Gelman A, Rubin DB. Inference from iterative simulation using multiple sequences. Statistical science. 1992;7:457–472.

Supporting information

S1 Appendix. Proof of Model Invariance.

S2 Appendix. Detailed MCMC Sampling Scheme.

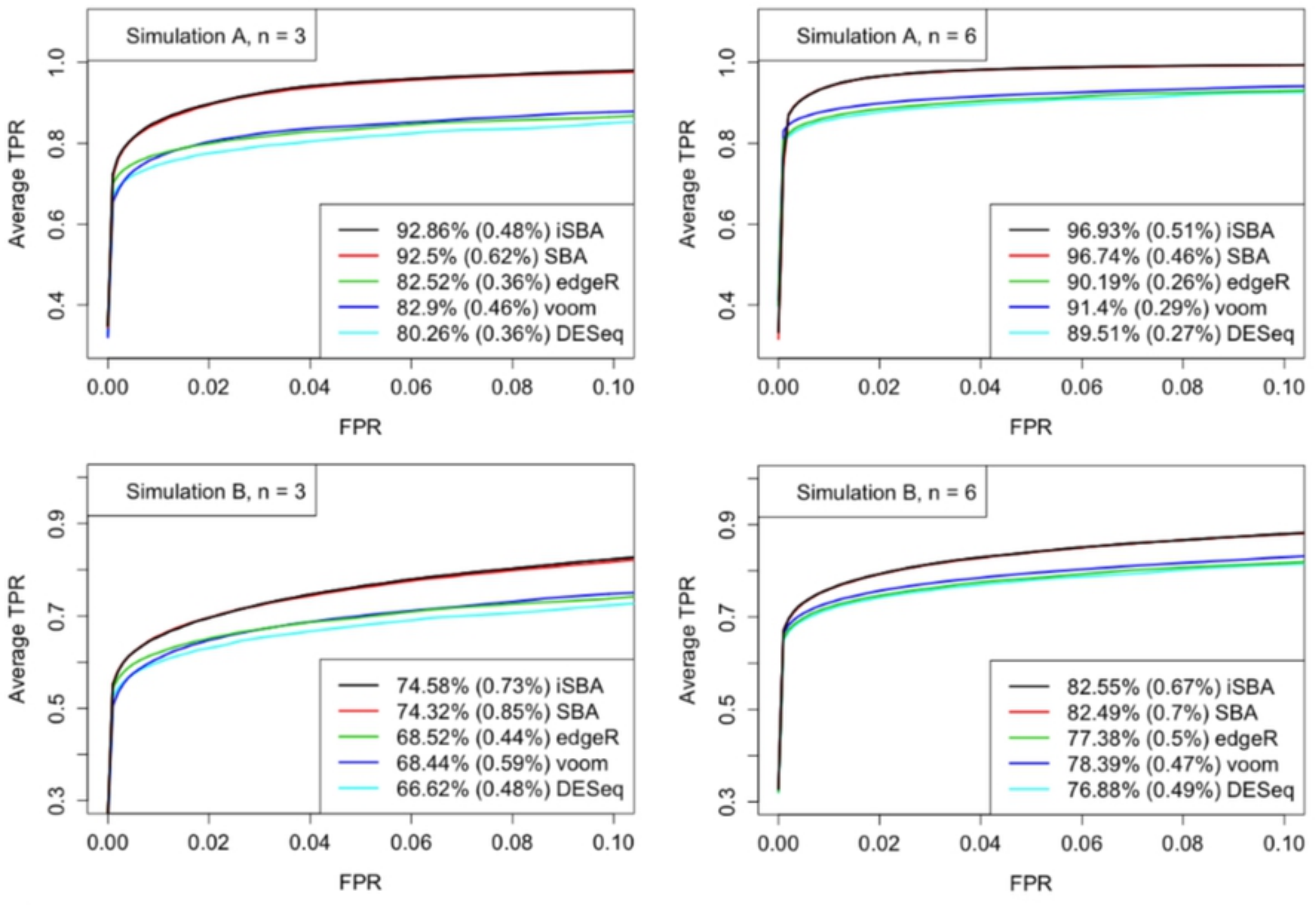

Simulation A, n = 3
Average TPR
FPR
92.86% (0.48%) iSBA
92.5% (0.62%) SBA
82.52% (0.36%) edgeR
82.9% (0.46%) voom
80.26% (0.36%) DESeq
Simulation A, n = 6
96.93% (0.51%) iSBA
96.74% (0.46%) SBA
90.19% (0.26%) edgeR
91.4% (0.29%) voom
89.51% (0.27%) DESeq
Simulation B, n = 3
74.58% (0.73%) iSBA
74.32% (0.85%) SBA
68.52% (0.44%) edgeR
68.44% (0.59%) voom
66.62% (0.48%) DESeq
Simulation B, n = 6
82.55% (0.67%) iSBA
82.49% (0.7%) SBA
77.38% (0.5%) edgeR
78.39% (0.47%) voom
76.88% (0.49%) DESeq

