Overcoming the Common Challenges in Differential Gene Expression Analysis Studies

by

Yan Huang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Haiyan Huang, Chair
Professor Sandrine Dudoit
Professor Lewis J Feldman

Summer 2019

Overcoming the Common Challenges
in Differential Gene Expression Analysis Studies

Abstract

Overcoming the Common Challenges in Differential Gene Expression Analysis Studies

by

Yan Huang

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Haiyan Huang, Chair

The ability to analyze gene expression data has had a fundamental impact in the biological sciences and on our understanding of the causes and mechanisms of disease. However, a significant statistical challenge is posed by the combination of the small number of replicates together with the large number of genes leading to an undesirable level of misclassified genes when identifying genes with differential expression levels. When multiple gene expression data sets are generated under the same set of experimental conditions, the question arises as to how to efficiently combine this information. Several methods in the literature have been suggested to aggregate ranked data from multiple sources. We introduce a new classifier, underpinned by Bayesian principles, called *Peer Reinforced Ranker* (PR-Ranker) which uses density estimation to approximate the probability that a gene is differentially expressed given a collection of ranked lists.

Our classifier is amenable to theoretical analysis when the number of genes and lists is large using the theory of large deviations. Under modest technical assumptions we show that asymptotically PR-Ranker has the smallest loss of any rank aggregation procedure. Moreover, we prove that other more ad hoc methods, such as Borda, have a strictly higher asymptotic rate of loss.

While the theoretical results are asymptotic, we perform a series of simulation studies that demonstrate that our classifier outperforms existing methods on datasets of realistic size for biological data. Furthermore, we show that the outperformance is even greater when the lists exhibit varying levels of noise or when some sources are corrupted. PR-Ranker automatically adapts to varying data quality and efficiently combines the data from different sources. Finally we apply PR-Ranker to a gene expression data set in a preeclampsia study. The top ranked genes identified were known to be biologically relevant to preeclampsia and our method achieved a substantially higher Consistency Index than other rank aggregation procedures.

In fond memory of my beloved father, Hua Huang.

Contents

Acknowledgments

The road leading to the completion of my doctorate degree many times seemed endless. At times it felt as if I was lost in a gigantic maze, running through the seemingly familiar passages and only finding myself ending up at the same spot over and over.

The main force that kept me going was the wonderful people surrounding me. My advisor, Haiyan Huang, who is both a mentor and a friend to me, saw the potential in me even when I was thinking about giving up on finishing my degree. My parents always provided me an ample amount of liberty to pursue my interests (even when they were worried about my choices!). When I was a child, Mother would walk me to Sunday drawing school every weekend. For three years, through the pouring rain, in the searing heat and in the freezing cold, we hardly missed a class. Then, suddenly one day I declared that I was bored and would like to stop; instead of the anticipated scolding, the only response that I received from Mother was "are you sure?" After that, we stopped our weekly march to the drawing school and Mother never resented the three years that she invested in for me. A few months after that, I realized that I had made a mistake and without a single complaint my parents were running around the city trying to get me into the top drawing school in the city which eventually led me to my brief Graphic Design career before I discovered the joy of Statistics. I never fully understood why Father was so proud of me practically about everything, bragging about me to everyone, including the chef in the neighborhood restaurant and random strangers that we met on the street. After years of training I had developed the reflex of seeking immediate concealment for myself whenever the “embarrassing” bragging began. Then, one day, before I could intrinsically appreciate his deep affection, Father was gone. It was so sudden that the entire family was caught unprepared. It was in the middle of my Ph.D. study. I was devastated, thinking how unfair life was to take away someone that was so loving and kind. However, part of growing up is to live with the endearing memories of the past and still to be able to appreciate the beauty of the present. I have transitioned from a daughter to a wife and my husband, Allan Sly, reminds me a lot of Father when it comes to kindness, affection, sense of humor, giving me the freedom to be myself and the passion for food too! We have lived in difference places, made significant life decisions together and supported each other on daily basis and in major life events. As two intertwined vines, we gain strength in relying on each other and together we continue growing strongly. The completion of this dissertation would not be possible without my husband’s persistent support.

Chapter 1

Introduction

Gene expression profiling technology has opened the door for researchers to study the activities of tens of thousands of genes at a time and the technology has become one of the major achievements in experimental molecular biology. A growing amount of gene expression data is generated every year, awaiting scientist to reveal the hidden biological information in them.

One remarkable effort in gene profiling analysis is the detection of genes that are differentially expressed (DE) under two or more experimental or biological conditions. Due to cost and time constraints biologists normally have the resources to study only a limit number of the differentially expressed genes (DEG) detected in the analysis. Because of this, methods in differential gene expression (DGE) studies revolve around establishing a rank list of the genes to prioritize genes that demonstrate strong evidence that they are DE.

Traditionally gene expression datasets were analyzed individually. The major difficulties in gene expression analysis are due to two intrinsic properties of gene expression data: the small number of replicates and the large number of genes. The first property makes getting a reliable sample statistic (such as the sample variance) difficult; the second property inevitably introduces the multiple testing problem. Many methods have been developed to tackle these challenges.

As more gene expression data generated under the same set of experimental or biological conditions became available, a new class of rank aggregation methods emerged. These methods combine the results from multiple studies by using the rank statistics calculated from the individual studies. There are many benefits of using a rank-based approach to aggregate results from different experiments. For one, rank statistics are scale invariant; secondly, rank-based methods usually require few distributional assumptions.

In this thesis we develop a rank aggregation method for classifying DEG.

Our approach is based on Bayesian principles and we prove using large deviation theory that our classifier has the smallest loss among all the rank-based classifiers. In addition, through simulations and a real data application we show that our classifier is robust in the presence of data corruption or when the strength of signal varies among the results being combined.

Our method is broadly applicable and can be used to combine datasets collected with different kinds of sequencing technology which includes, but is not limited to, microarray and RNA-seq. In fact, the applications of our method are not restricted to gene expression datasets and can be any rank datasets that satisfy the few assumptions of our model.

1.1 Background

Gene expression profiling technology enables scientists to study simultaneously the behavior of tens of thousands of the genes of an organism under a certain condition in one experiment. For example, one of the most popular gene expression profile applications is to detect changes in gene expression levels across changes in phenotype or under different experimental conditions. If a given gene's expression level's change can be associated with a change in phenotype, then perhaps the gene plays a role in the initiation or progression of the phenotype of interest. For example, one focus of functional genomics is the study of understanding and curing disease. It is well known that many genetic alternations signify presence of abnormalities or diseases. For instance, point mutations might induce altered protein or changes in expression level [21], loss or gain of gene copies–which might result in reduced or increase in expression level–are related to tumor suppression or oncogene activation, respectively, and methylation might cause changes in expression level and is also related to oncogene tumor suppressors [41].

In DGE analysis studies, researchers are interested in detecting changes in gene expression levels across variations in a phenotype. A gene whose expression level varies in response to changes in a particular phenotype of interest worths the attentions of researchers to further investigate the gene's role in the initiation or progression of a particular change in the phenotype. However, DGE analyses often face analytical challenges in areas, such as, data normalization and statistical analysis. In the next section we will focus our discuss on the major statistical challenges presented in DGE analyses.

1.2 Challenges in Gene Expression Analysis

One major challenge in gene expression statistical analyses is introduced by the small number of replicates accompanied with the large number of genes properties of gene expression data.

Small Sample Size

Cost and biological constraints often lead to insufficient replicate samples for obtaining stable and accurate variance estimates for the analysis. In addition, even in the cases where a relatively large number of samples are available often times the measurements are technically replicated samples taken from a small number of tissues/cells; i.e., the number of biologically replicated samples is small. Thus, it is difficult to separate the biological variation from the systematic measurement errors in the data. As a result, the small sample size property of the data creates a serious obstacle in DE gene detections since most traditional statistical methods are based on the assumption that the samples are independent and identically distributed (i.i.d.) measurements from a distribution and that the sample size is sufficiently large.

Large Number of Genes

Another difficulty in gene expression analysis was imposed by the large number of genes in experiment. For example, it is estimated that there are at least about 20k human protein-coding genes [18]. Suppose gene expressions are measured under two experimental conditions, control and treatment, and suppose one would like to identify the set of genes that are differentially expressed under the treatment condition. One can calculate the p-values against the null hypothesis that there is no change in the mean gene expressions under the two conditions; however, running 20k hypotheses increases the chance of getting at least one test wrong. For example, if we run the tests at 5% significance level, even if the null hypotheses were all true (i.e., none of the genes are DE) the number of genes that would be identified as DE with the tests just by random chance would be about $20,000 \times .05 = 1000$.

We will next review some of the existing statistical methods that have been developed specifically for analyzing gene expression data with the aforementioned properties.

Chapter 2

Review of Existing Methods

As mentioned in the last chapter the major obstacles in gene expression analysis are due to the the large number of genes and the small number of samples in gene expression data. Many statistical methods have been proposed to overcome these problems.

Among these methods, information sharing often tends to play an important role in guiding the development of the techniques to remedy the small sample size issue; for the issue with the large number of genes, some algorithms for controlling family-wise error rate and for controlling false discovery rate have been proposed. The methods for analyzing biologically defined gene sets rather than individual genes have also been proposed; these methods group genes with similar biologically roles into a group and test the significance of the gene set rather than the individual genes; such approach respects the biological relationship between the genes and reduces the number of hypothesis tests required (in turn lessening the issue with multiple testing).

On the other hand, as more gene rank lists (i.e., lists of ranks that sort genes according to the strength of evidence that they are DE) became more readily available, aggregating results produced by different experiments under the same set of biological or experimental conditions has increasingly gained in popularity; the objective of these aggregation methods is to increase the power of the tests and to reduce the false positive error rate by combining information obtained from multiple experiments.

We will describe these methods in the context of a DGE study with two experimental conditions: control and treatment. Note that most of these methods can be extended to the case when there are multiple experimental conditions; however, for simplicity of the notations the case of two experimental conditions will be utilized here for the demonstration.

Suppose in a DGE study the biologist observes the expressions for gene

i: $x^c_{i,1}...x^c_{i,m_c}$ and $x^t_{i,1}...x^t_{i,m_t}$ under the control and treatment conditions, respectively, where m_c and m_t are the number of replicates for gene i under the control and treatment conditions, respectively. The goal of DGE analysis is to provide the biologist a list of genes whose expressions are believed to be altered under the treatment condition compared to that under the control condition. As mentioned in the previous chapter these genes are often of particular interest to biologists since they often play special roles in some biological process.

A common practice in DGE analyses is to rank all the genes of interest according to their p-values against the null hypothesis that there is no change in the mean DGE. From a practical point of view, getting a gene rank list is as useful, if not more, as finding a set of statistically significant genes. For instance, even if there were 200 genes found to be significantly differentially expressed, the researcher might have the resource to look at only the 100 most significant ones; on the other hand, in the case where there was no gene being classified as significant, the researcher might still want to investigate the top ranked genes.