Figure 1

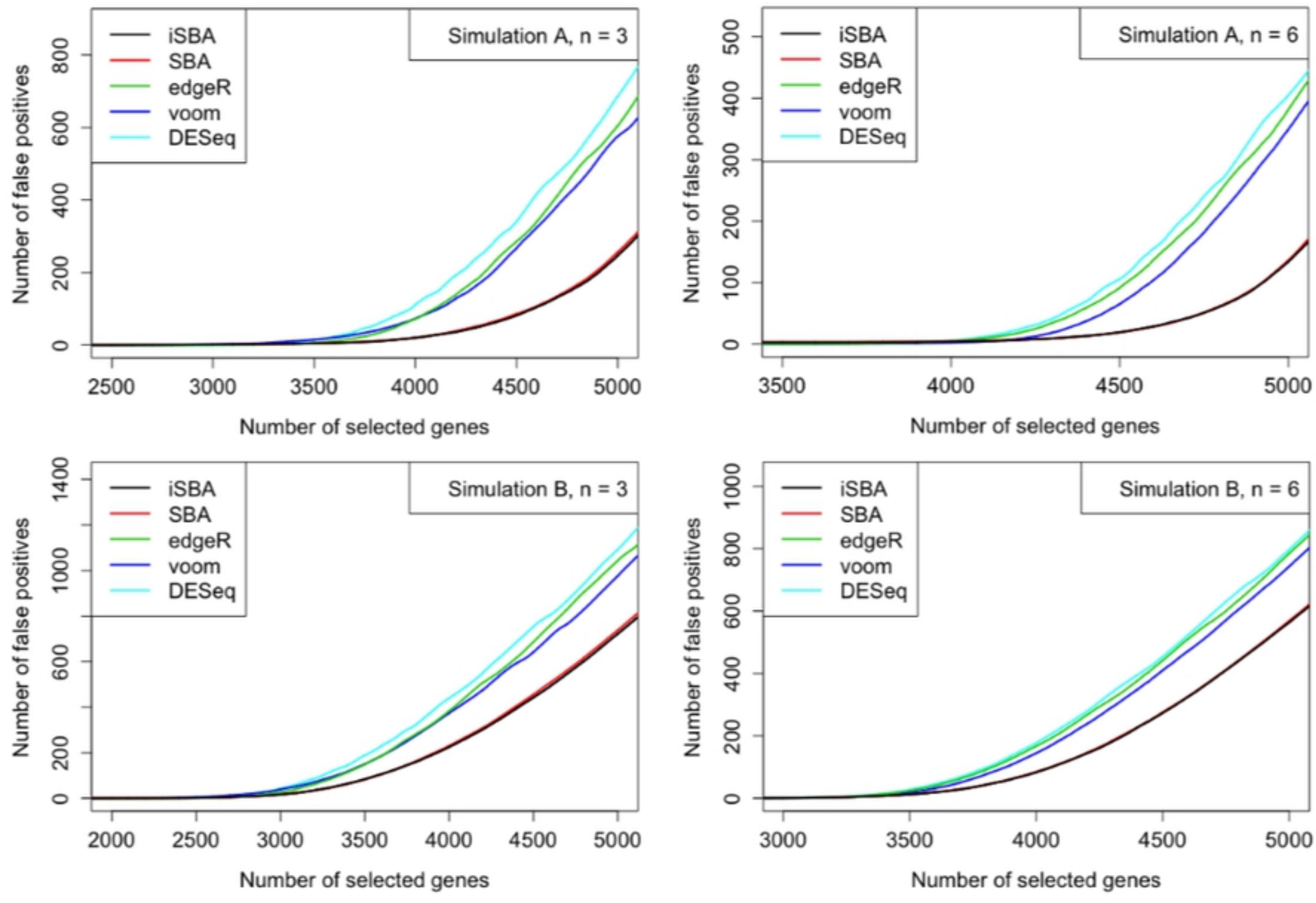

Simulation A, n = 3
iSBA
SBA
edgeR
voom
DESeq
Number of false positives
0
200
400
600
800
2500
3000
3500
4000
4500
5000
Number of selected genes
Simulation A, n = 6
iSBA
SBA
edgeR
voom
DESeq
Number of false positives
0
100
200
300
400
500
3500
4000
4500
5000
Number of selected genes
Simulation B, n = 3
iSBA
SBA
edgeR
voom
DESeq
Number of false positives
0
200
600
1000
1400
2000
2500
3000
3500
4000
4500
5000
Number of selected genes
Simulation B, n = 6
iSBA
SBA
edgeR
voom
DESeq
Number of false positives
0
200
400
600
800
1000
3000
3500
4000
4500
5000
Number of selected genes