To identify the genes of interest one could construct a two-sample Welch t-test for gene i with the t-statistics:

$$t_i = \frac{\bar{x}_{i,t} - \bar{x}_{i,c}}{s_i} \tag{2.1}$$

where $i = 1, \ldots, n$. Here $\bar{x}_{i,t} = \Sigma^{m_t}_{k=1} x^t_{i,k}$ and $\bar{x}_{i,c} = \Sigma^{m_c}_{k=1} x^c_{i,k}$ are the gene expression sample means for the treatment and control groups, respectively, and

$$s_i = \sqrt{\frac{s^2_{i,t}}{m_t} + \frac{s^2_{i,c}}{m_c}} \tag{2.2}$$

is the estimated standard error (SD) for the sample mean differences, where $s^2_{i,c} = \frac{\Sigma^{m_c}_{k=1}(x^c_{i,k} - \bar{x}_{i,c})^2}{m_c - 1}$ and $s^2_{i,t} = \frac{\Sigma^{m_t}_{k=1}(x^t_{i,k} - \bar{x}_{i,t})^2}{m_t - 1}$ are the sample SD for the control and treatment groups, respectively.

There are several issues with using the Welch t-test for the application in discussion. First, due to the small sample sizes in the data the estimate, s_i, of the standard deviation (SD), is often unstable. Secondly, because of the large number of genes, testing all the genes simultaneously raise the issue of multiple testing.

The rest of this chapter will be broken down into two main sections. Section 2.1 describes the more traditional methods that were developed to handle the two issues in DGE analysis; these methods deals with the results from one DGE study at a time. Section 2.2 reviews some novel approaches that aggregate results from multiple experiments; such aggregation methods have become increasingly popular because of the rising availability of datasets that are suitable for this kinds of methods.

2.1 Single List Approaches

Regulating Variance Estimations with Peer Genes

When dealing with one dataset at a time the classical t-test seems to be one of the most intuitive tests to use to rank the genes. However, the performance of the the t statistic is greatly dependent on the accuracy of the variance estimate of the differential expressions. Technical and cost constraints usually prevent biologists from obtaining large enough samples to obtain a reliable estimate on the variance. Small sample size and multiplicity are the technical obstacles that often induce unstable estimate of the variances. For example, even though the chance of getting a variance estimate that is much different from the size of the true variance is small for an individual gene, because of the large number of genes being tested it is inevitable that some of the variance estimates are much smaller than the true variance. As a result the small variance estimates inflate the test statistic and some of the non-DE genes will be falsely classified as significant [28]. Having these inflated test statistics are particularly unfavorable–since the percentage of truly differentially expressed genes is very small, the inflated test statistics will have greater damaging impact on the sensitivity of the test.

To adjust for the unstable variance estimates due to the small sample size, various approaches have been proposed to group genes in a meaningful way to artificially "increase" the sample sizes. The granularity of the groups ranges from using the entire set of genes being studied to numerous subsets of the genes. The common theme among these approaches is to allow information sharing among genes that are similar to each other in some way. Although the number of replicates of each gene is small, perhaps one can take advantage of the large number of genes and use the information from other genes to regulate the variance estimate for the gene of interest.

Information Sharing among all Genes

The multivariate empirical Bayes statistic proposed by Tai and Speed [54] and the James-Stein-type shrinkage method proposed by Opgen-Rhein and Stimmer [44] both use some global quantity of the gene set, such as the median of the sample variances of all the genes in the dataset, as a regulator for the variance estimate for a single gene.

The multivariate empirical Bayes statistic proposed by Tai and Speed [54] is a modified version of the usual likelihood ratio statistic applied to hierarchical models. In the context of the paper by Tai and Speed a hierarchical model is used to regularize the variance estimate. To regulate the DGE variance estimate, prior distributions are put on the mean and the variance of a gene and the hyper-parameters of the priors are estimated from the data for all other genes. The idea is that if the distribution of the mean has a wide spread then more information is needed to be borrowed from other genes to regularize the variance estimate for a particular gene. Although initially this multivariate hierarchical empirical Bayesian model makes the calculation on the likelihood ratio statistic difficult, a proposed transformation of the data that separates the DGE into two independent parts–i.e., constant and non-constant parts–gives a close form of the statistic.

The intuition behind Tai and Speed's model is that although the number of replicates for each gene is small, perhaps one can pool genes that are similar in some way and try using the properties shared by these pooled genes to get a better estimate of the variance.

Opgen-Rhein and Stimmer proposed the James-Stein-type shrinkage method [44] that utilizes the global information shared by all the genes. The method searches for an estimator that is in the form of a linear combination of the sample variance and an regulator calculated by using some global information of the genes in the dataset, such as the median of the variances of all genes. Given a loss function the shrinkage factor is the one that minimizes the corresponding risk. Note that this shrinkage method requires much more relaxed assumptions on the data than the multivariate empirical Bayes model since it does not make any distributional assumptions on the data. However, although the solution of this method has a close form in case of a quadratic loss function is used, the solution is not guaranteed to have a close form in general.

In regard to the methods discussed in this section, although information sharing seems to be a good idea, we are concerned with using global information to smooth the variance estimate; since the number of genes in a dataset is usually very large, utilizing the global information is likely to result in over-smoothing. One might argue that both the empirical Bayes and the Jame-Stein-type statistics can be easily modified to use local rather than global information as a smoother; however, the difficulty comes in when one has to decide on defining the subsets for local information sharing.

Grouping According to Sample Mean Expressions

A couple of more localized information-sharing methods have been reviewed by Huang and Pan [26]. One is to smooth (either with or without weights) the sample variances (for each experimental condition) for genes with similar mean expression levels. A similar but more complex approach is to, first, use a linear model to predict the gene expression variance for each gene. Then, smooth the predicted gene expression variances for genes with similar mean expression levels. Huang and Pan suggested using LOESS for the smoothing. Both of these methods were built on the assumption that the variance of the gene expression levels is a function of the mean [26] (and there is empirical evidence to support this) so genes with similar expression level have similar variances.

Grouping Similar Genes: Model Based Variance Estimation

Cai and Giannakis extended the idea of smoothing the estimated variance values based on expression sample mean values and introduced a method that groups genes not only by using their mean expression values but also the variances of the expressions: it organizes the genes with similar means and variance/covariance structures into clusters and estimates the variance for a particular gene by using the genes that are within the cluster. Then, Cai and Giannakis suggested using the estimated variances in the calculation for s_i in (2.2) instead. The attractiveness of this approach is that it respects the biological structure of the dataset.

Recall that in our setup with two experimental conditions $x^c_{i,1}...x^c_{i,m_c}$ and $x^t_{i,1}...x^t_{i,m_t}$ are the gene expression levels for gene i under the control and treatment conditions, respectively, where m_x and m_y are the number of replicates for gene i under the control and treatment conditions, respectively. Let $\mathbf{x}_i = [x^c_{i,1}...x^c_{i,m_c}, x^t_{i,1}...x^t_{i,m_t}]$. Cai and Giannakis' approach is built on the assumption that $\mathbf{x}_i$ follows a Normal mixture model; i.e., the probability density function of $\mathbf{x}_i$ is

$$f(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $0 \leq \pi_k \leq 1$, with $\sum_{k=1}^{K} \pi_k = 1$, is the mixing proportion of cluster k and $f_k(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the density function for the Normal distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$:

$$f_k(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-m/2}|\boldsymbol{\Sigma}_k|^{-1/2} exp\left\{-\frac{1}{2}(x_i - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(x_i - \boldsymbol{\mu}_k)^T\right\}$$

Here π_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ and K, where $k = 1, \ldots, K$, are the unknown parameters that need to be estimated. The Normal mixture model assumption here is reasonable since many distributions can by approximated by the Normal mixture model with proper choice of the parameters.

The EM algorithm was utilized to estimate the parameters $\{\pi_k\}_{k=1}^K$, $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ in the mixture model. (The EM algorithm is an iterative procedure for maximizing the log likelihood function of $\boldsymbol{\theta}$ given the data $\boldsymbol{X}$: $L(\boldsymbol{\theta}|\boldsymbol{X})$, where $\boldsymbol{\theta}$ contains all the parameters that need to be estimated; i.e., $\boldsymbol{\theta}$ is the set of $\{\pi_k\}_{k=1}^K$, $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$. Instead of maximizing $L(\boldsymbol{\theta}|\boldsymbol{X})$ at every iteration, the EM algorithm maximizes a function that is a lower bound on $L(\boldsymbol{\theta}|\boldsymbol{X})$. This lower bound of the log likelihood function can be chosen with the aid of the Jensen's inequality for convex functions. Jensen's inequality applies here because the log likelihood function is a concave function of the parameters that we are trying to estimate for the given data.)

An important point to note here is that the mixture model allows genes with similar mean and variance to share their information among them. In addition, as we will see next that with different forms of the cluster variance $\boldsymbol{\Sigma}_k$ the model has the flexibility to accommodate different assumptions about the variances for the replicates.

Three forms of the cluster variance $\boldsymbol{\Sigma}_k$ were proposed by Cai and Giannakis:

1. **model 1:** $\hat{\boldsymbol{\Sigma}}_k = \hat{\lambda}_k \mathbf{I}_m$;
2. **model 2:** $\hat{\boldsymbol{\Sigma}}_k = \mathsf{diag}\left(\hat{\lambda}_{k,c}\mathbf{I}_{m_c}, \hat{\lambda}_{k,t}\mathbf{I}_{m_t}\right)$;
3. **model 3:** $\hat{\boldsymbol{\Sigma}}_k = \mathsf{diag}\left(\hat{\lambda}_{k1}, \ldots, \hat{\lambda}_{km}\right)$.

Model 1 says that all the expression replicates for the genes in the same cluster share the same variance; model 2 says that all the replicates under the control condition have the same variance, and similarly for the replicates under that treatment condition; finally, model 3 says that all the replicates could have different variances.

Then, Bayesian information criterion (BIC) was used to choose the best candidate for K and the best model for the variance:

$$BIC = 2log[L(\hat{\boldsymbol{\theta}})] - n_p log(n) \tag{2.3}$$

where $L(\hat{\boldsymbol{\theta}})$ is the likelihood of the data with the estimated parameters contained in $\boldsymbol{\theta}$ and n_p is the number of independent parameters. (BIC can be thought of as a variation of the scaled maximum likelihood plus an additional term that penalizes overfitting.)

Discussion

The model proposed by Cai and Giannakis allows more flexibility for the biological structure of the dataset compared to the smoothing methods reviewed by Huang and Pan. In Cai and Giannakis's approach different forms of the cluster variance model different relationships between the replicates. In addition, because the clusters could have different sizes, the variances could be approximated with genes of different group sizes; (in contrast, the methods reviewed by Huang and Pan use a fixed window size for the smoothing).

Compared to the hierarchical empirical Bayes and the shrinkage models, the model based clustering method proposed by Cai and Giannakis uses local information rather than global information to avoid over-smoothing; however, the model based clustering method has a more restricted constraint since it assumes that the replicates are uncorrelated when the hierarchical empirical Bayes model relaxes that assumption.

Another disadvantage of Cai and Giannakis' method is that it could become computationally intensive when the number of genes being studied becomes large. Given that the number of genes in gene expression data is usually large, this posts a practical limitation on the method.

Controlling the Error Rate for Multiple Testing

As discussed in Section 1.2 one of the obstacles in gene expression analysis is induced by the large number of genes in gene expression datasets. A typical gene expression dataset comprising at least a few thousands of genes; thus, testing for all the genes in the dataset simultaneously brings up the concern of multiple testing. Even if one could estimate the variances in the denominator of the t-statistics accurately, running tests on all the genes in the dataset simultaneously will inevitably inflate the the family-wise type I error rate (FWER). Recall that the type I error in hypothesis testing is the error of rejecting the null hypothesis when it is actually true. With a single test we can control the type I error by setting the significance level of the test. In the context of our problem for a single gene we test the null hypothesis that the gene is non-DE under the treatment condition and we can use the conventional value 0.05 for the significance level of the test, α. However, the upper bound for FWER increases quickly as the number of tests increases. To see this note that if we test n genes independently at the α level, then

$$\begin{aligned} & P(\text{Making a type I error in one test}) \leq \alpha \\ \Longrightarrow\ & P(\text{Not making a type I error in a test}) \geq 1-\alpha \\ \Longrightarrow\ & P(\text{Not making a type I error in n tests}) \geq (1-\alpha)^n \\ \Longrightarrow\ & P(\text{Making at least one type I error in n tests}) \leq 1-(1-\alpha)^n \qquad (2.4) \end{aligned}$$

Thus, the upper bound for the FWER grows exponentially with the number of tests. For instance, for 100 genes and with $\alpha = 0.05$ the chance of mistakenly identifying at least one non-DE gene as DE could be as large as $1-(1-0.05)^{100} \approx 99.4\%$; thus, there is little control on the FWER when n is large.

Numerous methods have been proposed to control for the FWER for multiple testing problems. Instead of following the traditional procedure for hypothesis testing and rejecting the null hypothesis when the p-value of the test is smaller than α, these methods defined an adjusted p-value and use it in place of the original p-value.

Controling for Family-wise Error Rate

From (2.4) one can see that if we replace the original p-value for gene i, $p_i, i = 1, \ldots, n$ with the adjusted p-value, $\tilde{p_i} = 1-(1-p_i)^n$, then using the significance level α on the adjusted p-values will keep the FWER under α; this adjustment was proposed by Šidák [50]. The Šidák p-value adjustment is a single step procedure [14] since it performs the same form of adjustment on the p-values for all the hypotheses regardless of the ordering of the unadjusted p-values.

Another well-known single step p-value adjustment procedure for multiple testing is the Bonferroni procedure, which replaces the original p-value p_i with the adjusted p-value $\tilde{p_i} = \min\{1, np_i\}$. Note that with this adjustment and again using α as the significance level

$$\begin{aligned} \text{FWER} &= P(\text{Making at least one type I error in n tests}) \\ &\leq \sum_{i=1}^{n} P(\text{Making a type I error on test i}) \\ &\leq \sum_{i=1}^{n} \frac{\alpha}{n} = \alpha \end{aligned}$$

Although it is still widely being used, the Bonferroni procedure is one of the most conservative procedures for controlling FWER in multiple testing problems. In fact, in general even though single step adjusted p-values are simple to calculate, they tend to result in low power for the tests because they are too conservative [14]. Another disadvantage of using single step p-value adjustment procedures in gene expression analyses is that some of these procedures, such as the Šidák [50] procedure, assumes independence between the test statistics; however, in gene expression studies dependence exists within groups of genes because of the co-regulation among them [14]. Methods for controlling FWER with adjusted p-values without the assumption that the genes are independent have been developed; e.g., see Dudoit et al [14].

Controlling for False Discovery Rate

As mentioned before Bonferroni and Šidák are two conservative methods that often result in low power of the tests. Another way to control the error rate in multiple testing problems is to control for the false discovery rate (FDR):

$$\mathrm{FDR} = \mathbb{E}\left(\frac{\#\text{tests being wrongly called positive}}{\#\text{tests being called positive}}\right)$$

Controlling FDR rather than FWER has its practical attractiveness since biologists are often willing to tolerate some amount of error in the list of genes statisticians provide for investigation, and having a few positive is more preferable than not detecting the positives at all.

Benjamini and Hochberg [4] first proposed the notion of false discovery rate in 1995 and provided an algorithm to select the tests to reject; the goal of the algorithm is to limit the FDR to a user-defined level. To illustrate the algorithm let's suppose that we would like to limit the FDR to α. The algorithm first sorts the p-values of the tests in an increasing order: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(n)}$. Then, for each ordered p-value, $p_{(i)}$ the algorithm checks if the inequality $\{p_{(i)} \leq \frac{\alpha}{n} i\}$ is satisfied. The cutoff for the tests is the largest p-value that satisfies the inequality, so all the tests with p-values less than or equal to the cutoff will be rejected.

Besides its practical appeal another benefit of the Benjamini-Hochberg algorithm is that it can be modified to control the FDR even when the tests are dependent [5]. Let $H_0 = \cap H_0^i$, where H_0^i is the null hypothesis that gene i, is non-DE and $i = 1, 2, \ldots, n$. Thus, H_0 is the null hypothesis that all the genes are non-DE.

Assume that the $\{H_0^i; i = 1, 2, \ldots, n\}$ are independent. One can estimate the distribution of the t-statistics under H_0 by permuting the control and treatment labels for all the genes [24]; if the dataset is structured in such a way that the rows correspond to the genes and the columns correspond to the replicates, this means permuting the columns of the dataset. Note that permuting the columns preserves the biological dependence between the genes.

For any given cutoff value for the t-statistics (calculated with the gene expression dataset), one can use the permutation distribution of the t-statistics to estimate the FDR for the case of rejecting all tests with t-statistics smaller or equal to the cutoff [24]. It can be shown that this procedure involving the permutation distribution of the t-statistics is equivalent to the Benjamini -Hochberg algorithm [24] and that the FDR estimated with the permutation procedure is a consistent estimator of the FDR. For more detailed information on FDR please see Benjamini et al [4, 5] and Storey et al [51, 52].

Gene Set Enrichment

Besides the methods that assess genes individually and then compare the p-values of the statistics across all genes to obtain a gene rank list estimate, there are also methods working with gene sets rather than individual genes. There are several reasons to consider the differential expression of sets of genes rather than individual genes themselves. Examining sets of genes allows researchers to analyze changes of the level of a biological pathway and this approach may have more relevance than single-gene analysis for complex diseases such as cancer and diabetes. Grouping genes into sets decreases the number of hypothesis tests and lessens the problem that arises for multiple testing. Furthermore, an analysis of gene sets rather than genes may have more statistical power under certain circumstances. [1]

Gene Set Enrichment Analysis

Gene set enrichment analysis (GSEA) is a method that determines the differential expression of sets of genes between two phenotypes using gene expression data. The sets of genes are predetermined from existing biological literature and are meant to contain genes that are related by biological function, chromosomal location, or regulation [53].

Although GSEA is the most widely-used method of its kind, many others exist for the purpose of finding differentially expressed gene sets. Among methods whose first step is to assign a scoring statistic at the individual gene level, the procedures roughly fall under the same general framework: first, individual gene-level statistic are computed for each gene in the dataset, then an optional transformation of this gene-level statistics is made in the event that one wishes to account for both up- and down-regulation or to improve the robustness of the statistic. One then uses the gene-level statistics for a given gene set to compute the gene set statistic for that set. Finally, the significance of the gene set is then determined by using a permutation procedure (that permutes either the gene labels or the phenotype labels depending on the assumption made in the null hypothesis) [1]. GSEA suggested several options for the gene-level statistics–most commonly used is the signal-to-noise ratio– and the suggested gene set level statistic is the Kolmogorov-Smirnov statistic, followed by a permutation test to determine significance [53].

Other Methods

In addition to GSEA proposed by Subramanian et al, several other efforts developed similar methods contemporaneously. One such method is called Significance Analysis of Function and Expression (SAFE). In this method, the Welch t-statistic was chosen as the measure of gene-level differential expression and the Wilcoxon rank sum as the gene-set level statistic due to the unknown correlation structure between gene expression levels [3]. Another method incorporates linear regression into GSEA in order to adjust for known explanatory covariates, such as chromosomal rearrangement status in certain cancers, to identify influential samples, or to evaluate model fit [45]. A third method proposed by Tian et al was developed to analyze GSEA's sensitivity to the gene set size and the influence of the gene sets not under consideration. This method was also designed to account for the correlation structure within and between gene sets [56].

Other researchers have sought to improve GSEA. Parametric Analysis of Gene Set Enrichment (PAGE) seeks to improve the power of GSEA by assuming a normal distribution and calculating a Z-score as the gene set statistic. Although they claim to be able to detect more significant gene sets with smaller p-values than GSEA, no mention is made of a correction for multiple hypothesis testing [31]. Jiang and Gentleman point out that overlap between gene sets can make it difficult to determine which set is responsible for the differential expression. In such cases, they suggest dividing the sets into their shared and unique components and analyzing the these groups separately, since genes that are shared among different biological pathways may be regulated differently than genes that are regulated independently. Furthermore, they advocate using principle component analysis to determine which gene sets can be reduced to two or three dimensions, and they claimed that genes in those sets are likely to be co-regulated [29].

Discussion

In summary, GSEA and methods alike incorporate the biological interpretations of the gene relationship in the analysis. Grouping genes into sets and analyzing the sets rather than individual genes also help lessen the problem of multiple testing. However, just as it is difficult to determine on how to divide genes into appropriate groups to get a better estimate of the variance, it is also challenging to accurately define gene sets for the analysis and GSEA is extremely dependent on the scope, composition, and accuracy of the gene sets used in the analysis. For more information see [1].

2.2 Multiple List Approaches: Rank-Based List Aggregation

In this section we will review some methods that take a completely different approach than the ones we have described so far. As there have been more datasets produced by different platforms and labs becoming available, methods to aggregate gene expression datasets measured under the same set of biological or experimental conditions have become popular [60, 12, 37].

The scale-invariant property of rank statistics makes them advantageous over the statistics used in single-list approaches since datasets from different sources are not always directly comparable. In addition, rank-based methods are more robust in general since ranks are less affected by outliers than other test statistics such as the t-test statistics.

The multiple testing problem that we discussed previously can also be improved by comparing the ranking results across lists. One of the early attempts of combining multiple DGE analyses to identify DE genes was made by Rhodes et al. Rhodes et al [47, 48] proposed using the Benjamini and Hochberg algorithm that we discussed in Section 2.1 to calculate the adjusted p-values for the genes in each list, and then defining the significant genes as the ones with small adjusted p-values in at least J lists, where the value of J was chosen by permutation testing. The procedure proposed by Rhodes can be viewed as using a two-stage filtering process to keep the false positive error rate low. The first round of the filtering was done by using the Benjamini and Hochberg algorithm; this step controls the FDR within each list. The second round of the filtering was done by checking the results across lists to remove genes that land on the top ranks of a particular list just by random chance. The advantage of combining individual list results is apparent in this example; having the second filtering helps to improve the power without compromising the sensitivity of the test: since one can check the result of one list against that of other lists to alleviate the multiple testing problem and removing the false positives in the top list allows more room for the the true positives.