Figure 2

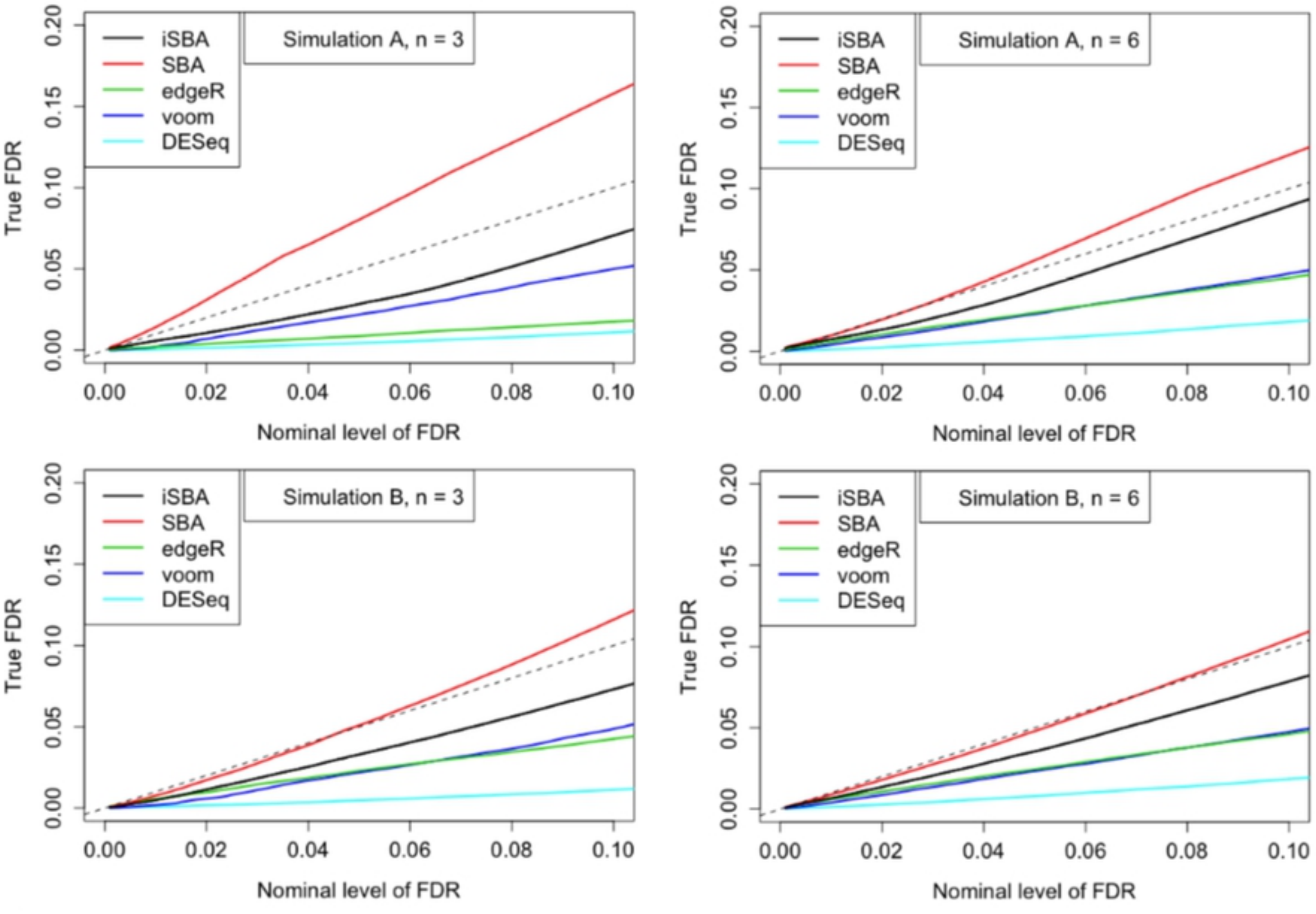

Simulation A, n = 3
Simulation A, n = 6
Simulation B, n = 3
Simulation B, n = 6
iSBA
SBA
edgeR
voom
DESeq
True FDR
Nominal level of FDR
0.00
0.05
0.10
0.15
0.20
0.02
0.04
0.06
0.08

Figure 3

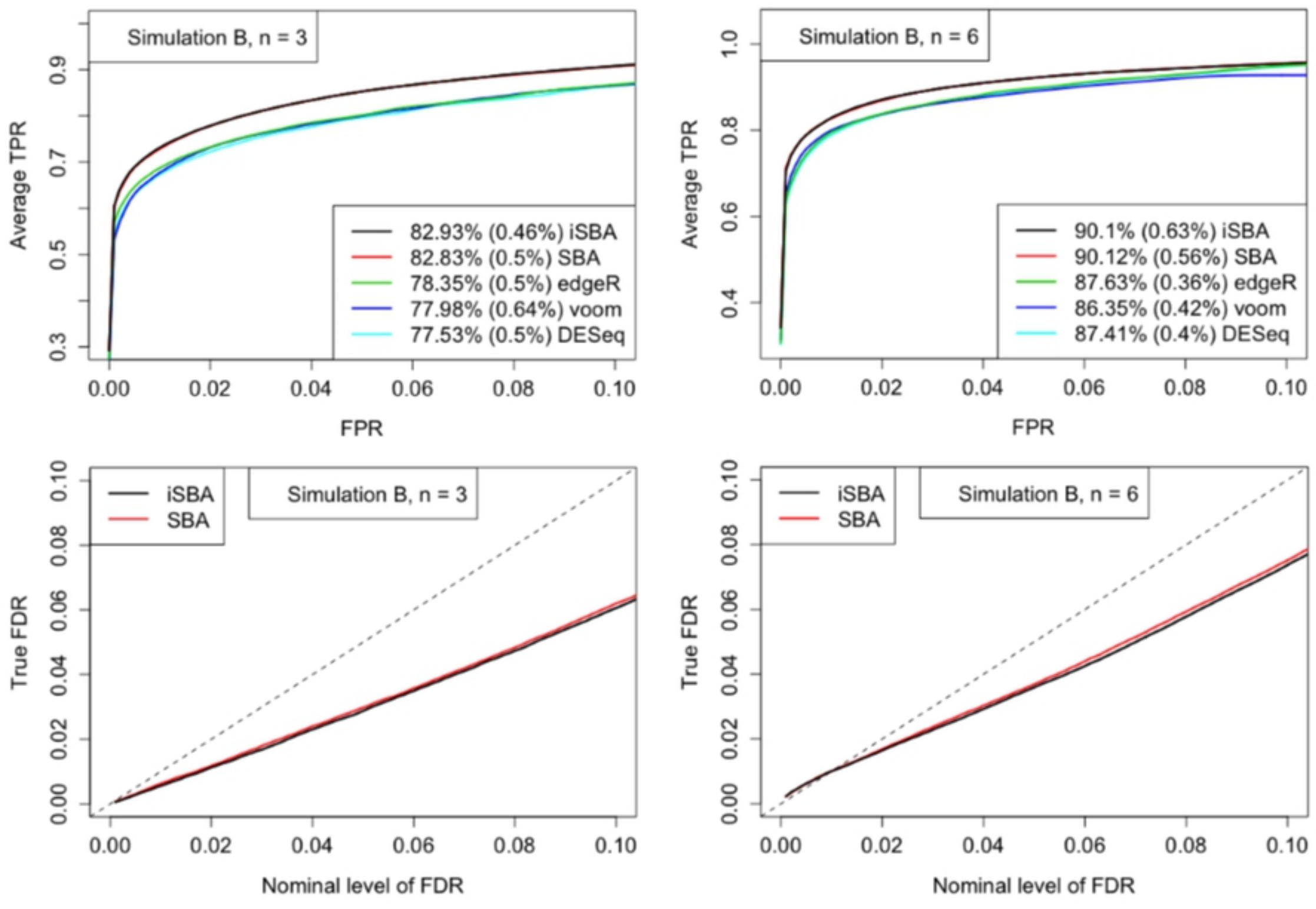