We will next review a few methods that are popular and representative in rank-based gene expression analysis aggregation applications. The review is not meant to be exhaustive but to lay the groundwork for the motivations of our method.

Borda and Spearman Footrule for Complete Lists

Borda count is one of the most well-known algorithms used to aggregate preference rank lists [37, 16]. The method was devised in 1770 initially as a voting system.

We will describe the Borda classifier in the case where complete lists are available; i.e., in the case where all the lists are ranking the same set of items and there are no missing ranks. Borda suggested giving each item on a particular list a score that is the number of items ranked below this item on the list (this means that items that are ranked high on an individual list will receive a high score); then, to aggregate the lists Borda suggested taking the sum of the scores across the lists; finally, the items are then sorted in a decreasing order according to the sums [6]. In mathematical notations and in the context of gene ranking, supposed there are L rank lists and N genes. For each list $j = 1, \ldots, L$ and for gene $i = 1, \ldots, N$, the Borda total score is

$$B(i) = \Sigma_{j=1}^{L}(N - r_j(i))$$

where $r_j(i)$ is the rank for item i on list j. Then, to get the aggregated rank list the genes will be ranked starting from the largest Border total score.

Note that for complete lists Borda's method is equivalent to ranking according to the averages of the ranks for each gene across the lists (with the smallest average first).

Another example where a seemingly more complicated procedure is equivalent to a simple algorithm is the Spearman Footrule [46]. The Spearman Footrule (SF) is a distance metric measuring how different two rank lists are. According to the SF procedure the optimal aggregated rank list is the list that minimizes the SF distance between the aggregated rank list and the individual lists. For complete lists it can be shown that the SF procedure is equivalent to ranking according to the median of the ranks for each item across the lists [60].

The attractiveness of Borda's total score and the SF procedure is that they are intuitive and computationally easy as they can be computed in linear time.

MC Algorithms for Complete Lists

Another heuristic way to combine multiple rank lists is to use Markov Chain (MC) algorithms. The use of Markov Chain algorithms for aggregating rank lists received the attention from the community when it was first proposed by Dwork to combine ranking results from multiple internet search engines [16]. The algorithm mimics the procedure when a ranker is continuously given pairs of items to compare.

In the context of gene expression data each gene is being represented by a state in the MC algorithm so for a total of n genes the size of the transition matrix should be $n \times n$. Heuristically, the (i, j) entry on the transition matrix represents the estimated weight that gene j would be ranked ahead of gene i; i.e., gene j would be more likely to be DE compared to gene i. After all the (i, j) entries are filled, an adjustment is then made to the entries so that all entries will be positive; the purpose of this technical step is just to make the MC irreducible so that there exits a unique stationary distribution for the MC. The genes are then ranked according to the stationary distribution with the gene with the highest stationary probability being the first.

Let M be the transition matrix used for the MC algorithm. Some suggestions to assign the values of the weights have been made by DeConde et al and Dwork et al [16, 12]:

- *Majority rule*: $M(i, j) = 1$ If gene j is ranked ahead of gene i at least 50% of the times among the lists; otherwise, $M(i, j) = 0$;

- *Frequency rule*: $M(i, j) =$ the fraction of times gene j is ranked ahead of gene i among the lists;

- *Minority/Specialist protection rule*: $M(i, j) = 1$ If gene j is ranked ahead of gene i at least once among the lists; otherwise, $M(i, j) = 0$.

After assigning the weights, the entries on each row of the matrix M will be rescaled and adjusted slightly so that the entries on the row will sum up to one and all the entries will be positive. The adjustment to make all the entries positive is necessary to ensure that the MC is irreducible.

Dwork et al mentioned that solving for the stationary distribution directly is computationally intensive [16] and suggested using simulations to improve the efficiency of the algorithm.

Borda and MC Algorithms for Incomplete Lists

Modifications can be made to the rank-based aggregation algorithms that we have discussed so far to make them adaptable to incomplete lists (i.e., lists with at least one list that misses at least one of the gene that is ranked by another list) [12, 37].

One suggestion is to add a preprocessing step to treat the missing values of the ranks for the Borda algorithm [37]. The procedure of this step varies depending on the cause for the missing ranks. For list j, if the reason for the missing ranks is because we only have the ranks for the top-k genes from the experiment that produced list j (i.e., the experiment originally did rank the genes that are now with missing ranks in list j; however, we do not know what ranks exactly the experiment assigned to these genes but we know that the ranks for these genes were lower than k in the experiment; this case is common if one gets the unaggregated ranks from publications that report only the top-ranked results), then Lin suggested assigning the rank $k+1$ to all the genes with missing values on list j. On the other hand, if the reason for the missing data is that the gene was never being studied in the experiment that produced list j (this is common when combining data from different platforms) then it was suggested to assign an NA to the rank value. After preprocessing the rank lists one can then proceed with the Borda algorithm described in Section 2.2.

For the MC algorithms the value 0.05 is assigned as the weights for the genes with missing ranks. This is an intermediate value between the maximum and the minimum weights defined in the MC algorithms for complete lists and the choice of the value 0.05 is somewhat arbitrary.

Attempt to Improve Data Quality with Truncated Lists

It is suggested that one could use the value $k+1$ to replace the ranks for all the genes ranked below k if one believes that the ranks for these genes are not reliable [37]. While this seems to be a reasonable approach it is practically challenging since it is difficult to decide on the value of k without knowing the distribution of the gene expressions. We will show with simulations that the quality of the algorithm is dependent on the choice of the k value.

Discussion

In summary there are numerous advantages of using rank-based methods to aggregate results from multiple studies:

- Rank statistics are more robust in general and less sensitive to outliers;
- Rank statistics are scale-invariant (with respect to the distribution of the data that are used to generate the ranks) and therefore are excellent choice for combining datasets that are not always directly comparable;
- The methods rely on few or no assumptions about the underlying distribution of the gene expressions;
- Comparing results across lists lessen the multiple testing problem and as the number of the lists increases the statistical power of the method increases.

As a result these methods have increasingly gained in popularity. However, there are two main drawbacks about these methods:

- Many of the rank-based aggregation algorithms were constructed heuristically; this makes it difficult to theoretically analyzing the behaviors of the algorithm.
- In addition, procedures such as the MC algorithms are computationally intensive as the number of genes and the number of lists increase.

Inspired by the rank aggregation methods that we have reviewed we will propose a classifier that possess the strengths of the rank-based aggregation methods that we have discussed to a great extent. In addition, we will propose a theoretical framework that allows us to study the behavior of our classifier when the number of the genes and the number of lists go to infinity. Our method is computationally less intensive compared to the MC algorithms and has a smaller asymptotic error rate compared to Borda and other ranking methods that are based on a statistic that is a function of the individual ranks.

Chapter 3

Theoretical Analysis

3.1 General Setup of the Problem

The general setup of the problem is the following. Suppose that there are n genes and we are interested in finding out which of these n genes are DE under a set of biological conditions (treatment conditions) compared to another set of controlled conditions. In addition, suppose that J sources did experiments on the same set of n genes independently and measured the gene expressions under the same set of aforementioned control and treatment conditions. Based on the t-statistics of the gene expressions under control and treatment conditions, each of the J sources then ranked the genes according to their t-statistics. (Note that we are using ranks based on t-statistics here as an example. In general, a broad range of statistics can be used as long as they satisfy Assumption 3.5.1.)

Inspired by previous work in this area [54] where genes are ranked according to the degree of evidence against the null hypothesis that the gene is not differentially expressed, we rank genes according to the conditional probability $\mathbb{P}(\text{the gene is DE} \mid \text{test results } 1, ..., J)$.

We assume that gene expressions are measured independently on genes from two classes: DE and non-DE genes. There are J independent lists, each consisting of measurements on the same set of n genes. Let d be the proportion of the DE genes; we assume that d is fixed for all lists and remains constant as $n \to \infty$. We also assume i.i.d. relations between the lists given the class labels and with appropriate scaling.

We are focused on the case where only ranks are known to us (i.e., we do not have the measurement values that generated the ranks) and we need to determine which genes are DE based on only the ranks. This is quite common in practice. Even in the case when the raw data is available the use of different technologies in different studies results in gene expression measurements that are not directly comparable across studies.

3.2 Main Result

In Section 3.3 we define a *Peer Reinforced Reranker* (PR-Ranker) classifier, $\mathcal{C}_{\text{PR}}$, for identifying genes that are DE. Our main result establishes that our classifier is asymptotically optimal among all classifiers where only rank information is given. We let $\mathcal{L}_{\text{PR}}$ denote the expected probability that a gene is misclassified by our classifier $\mathcal{C}_{\text{PR}}$. We will eventually prove Theorem 3.2.1.

Theorem 3.2.1. *Given data subject to Assumption 3.5.1 the classifier $\mathcal{C}_{PR}$ achieves an error rate*

$$\lim_{J\to\infty}\lim_{n\to\infty}\frac{1}{J}\log(\mathcal{L}_{PR_{n,J}})=\rho$$

where n is the number of genes and J is the number of lists of genes. No other rank based estimator can achieve an error rate with a smaller value of ρ.

We will later prove a stronger result, that no other procedure which has access to the t-statistics of the differential expressions of the genes and which may use the underlying distributions of the differentially expressed and non-differentially expressed genes can achieve a better rate. While this result is asymptotic, in Chapter 4 we give simulation results showing that our classifier outperforms other methods for realistic values of n and J.

3.3 Motivation

Suppose an object was tested independently by J laboratories and from the test results we need to draw a conclusion as to whether the object is positive of certain condition; e.g., in the context of our project, a gene is tested by J independent sources and we want to determine whether the gene is DE based on the J lists of rankings. In the ideal case if $\mathbb{P}(\text{object is }+|\text{ test results }1,...,J)$ were attainable, ranking according to such probability would give us the optimal ranking result. Alternatively, note that since the tests are independent given the object

$$\begin{aligned}&\mathbb{P}(\text{object is} + |\text{test results } 1,...,J)\\ =&\frac{\mathbb{P}(\text{object is }+)\mathbb{P}(\text{test results } 1,...,J|\text{object is }+)}{\mathbb{P}(\text{test results } 1,...,J)}\\ =&\frac{\mathbb{P}(\text{object is }+)\prod_{j=1}^{J}\mathbb{P}(\text{test result } j|\text{object is }+)}{\mathbb{P}(\text{test results } 1,...,J)}\end{aligned}$$

The probability in the denominator is usually difficult to estimate without knowing the joint distribution of the test results. Instead, we rank according to $\frac{\mathbb{P}(\text{object is}+|\text{test results } 1,...,J)}{\mathbb{P}(\text{object is}-|\text{test results } 1,...,J)}$ which is equivalent to and more convenient than using $\mathbb{P}(\text{object is} + |\text{test results } 1,...,J)$. Now,

$$\begin{aligned}&\frac{\mathbb{P}(\text{object is} + |\text{test results } 1,...,J)}{\mathbb{P}(\text{object is} - |\text{test results } 1,...,J)}\\ =&\frac{\mathbb{P}(\text{object is }+)\prod_{j=1}^{J}\mathbb{P}(\text{test result } j|\text{object is }+)}{\mathbb{P}(\text{object is }-)\prod_{j=1}^{J}\mathbb{P}(\text{test result } j|\text{object is }-)}\\ =&\frac{\mathbb{P}(\text{object is }+)\prod_{j=1}^{J}\frac{\mathbb{P}(\text{object is }+|\text{test result } j)\mathbb{P}(\text{test result } j)}{\mathbb{P}(\text{object is }+)}}{\mathbb{P}(\text{object is }-)\prod_{j=1}^{J}\frac{\mathbb{P}(\text{object is }-|\text{test result } j)\mathbb{P}(\text{test result } j)}{\mathbb{P}(\text{object is }-)}}\\ =&\left(\frac{\mathbb{P}(\text{object is }-)}{\mathbb{P}(\text{object is }+)}\right)^{J-1}\prod_{j=1}^{J}\frac{\mathbb{P}(\text{object is } +|\text{test result } j)}{\mathbb{P}(\text{object is } -|\text{test result } j)}\\ =&\left(\frac{\mathbb{P}(\text{object is }-)}{\mathbb{P}(\text{object is }+)}\right)^{J-1}\prod_{j=1}^{J}\frac{\mathbb{P}(\text{object is } +|\text{test result } j)}{1-\mathbb{P}(\text{object is } +|\text{test result } j)}\end{aligned}$$

Note that in the last expression above, the quantity $\left(\frac{\mathbb{P}(\text{object is }-)}{\mathbb{P}(\text{object is }+)}\right)^{J-1}$ is fixed regardless of what the individual test results are; therefore, if all objects have the same probability of being positive, ranking according to the product of the conditional odd ratios gives us an equivalent way as ranking according to $\mathbb{P}(\text{ object is }+|\text{ test results } 1,...,J)$.

Furthermore, with the monotonic property of logarithm, ranking according to the product of the conditional odd ratios is also equivalent to ranking according to the sum of the log conditional odd ratios

$$\sum_{j=1}^{J}\log\left(\frac{\mathbb{P}(\text{object is } +|\text{test result } j)}{1-\mathbb{P}(\text{object is } +|\text{test result } j)}\right).$$

As a result, we have established that ranking according to $\mathbb{P}(\text{object is } +|$ test results $1, ..., J)$ is equivalent to ranking according to $\sum_{j=1}^{J} \log\left(\frac{\mathbb{P}(\text{object is } +|\text{test result } j)}{1-\mathbb{P}(\text{object is } +|\text{test result } j)}\right)$, and since in practice the latter is a quantity that is easier to obtain, our ranking estimator will be constructed according to this quantity.

Proposed Approach

For each gene i we let B_i be the event that it is differentially expressed and let R_i^j be its rank in list j. Our proposed method was motivated by the question "what is the best one can do to estimate $\mathbb{P}(B_i|R_i^j)$ in the case where the only information given is the ranks of the genes on each list." We propose a solution to the problem by providing an approximation to $\mathbb{P}(B_i|R_i^j)$ and ranking genes according to,

$$\sum_{j=1}^{J} \log\left(\frac{\widehat{\mathbb{P}}(B_i|R_i^j)}{1-\widehat{\mathbb{P}}(B_i|R_i^j)}\right) \tag{3.1}$$

where $\widehat{\mathbb{P}}(B_i|R_i^j)$ is our estimated probability which will be shown to converge to $\mathbb{P}(B_i|R_i^j)$ as the number of genes and the number of lists increase. Consequently, our solution converges to the optimal solution as the number of genes and number of lists increase. Because our solution is an approximation to the optimal solution, our solution will outperform Borda when the number of genes and the number of lists are large.

3.4 Notations

In terms of notations, we use tilde $\widetilde{}$ to denote quantities relating to the DE gene population and asterisk (*) to denote quantities relating to the mixture population. For example, we let T^j and $\widetilde{T}^j$ be the minus of the absolute values of the t-test statistics on list j for a gene from the non-DE and DE classes, respectively, and $\phi^j(t)$ and $\widetilde{\phi}^j(t)$ be the associated densities respectively. Furthermore, we let $F^j(t)$ and $\widetilde{F}^j(t)$ be the CDFs associated with $\phi^j(t)$ and $\widetilde{\phi}^j(t)$, respectively. For the mixture model, define $F^{*j}(t) := (1-d)F^j(t)+d\widetilde{F}^j(t)$ and $\phi^{*j}(t) := (1-d)\phi^j(t) + d\widetilde{\phi}^j(t)\ \forall\ t \leq 0$, and let T^{*j} be the associated random variable; i.e., T^{*j} is the minus of the absolute value of the t-test statistic from the mixture of two classes on list j, and F^{*j} is its associated CDF.

In addition, let R_i^j be the rank for gene i on list j; similarly, let T_i^j be the minus of the absolute values of the t-test statistics on list j for gene i.

We assume that ranking is defined in a descending order; [1] this way genes with t-statistics further away from zero will be ranked ahead of genes that are close to zero; such ranking order is consistent with the convention used in the existing literature in the area [54]. Let $(U_1^j, ..., U_n^j)$ be the ordered statistics of the $(T_1^j, ..., T_n^j)$. For example, if $(T_1^1, T_2^1, T_3^1) = (-0.3, -1.5, -0.2)$ for genes $1, 2$ and 3 on list 1 then their ranks on list 1 would be $(R_1^1, R_2^1, R_3^1) = (2, 1, 3)$ and $(U_1^1, U_2^1, U_3^1) = (-1.5, -0.3, -0.2)$.

To make the notations simpler we assume that given the class labels (i.e., DE and non-DE gene classes) the T_i^j's are i.i.d. random variables from each of the two distributions, one for the DE genes and one for the non-DE genes.

We follow the usual convention and denote the inverse CDF of F^{*j} to be $(F^{*j})^{-1}(p) = \inf\{t; F^{*j}(t) \geq p\}$ and let $F_n^{*j}(t)$ be the empirical CDF (ECDF) that associates with T_i^j; i.e., $F_n^{*j}(t) = \frac{\sum_{i=1}^n I(T_i^{*j} \leq t)}{n}, \forall\ t \in \mathbf{R}$. We let $A_r = A_r^j$ denote the event that the gene ranked r (among n genes) in list j is DE and define

$$p_n^j(r) := \mathbb{P}(A_r^j) = \mathbb{P}(B_i \mid R_i^j = r).$$

Let $i^j(r)$ be the gene index for the gene that is ranked r on list j, so $R_{i^j(r)}^j = r$. In addition, let $R_i^{-j} = (\sum_{l=1}^J R_i^l) - R_i^j$ be the aggregate ranking of gene i in all the lists *except* list j.

3.5 Proposed Classifier and Assumptions

We will say that gene i is provisionally classified as DE in list j if R_i^{-j} is among the dn smallest among all the genes. This is equivalent to being ranked in the top dn genes by Borda applied to all the lists, except list j. Let $h^j(r)$ denote the indicator that the gene with rank r on list j is provisionally classified as DE according to the aggregated ranking of all other lists other than list j,

$$\begin{aligned} h^j(r) :&= I(\text{gene ranked } r \text{ on list } j \text{ is provisionally classified as DE}) \\ &= I\left(\#\left\{i : R_i^{-j} \leq R_{i^j(r)}^{-j}\right\} \leq dn\right) \end{aligned}$$

[1] Instead of ranking the minus of the absolute values of the t-test statistics in a descending order, we can also rank the absolute values of the t-test statistics in an ascending order; however, by defining T^* be the minus of the absolute values of the t-test statistics, the empirical CDF associates with T^* will be directly proportional to the ranks of T^*. Such definition of T^* will make our notation cleaner in the later steps and this is why definite T^* this way.

We define $q_n^j(r)$, to be a smoothed version of $h^j(r)$, as

$$q_n^j(r) = \frac{\sum_{\substack{r' \in \{1:n\} \\ |r'-r| \leq \sqrt{n}}} h^j(r')}{\#\{r' \in \{1:n\} : |r'-r| \leq \sqrt{n}\}}.$$

By averaging over a range of r we can approximate the likelihood that a gene will be provisionally classified as DE based on its rank in list j. It will become clear later in the chapter that $q_n^j(r)$ is a form of density estimation for $p_n^j(r)$, the actual probability of being DE for a gene with rank r on list j. Our definition of $q_n^j(r)$ is somewhat arbitrary in terms of the window size for the smoothing and is chosen for the sake of simplicity in terms of the notations used in the proof. In practice we would suggest using a more sophisticated smoothing procedure such as LOWESS (Locally Weighted Scatterplot Smoothing).

Our classifier $\mathcal{C}_{\mathrm{PR}}$ computes

$$\sum_{j=1}^{J} \log\left(\frac{q_n^j(R_i^j)}{1 - q_n^j(R_i^j))}\right) \tag{3.2}$$

for each gene i and classifies the top dn genes as DE and the remaining $(1-d)n$ genes as non-DE. This score gives a ranking based on how much we believe a gene is DE. We will establish Theorem 3.2.1 for this estimator under the following assumption on the density of the t-statistics.

Assumption 3.5.1. *We assume that the negative absolute value of the t-statistics from each class form a continuous distribution on* $(-\infty, 0)$ *and that the distribution for the DE class has a more negative mean than that for the non-DE class.*

We also assume that for all $t \leq 0$

1. *Full support:* $\phi^j(t)$ *and* $\widetilde{\phi}^j(t)$ *are positive and continuous.*
2. *Contiguity:* $0 < c_1 \leq \frac{\phi(t)}{\widetilde{\phi}(t)} < c_2$ *for some* $c_1, c_2 > 0$*; and that* $lim_{t \to -\infty} \frac{\phi(t)}{\widetilde{\phi}(t)} = c$ *for some finite positive constant* c.
3. *Stochastic Domination: That for all* $t \leq 0$, $F(t) \leq \widetilde{F}(t)$.

In practice Assumption 3.5.1.2 says that the distributions of the absolute values of the t-statistics for the DE genes and the non-DE genes should have a reasonable overlap. The case in which $c = 0$ is trivial (i.e., in the case where the DE genes have much bigger absolute values of t-statistics overall) and does not impose much technical challenge; therefore the case $c = 0$ is not of our interest.

3.6 Proof of Theorem 3.2.1

Proof Outline

Our proof is comprised of three parts.