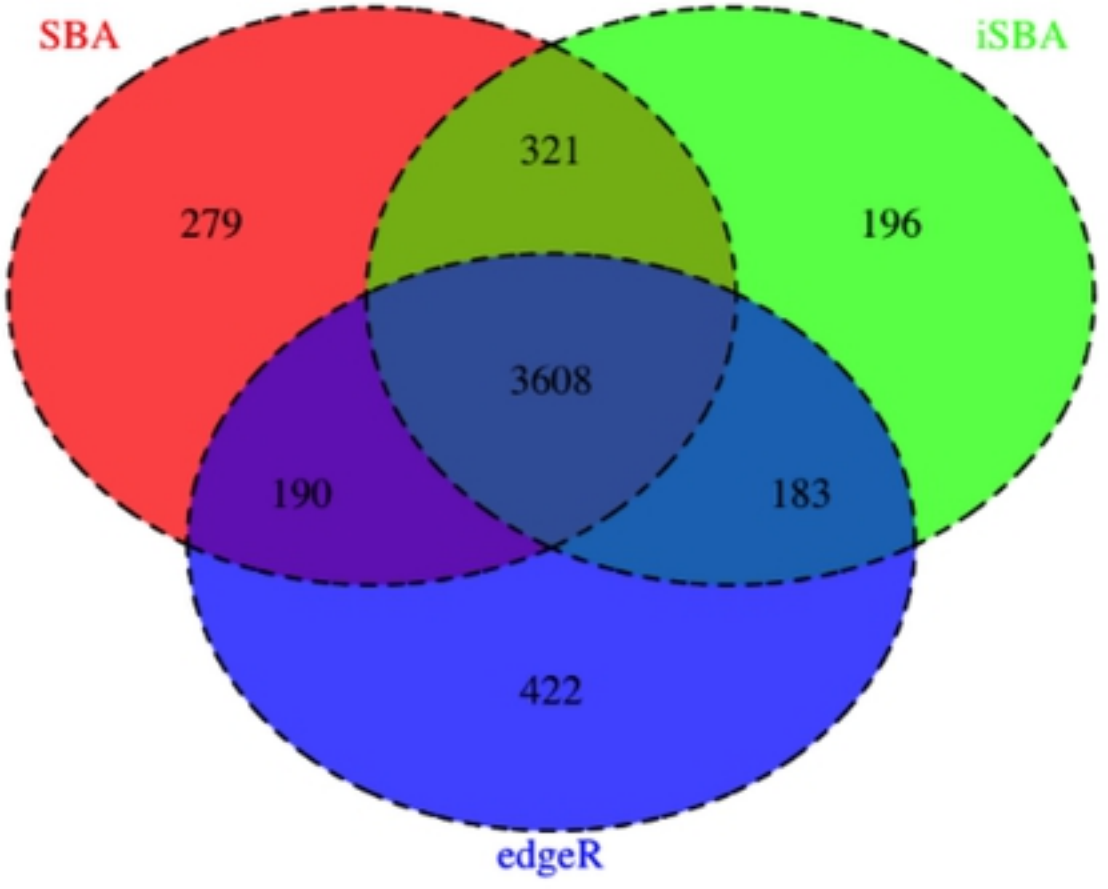