Part I: We observe that if we were given $p_n^j(r)$, then the probability that a gene ranked r is DE could be obtained by plugging this quantity into (3.1) and doing so will give us the ideal estimator for the probability. Section 3.6 then analyzes the behavior of our smoothed provisional classifier $q_n^j(r)$. We first note in proposition 3.6.1 that as n becomes large the function $p^{j*}(r/n)$, where

$$p^{j*}(\alpha) := \frac{d\widetilde{\phi}(F^{*-1}(\alpha))}{d\widetilde{\phi}(F^{*-1}(\alpha)) + (1-d)\phi(F^{*-1}(\alpha))},$$

will give a value that is close to $p_n^j(r)$. This motivates us to compare the asymptotic behavior of our smoothed provisional classifier to that of $p^{j*}(r/n)$. Then, Propositions 3.6.3 and 3.6.5 together show that when n and J are large, our smoothed provisional classifier $q_n^j(r)$ is close to $p^{j*}(r/n)$ with high probability.

Part II: In Section 3.6 we observe that if we were given the distributions of the negative of the absolute values of the t-statistics then for a particular gene the estimator (we will refer to this estimator as the *simplified Bayes estimator*) constructed by using the t-statistics for the gene across lists will be almost as good as the Bayes estimator constructed by using the t-statistics for *a*ll the genes across the lists.

Part III: In Section 3.6 we study another estimator ζ_i' (which is based on ranks rather than the negative of the absolute values of the t-statistics) and show that asymptotically ζ_i' behaves similarly to the simplified Bayes estimator. Then, we calculate the normalized log loss for ζ_i' and compare this loss with the loss for the ranking produced by using our smoothed provisional classifier. We then finally show that asymptotically ζ_i' and our classifier give similar loss; thus, our estimator is asymptotically optimal.

The Behavior of the Smoothed Provisional Classifier

Proposition 3.6.1. *The rank based estimator satisfies*

$$\max_r |p_n^j(r) - p^{j*}(r/n)| \to 0$$

in probability, as $n \to \infty$.

We will first give an intuitive interpretation of what Proposition 3.6.1 says. We expect the gene ranked r to have t-statistic approximately at $F^{*-1}(r/n)$. Given that a gene has an unconditional probability d of being DE, conditional on the t-statistic of the gene's expressions, t, Bayes rule implies that the probability that it is DE is $\frac{d\widetilde{\phi}(t)}{d\widetilde{\phi}(t)+(1-d)\phi(t)}$. Combining these principles motivates the definition of $p^{j*}(\alpha)$. However, to establish uniform convergence there are a number of challenges; first of all, $F^{*-1}(r/n)$ may not be concentrated for small r; secondly, there is dependence between the ranks. We defer the proof of the proposition to the appendix section.

The uniform convergence in proposition 3.6.1 is important as it ensures that for large n, the error between $p_n^j(r)$ and $p^{j*}(r/n)$ can be controlled simultaneously for all genes. In the later steps of the proof we will see that such an error bound is necessary to show that our proposed ranking method is a reliable and stable method asymptotically.

To analyze $q_n^j(r)$ we will compare it with another quantity where $p$rovisionally DE is replaced with $a$ctually DE. We define $\breve{h}_j(r) = I(B_{i^j(r)})$ the indicator that the gene ranked r on list j is in fact DE and define

$$\breve{q}_n^j(r) = \frac{\sum_{\substack{r\in\{1:n\}\\ |r-r'|\leq\sqrt{n}}} \breve{h}_j(r)}{\#\{r' \in \{1:n\} : |r-r'| \leq \sqrt{n}\}}.$$

As we establish in the following lemma, this closely approximates $p^{j*}(r/n)$.

Lemma 3.6.2. *For each list j,*

$$\max_r |\breve{q}_n^j(r) - p^{j*}(r/n)| \to 0$$

in probability as $n \to \infty$.

Proof. Without loss of generality we will treat the case for $r \leq n/2$, the case of $r > n/2$ follows similarly.

Let $N_r = \#\{r' \in \{1:n\} : |r-r'| \leq \sqrt{n}\}$ be the size of the set we are averaging over and note that $\sqrt{n} \leq N_r \leq 2\sqrt{n}+1$. Recall that A_r^j is the event that the gene ranked r on list j is DE and U_r^j is the t-statistic for the gene ranked r in list j. Let us write $\mathcal{F}_r^j$ as the smallest sigma-algebra generated by

$\{U_1^j, ..., U_r^j, I_{A_1}^j, ..., I_{A_r}^j\}$. Then,

$$\begin{aligned}\breve{q}_n^j(r) =& \frac{1}{N_r} \sum_{\ell=1\vee(r-\sqrt{n})}^{r+\sqrt{n}} I_{A_\ell} \\ =& \frac{1}{N_r} \sum_{\ell=1\vee(r-\sqrt{n})}^{r+\sqrt{n}} \left[I_{A_\ell} - \mathbb{P}(A_\ell|\mathcal{F}_{\ell-1}^j) + \mathbb{P}(A_\ell|\mathcal{F}_{\ell-1}^j)\right]\end{aligned}$$

For $k \leq r + \sqrt{n}$, define

$$M_k := \begin{cases} 0, & \text{if } k \leq 1 \vee (r - \sqrt{n}) \\ \sum_{\ell=1\vee(r-\sqrt{n})}^{k} \left[I_{A_\ell} - \mathbb{P}(A_\ell|\mathcal{F}_{\ell-1}^j)\right], & \text{otherwise.} \end{cases}$$

Note that

1. $\mathbb{E}(M_k) \leq 2\lfloor\sqrt{n}\rfloor + 1 < \infty$;
2. Since $M_k \in \mathcal{F}_k^j \ \forall n$, M_k is adapted to the filtration $\mathcal{F}_k^j$;
3. $\mathbb{E}(M_k|\mathcal{F}_{k-1}^j) = \mathbb{E}([M_{k-1} + I_{A_k} - \mathbb{P}(A_k|\mathcal{F}_{k-1})]|\mathcal{F}_{k-1}^j)$
   $= M_{k-1} + \mathbb{E}(I_{A_k}|\mathcal{F}_{k-1}^j) - \mathbb{P}(A_k|\mathcal{F}_{k-1}^j) = M_{k-1}$.

Thus, M_k is a martingale with respect to $\mathcal{F}_k^j$. Moreover, note that $|M_k - M_{k-1}|$ is uniformly bounded by 1. Thus, for any $\epsilon > 0$ by the Azuma-Hoeffding inequality

$$\begin{aligned}& \mathbb{P}\left(\left|\frac{1}{N_r} \sum_{\ell=1\vee(r-\sqrt{n})}^{r+\sqrt{n}} \left[I_{A_\ell} - \mathbb{P}(A_\ell|\mathcal{F}_{\ell-1}^j)\right]\right| > \epsilon\right) \\ =& \mathbb{P}\left(\left|\frac{1}{N_r} M_{r+\sqrt{n}}\right| > \epsilon\right) \\ \leq& \mathbb{P}(|M_{r+\sqrt{n}} - M_{(r-\sqrt{n}-1)\vee 0}| > \sqrt{n}\epsilon) \\ \leq& 2\exp\left(-\frac{n\epsilon^2}{2(2\sqrt{n}+1)}\right) = o(1/n).\end{aligned}$$

Taking a union bound we have shown that

$$\max_{r\leq n/2} \frac{1}{N_r} \Big| \sum_{\ell=1\vee(r-\sqrt{n})}^{r+\sqrt{n}} \left[I_{A_\ell} - \mathbb{P}(A_\ell|\mathcal{F}_{\ell-1}^j)\right] \Big| \to 0,$$

in probability as $n \to \infty$. Thus it suffices to prove that

$$\max_{r \le n/2} \frac{1}{N_r} \Big| \sum_{\ell=1 \vee (r-\sqrt{n})}^{r+\sqrt{n}} \mathbb{P}(A_\ell | \mathcal{F}_{\ell-1}^j) - p^{j*}(r/n) \Big| \to 0 \tag{3.3}$$

in probability as $n \to \infty$ which follows as a consequence of equation (A.1). This completes the proof of the lemma.

□

Let H_n be the ECDF of $\frac{R_\nu}{n}$ for all indices ν's for genes from the non-DE class on list j and similarly, let $\widetilde{H}_n$ be the ECDF of $\frac{R_{\widetilde{\nu}}}{n}$ for all indices $\widetilde{\nu}$'s for genes from the DE class.

Note that $H_n(x) = F_n(F_n^{*-1}(x))$. Thus, for a gene ranked r among all the genes, $H_n(r/n)$ gives its normalized rank among the non-DE genes. By almost surely uniform convergence of ECDF to the true CDF and by almost surely uniform convergence of empirical quantile to the true quantile we have

$$H_n(x) \overset{a.s.}{\to} H(x) := F(F^{*-1}(x)) \; uniformly \; \forall \; 0 \le x \le 1 \; as \; n \to \infty.$$

Similarly,

$$\widetilde{H}_n(x) \overset{a.s.}{\to} \widetilde{H}(x) := \widetilde{F}(F^{*-1}(x)) \; uniformly \; \forall \; 0 \le x \le 1 \; as \; n \to \infty.$$

Define $H_n^{(-j)}(x)$ to be the ECDF for $\frac{1}{n} R_\nu^{-j}$ for $\nu \in$ {index for non-DE genes}, the aggregated ranks obtained by summing up the normalized rankings of each of the non-DE genes across all J lists, except list j. Let $H^{(-j)}$ be the CDF of the sum of $(J-1)$ i.i.d. random variables, each with CDF $H(x)$. Then, $H_n^{(-j)}$ converges almost surely pointwisely to $H^{(-j)}$. Similarly, $\widetilde{H}_n^{(-j)}$, the counterpart of $H_n^{(-j)}(x)$ for DE genes, converges almost surely pointwisely to $\widetilde{H}^{(-j)}$, the CDF of the sum of $(J-1)$ random variables, each with CDF $\widetilde{H}(x)$.

Define

$$H^{(-j)*}(x) = (1-d)H^{(-j)}(x) + d\widetilde{H}^{(-j)}(x), \; \forall \; 0 \le x \le 1.$$

With this notation we can define the limiting behavior of q_n^j. We define

$$q^{*j,J}(\alpha) := H^{(-j)}((H^{(-j)*})^{-1}(d))(1 - p^{j*}(\alpha)) + \widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d))p^{j*}(\alpha).$$

In this equation $H^{(-j)}((H^{(-j)*})^{-1}(d))$ represents the average fraction of non-DE genes that are classified as DE (false positive rate) by our classifier, while $\widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d))$ represents the fraction of DE genes correctly classified as DE (true positive rate).

Proposition 3.6.3. *For each j,*

$$\max_r |q_n^j(r) - q^{*j,J}(r/n)| \to 0$$

in probability as $n \to \infty$.

For a fixed list j, let Γ denote the total number of non-DE genes that are provisionally classified as DE (i.e., total number of false positives) and let $\widetilde{\Gamma}$ denote the total number of DE genes ranked that are provisionally classified as DE (i.e., total number of true positives). Then by almost surely uniform convergence of ECDF to the true CDF and almost surely uniform convergence of empirical quantiles to the distribution quantiles we have

$$\frac{\Gamma}{n(1-d)} = H_n^{(-j)}((H_n^{(-j)*})^{-1}(d)) \overset{a.s.}{\to} H^{(-j)}((H^{(-j)*})^{-1}(d)) \tag{3.4}$$

and

$$\frac{\widetilde{\Gamma}}{nd} = \widetilde{H}_n^{(-j)}((H_n^{(-j)*})^{-1}(d)) \overset{a.s.}{\to} \widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d)) \tag{3.5}$$

as $n \to \infty$.

Conditional on being DE (respectively non-DE), every gene is equally likely to be classified DE given the ranking from list j. Hence if we condition on $\breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}$ we have that the conditional distribution $(q_n^j(r) | \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma})$ is given by $\frac{1}{N_r}(W_1 + W_2)$ where $N_r = \#\{r' \in \{1:n\} : |r' - r| \leq \sqrt{n}\}$ is the length of the window of genes used to estimate $q_n^j(r)$ and

$$W_1 \sim \text{hypergeometric}((1-d)n, \Gamma, N_r(1 - \breve{q}_n^j(r)))$$

and

$$W_2 \sim \text{hypergeometric}(dn, \widetilde{\Gamma}, N_r \breve{q}_n^j(r)).$$

The sum $W_1 + W_2$ is the total number of genes that we would provisionally classify as DE among the sample of N_r genes. In particular, W_1 is the number of false positive and W_2 is the number of true positive in the sample. We can think of this as if we divide the population of genes into two classes: $n(1-d)$ non-DE genes and nd DE genes, and we also divide our sample into two sub-samples: we first take a sample of size $N_r(1 - \breve{q}_n^j(r))$ from the $(1-d)n$ non-DE genes among which $\frac{\Gamma}{n(1-d)}$ portion of them are misclassified as DE; W_1 is the number of genes being misclassified as DE in our sample. Then, we take another sample of size $N_r \breve{q}_n^j(r)$ from the nd DE genes among which $\frac{\widetilde{\Gamma}}{nd}$ portion of them are correctly classified as DE; W_2 is the number of genes being correctly classified as DE in this sample. We will control W_1, W_2 through the following claim.

Claim 3.6.4. *For all r,*

$$\mathbb{P}\left(\left|\frac{W_1}{N_r} - \frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r))\right| > \epsilon \mid \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\right) \leq 2\exp\left(-\frac{N_r\epsilon^2}{2}\right)$$

and

$$\mathbb{P}\left(\left|\frac{W_2}{N_r} - \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r)\right| > \epsilon\right) \leq 2\exp\left(-\frac{N_r\epsilon^2}{2}\right)$$

.

We will show this for W_2, the case of W_1 will follow similarly. Let $\mathcal{S}_k$ be the σ-field generated by $\{A^j_{r-\lfloor\sqrt{n}\rfloor}, ..., A^j_k, \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\}$ for $k \in \{r-\lfloor\sqrt{n}\rfloor, ..., r+\lfloor\sqrt{n}\rfloor\}$ and let $\mathcal{S}_{r-\lfloor\sqrt{n}\rfloor-1}$ be the set $\{\breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\}$. For $k \in \{r-\lfloor\sqrt{n}\rfloor, ..., r+\lfloor\sqrt{n}\rfloor\}$. Define X_k as

$$X_k := \mathbb{E}(W_2|\mathcal{S}_k) = \begin{cases} \mathbb{E}(W_2 \mid \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}) = \frac{\widetilde{\Gamma}}{nd}N_r\breve{q}^j(r), \text{ if } k = r-\lfloor\sqrt{n}\rfloor-1; \\ \mathbb{E}(W_2|\mathcal{S}_k), \text{ if } \lceil r\rceil - \lfloor\sqrt{n}\rfloor \leq k \leq r+\lfloor\sqrt{n}\rfloor-1 \\ W_2 \text{ if } k = r+\lfloor\sqrt{n}\rfloor. \end{cases}$$

By construction X_k is a martingale with respect to $\mathcal{S}_k$ with bounded increments $|X_k - X_{k-1}| \leq 1$. Hence by the Azuma-Hoeffding inequality

$$\begin{aligned} &\mathbb{P}\left(\left|\frac{1}{N_r}\left(W_2 - \frac{\widetilde{\Gamma}}{nd}N_r\breve{q}_n^j(r)\right)\right| > \epsilon \mid \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\right) \\ &\quad = \mathbb{P}\left(\left|X_{r+\lfloor\sqrt{n}\rfloor} - X_{r-\lfloor\sqrt{n}\rfloor-1}\right| > N_r\epsilon \mid \breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\right) \\ &\quad \leq \quad 2\exp\left(-\frac{N_r\epsilon^2}{2}\right) = o(1/n). \end{aligned} \tag{3.6}$$

This completes the proof of the claim.

Now for the proof of the proposition, note that

$$
\begin{aligned}
&\mathbb{P}(\max_r \left|q_n^j(r) - q^{*j,J}(r/n)\right| > \epsilon) \\
&\quad \leq \mathbb{P}\left(\max_r \left|\frac{W_1(r)}{N_r} - \frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r))\right| > \frac{\epsilon}{3}\right) \\
&\qquad + \mathbb{P}\left(\max_r \left|\frac{W_2(r)}{N_r} - \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r)\right| > \frac{\epsilon}{3}\right) \\
&\qquad + \mathbb{P}\left(\max_r \left|\frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r)) + \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r) - q^{*j,J}(r/n)\right| > \frac{\epsilon}{3}\right) \\
&\quad \leq \sum_r \mathbb{E}\mathbb{P}\left(\left|\frac{W_1(r)}{N_r} - \frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r))\right| > \frac{\epsilon}{3}\middle|\breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\right) \\
&\qquad + \sum_r \mathbb{E}\mathbb{P}\left(\left|\frac{W_2(r)}{N_r} - \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r)\right| > \frac{\epsilon}{3}\middle|\breve{q}_n^j(r), \Gamma, \widetilde{\Gamma}\right) \\
&\qquad + \mathbb{P}\left(\max_r \left|\frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r)) + \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r) - q^{*j,J}(r/n)\right| > \frac{\epsilon}{3}\right).
\end{aligned}
$$

By Claim 3.6.4 and a union bound the first two terms in the sum are $o(1)$. For the final term,

$$
\begin{aligned}
&\mathbb{P}\Big(\max_r \left|\frac{\Gamma}{n(1-d)}(1-\breve{q}_n^j(r)) + \frac{\widetilde{\Gamma}}{nd}\breve{q}_n^j(r) - q^{*j,J}(r/n)\right| > \frac{\epsilon}{3}\Big) \\
&\quad \leq o(1) + \mathbb{P}\Big(\max_r \Big|H_n^{(-j)}((H_n^{(-j)*})^{-1}(d))(1-\breve{q}_n^j(r)) \\
&\qquad\qquad + \widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d))\breve{q}_n^j(r) - q^{*j,J}(r/n)\Big| > \frac{\epsilon}{6}\Big) \\
&\quad \to 0
\end{aligned}
$$

as $n \to \infty$ where the first term $o(1)$ follows from equations (3.4) and (3.5), and triangle inequality together with the result of a union bound.

The final limit follows by Lemma 3.6.2. Combining the above estimates we have that

$$\mathbb{P}(\max_r \left|q_n^j(r) - q^{*j,J}(r/n)\right| > \epsilon) \to 0$$

which completes the proof.

Proposition 3.6.5. *The function $q^{*j,J}(\alpha)$ converge uniformly to p^{j*} as $J \to \infty$, that is*

$$\lim_{J\to\infty} \sup_\alpha |q^{*j,J}(\alpha) - p^{j*}(\alpha)| = 0$$

By Proposition 3.6.3 and the definition of $q^{*j,J}$ it suffices to prove that as $J \to \infty$,

$$\widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d)) \to 1, \tag{3.7}$$

$$H_n^{(-j)}((H_n^{(-j)*})^{-1}(d)) \to 0 \tag{3.8}$$

as $n \to \infty$.

Let μ and $\widetilde{\mu}$ be the means of the distributions $H(x)$ and $\widetilde{H}(x)$ respectively, the limiting distributions of the ranks of the non-DE and DE genes. By the stochastic domination assumption in Assumption 3.5.1 we have that $\widetilde{\mu} < \mu$. Define γ as the average $\gamma := \frac{\mu+\widetilde{\mu}}{2}$, so we have that $\widetilde{\mu} < \gamma < \mu$.

Since the distributions $H^{(-j)}$ and $\widetilde{H}^{(-j)}$ are for the sum of $J-1$ independent copies of the normalized ranks, by the Central Limit Theorem we have that $H^{(-j)}(\gamma(J-1)) \to 0$ and $\widetilde{H}^{(-j)}(\gamma(J-1)) \to 1$ as $J \to \infty$. This in term implies that

$$H^{(-j)*}(\gamma(J-1)) = (1-d)H^{-j}(\gamma(J-1)) + d\widetilde{H}^{-j}(\gamma(J-1)) \to d$$

as $J \to \infty$. Now let u_J be the quantity such that $H^{(-j)*}(u_J) = d$. Then,

$$\widetilde{H}^{(-j)}(u_J) = \widetilde{H}^{(-j)}(\gamma(J-1)) + \left[\widetilde{H}^{(-j)}(u_J) - \widetilde{H}^{(-j)}(\gamma(J-1))\right]$$

Since $\widetilde{H}^{(-j)}(\gamma(J-1)) \to 1$, we will establish (3.7) by showing that $|\widetilde{H}^{(-j)}(u_J) - \widetilde{H}^{(-j)}(\gamma(J-1))| \to 0$ as $J \to \infty$. We have that

$$\begin{aligned}
&|\widetilde{H}^{(-j)}(u_J) - \widetilde{H}^{(-j)}(\gamma(J-1))| \\
&\quad = \frac{1}{d}|d\widetilde{H}^{(-j)}(u_J) - d\widetilde{H}^{(-j)}(\gamma(J-1))| \\
&\quad \leq \frac{1}{d}\Big|d\widetilde{H}^{(-j)}(u_J) + (1-d)H^{(-j)}(u_J) \\
&\qquad\quad - d\widetilde{H}^{(-j)}(\gamma(J-1)) - (1-d)H^{(-j)}(\gamma(J-1))\Big| \\
&\quad = \frac{1}{d}|H^{(-j)*}(u_J) - H^{(-j)*}(\gamma(J-1))| = \frac{1}{d}|d - H^{(-j)*}(\gamma(J-1))| \to 0
\end{aligned}$$

as $J \to \infty$, where the inequality follows from the fact that $(d\widetilde{H}^{(-j)}(u_J) - d\widetilde{H}^{(-j)}(\gamma(J-1)))$ and $((1-d)H^{(-j)}(u_J) - (1-d)H^{(-j)}(\gamma(J-1)))$ always have the same sign. Hence $\widetilde{H}^{(-j)}((H^{(-j)*})^{-1}(d)) \to 1$ establishing equation (3.7). Equation (3.8) follows similarly. This completes the proof of the lemma.