Simulation B, n = 3
Average TPR
FPR
82.93% (0.46%) iSBA
82.83% (0.5%) SBA
78.35% (0.5%) edgeR
77.98% (0.64%) voom
77.53% (0.5%) DESeq
Simulation B, n = 6
90.1% (0.63%) iSBA
90.12% (0.56%) SBA
87.63% (0.36%) edgeR
86.35% (0.42%) voom
87.41% (0.4%) DESeq
iSBA
SBA
True FDR
Nominal level of FDR

Figure 4


SBA
iSBA
279
321
196
3608
190
183
422
edgeR

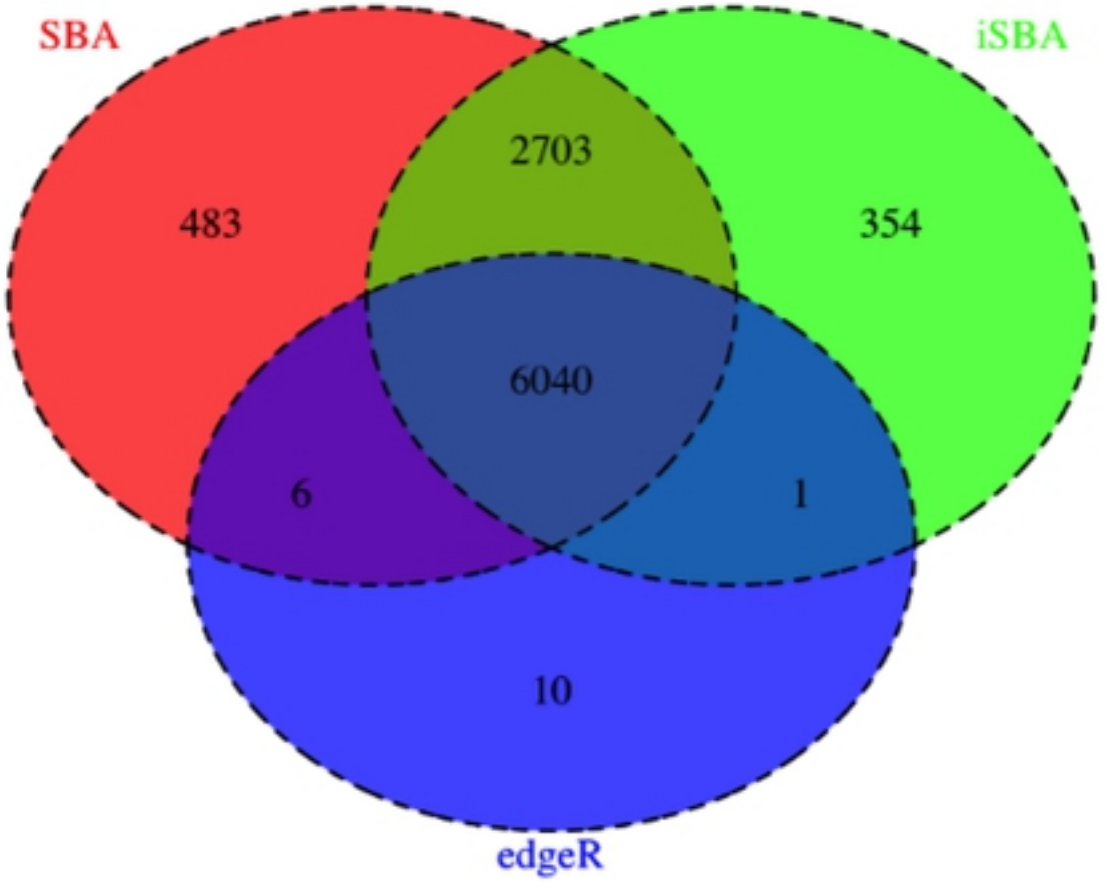

SBA
iSBA
483
2703
354
6040
6
1
10
edgeR

Figure 5