Optimal Unrestricted Inference

In order to establish the asymptotic optimality of our rank based estimator we will consider the performance of a Bayesian estimator in the case where the parameters of the model are known (i.e., the distribution of $F(t), \widetilde{F}(t)$ are given) and where all the t-statistics of all the lists are given. Let $\mathcal{G}_i$ denote the σ-algebra generated by $\{T_i^j\}_{j=1\dots,J}$, the t-statistics for gene i and let $\mathcal{G}$ denote the σ-algebra generated by all the t-statistics $\{\mathcal{G}_i\}_{i=1,\dots,n}$. By Bayes rule the conditional probability that gene i is DE given $\mathcal{G}_i$ is

$$\xi_i := \mathbb{P}[B_i \mid \mathcal{G}_i] = \frac{d\prod_{j=1}^J \widetilde{\phi}(T_i^j)}{d\prod_{j=1}^J \widetilde{\phi}(T_i^j) + (1-d)\prod_{j=1}^J \phi(T_i^j)}. \tag{3.9}$$

In the following lemma we show that the conditional probability above is asymptotically almost identical to that when we condition on the full set of t-statistics.

Lemma 3.6.6. *For each i,*

$$\mathbb{E}|\mathbb{P}[B_i \mid \mathcal{G}] - \mathbb{P}[B_i \mid \mathcal{G}_i]| \to 0 \tag{3.10}$$

as $n \to \infty$.

We defer the proof of Lemma 3.6.6 to the appendix. Let $\mathcal{A}$ be the set of genes i with the dn largest values of $\mathbb{P}[B_i \mid \mathcal{G}]$. The optimal selection of dn genes is then $\mathcal{A}$ and the probability that a gene is misclassified is

$$\mathcal{L}_{\text{Bayes},n,J} := \mathbb{P}[\text{gene misclassified}] = \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}} \mathbb{P}[B_i^c \mid \mathcal{G}] + \frac{1}{n}\sum_{i\in\mathcal{A}^c} \mathbb{P}[B_i \mid \mathcal{G}]\Big).$$

This is the smallest misclassification rate of any estimator.

It is, however, simpler to rank genes according to ξ and with this in mind we let $\mathcal{A}'$ be the set of genes with the dn largest values of ξ_i. This simplified Bayes estimator has classification error

$$\mathcal{L}_{\xi,n,J} = \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}'} \mathbb{P}[B_i^c \mid \mathcal{G}_i] + \frac{1}{n}\sum_{i\in\mathcal{A}'^c} \mathbb{P}[B_i \mid \mathcal{G}_i]\Big).$$

By optimality of the full Bayesian classifier we of course have that $\mathcal{L}_{\text{Bayes},n,J} \leq \mathcal{L}_{\xi,n,J}$. In the other direction

$$\begin{aligned}
\mathcal{L}_{\text{Bayes},n,J} &= \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}}\mathbb{P}[B_i^c \mid \mathcal{G}] + \frac{1}{n}\sum_{i\in\mathcal{A}^c}\mathbb{P}[B_i \mid \mathcal{G}]\Big) \\
&\geq o(1) + \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}}(1-\xi_i) + \frac{1}{n}\sum_{i\in\mathcal{A}^c}\xi_i\Big) \\
&\geq o(1) + \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}'}(1-\xi_i) + \frac{1}{n}\sum_{i\in\mathcal{A}'^c}\xi_i\Big) \\
&= o(1) + \mathbb{E}\Big(\frac{1}{n}\sum_{i\in\mathcal{A}'}\mathbb{P}[B_i^c \mid \mathcal{G}_i] + \frac{1}{n}\sum_{i\in\mathcal{A}'^c}\mathbb{P}[B_i \mid \mathcal{G}_i]\Big) \\
&= o(1) + \mathcal{L}_{\xi,n,J} \qquad (3.11)
\end{aligned}$$

where the first inequalities follow by Lemma 3.6.6 and the second inequality follows by the definition of $\mathcal{A}'$ as the set of dn genes with the largest values of ξ_i. Thus

$$|\mathcal{L}_{\text{Bayes},n,J} - \mathcal{L}_{\xi,n,J}| = o(1), \qquad (3.12)$$

so, as $n \to \infty$, the simplified Bayesian classification is essentially as good. Now conditional on the $\{B_i\}$ the ξ_i are conditionally independent and so the ECDF of the ξ_i converges almost surely to $\Xi(x)$ the CDF of ξ_i. Then

$$\lim_n \mathcal{L}_{\xi,n,J} = \int_0^{\Xi^{-1}(1-d)} x d\Xi(x) + \int_{\Xi^{-1}(1-d)}^1 (1-x)d\Xi(x).$$

Then by equation (3.12) we have that

$$\lim_n \mathcal{L}_{\xi,n,J} = \lim_n \mathcal{L}_{\text{Bayes},n,J}$$

which we denote $\mathcal{L}_{\text{Bayes},J}$. In the next section we show that our estimator asymptotically achieves this level.

Asymptotic Error analysis

Let

$$\zeta_i = \frac{\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^J \frac{q_n^j(R_i^j)}{1-q_n^j(R_i^j)}}{1+\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^J \frac{q_n^j(R_i^j)}{1-q_n^j(R_i^j)}}$$

and

$$\zeta_i' = \frac{\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{p^{j*}(R_i^j/n)}{1-p^{j*}(R_i^j/n)}}{1+\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{p^{j*}(R_i^j/n)}{1-p^{j*}(R_i^j/n)}}.$$

Since $\frac{\left(\frac{1-d}{d}\right)^{J-1}x}{1+\left(\frac{1-d}{d}\right)^{J-1}x}$ is an increasing function of x, the ordering of the ζ_i is the same as the ordering according to $\prod_{j=1}^{J}\frac{q_n^j(R_i^j)}{1-q_n^j(R_i^j)}$ and thus our classifier is equivalent to choosing the dn genes with the largest values of ζ_i. Our construction of q_n^j was designed to approximate p^{j*} as demonstrated in Proposition 3.6.5 together with Proposition 3.6.3 so we begin by considering ζ_i' and comparing it to ξ_i.

Lemma 3.6.7. *For each list j,*

$$\max_i\left|p^{j*}(R_i^j/n)-\frac{d\widetilde{\phi}(T_i^j)}{d\widetilde{\phi}(T_i^j)+(1-d)\phi(T_i^j)}\right|\to 0 \tag{3.13}$$

in probability as $n\to\infty$ and hence

$$\max_i\left|\zeta_i'-\xi_i\right|\to 0 \tag{3.14}$$

in probability as $n\to\infty$.

Proof. By the Glivenko-Cantelli Theorem

$$\max_i\left|F^*(T_i^j)-R_i^j/n\right|\to 0$$

in probability as $n\to\infty$ and since

$$p^{j*}(F^*(T_i^j))=\frac{d\widetilde{\phi}(T_i^j)}{d\widetilde{\phi}(T_i^j)+(1-d)\phi(T_i^j)}$$

and $p^{j*}(\alpha)$ is uniformly continuous on $[0,1]$ we have equation (3.13). Now plugging the approximation of equation (3.13) into the formula for ζ' and using the fact that we have that $p^{j*}(\alpha)$ is bounded away from 0 and 1 we have that

$$\max_i\left|\zeta_i'-\frac{\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{d\widetilde{\phi}(T_i^j)}{(1-d)\phi(T_i^j)}}{1+\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{d\widetilde{\phi}(T_i^j)}{(1-d)\phi(T_i^j)}}\right|\to 0$$

in probability as $n \to \infty$. Rearranging the second term, we have that

$$\frac{\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{d\widetilde{\phi}(T_i^j)}{(1-d)\phi(T_i^j)}}{1+\left(\frac{1-d}{d}\right)^{J-1}\prod_{j=1}^{J}\frac{d\widetilde{\phi}(T_i^j)}{(1-d)\phi(T_i^j)}} = \frac{d\prod_{j=1}^{J}\widetilde{\phi}(T_i^j)}{d\prod_{j=1}^{J}\widetilde{\phi}(T_i^j)+(1-d)\prod_{j=1}^{J}\phi(T_i^j)} = \xi_i$$

which competes the proof of equation (3.14). □

Let $\mathcal{C}_{\zeta'}$ denote the classifier which takes the dn genes with the highest value of ζ_i' and let $\mathcal{L}_{\zeta',n,J}$ denote its misclassification rate. Then since

$$\max_i \left|\zeta_i' - \mathbb{P}[B_i|\mathcal{G}]\right| \to 0$$

by (3.10) and (3.14) we can apply the same argument as equation (3.11) to get that

$$\lim_n \mathcal{L}_{\zeta',n,J} = \mathcal{L}_{\text{Bayes},J}.$$

We are now ready to establish our main result, Theorem 3.2.1 giving the asymptotic error rate for our estimator. Let R and $\widetilde{R}$ be random variables with CDFs, $H(x)$ and $\widetilde{H}(x)$ respectively. These are the limiting distributions as $n \to \infty$ of $\frac{R_\nu}{n}$ and $\frac{R_{\widetilde{\nu}}}{n}$, the normalized ranks of non-DE and DE genes respectively. For independent copies R^j and $\widetilde{R}^j$ we define

$$Z^j := \log\left(\frac{p^{j*}(R^j)}{1-p^{j*}(R^j)}\right), \qquad \widetilde{Z}^j := \log\left(\frac{p^{j*}(\widetilde{R}^j)}{1-p^{j*}(\widetilde{R}^j)}\right)$$

as asymptotic limits of the building blocks of our estimator. We will use large deviation theory, a summary of which is given in Section A.2 of the appendix, to analyze $\sum_j Z^j$ and $\sum_j \widetilde{Z}^j$ and then compare this with our estimator. As $n \to \infty$, conditional on gene i being non-DE ζ_i' converges in distribution to $\frac{\left(\frac{1-d}{d}\right)^{J-1}\exp(\sum_j Z^j)}{1+\left(\frac{1-d}{d}\right)^{J-1}\exp(\sum_j Z^j)}$. Similarly conditional on gene i being DE ζ_i' converges in distribution to $\frac{\left(\frac{1-d}{d}\right)^{J-1}\exp(\sum_j \widetilde{Z}^j)}{1+\left(\frac{1-d}{d}\right)^{J-1}\exp(\sum_j \widetilde{Z}^j)}$.

As we have assumed that the lists are independent and identically distributed the random variables Z^j and $\widetilde{Z}^j$ are also i.i.d. By the assumption on the densities that $0 < C_1 < \phi(t)/\widetilde{\phi}(t) < C_2$ it follows that $p^{*j}(\alpha)$ is bounded away from 0 and 1. Thus Z^j and $\widetilde{Z}^j$ are bounded random variables with finite mean. The density of $\widetilde{R}^j$ is given by $d^{-1}p^{*j}(r)$ and since $\log(p/(1-p))$ is a strictly increasing function of p we have that

$$\begin{aligned}\mathbb{E}\widetilde{Z}^j &= \int_0^1 \log\left(\frac{p^{j*}(r)}{1-p^{j*}(r)}\right) d^{-1}p^{*j}(r)dr \\ &> \int_0^1 \log\left(\frac{p^{j*}(r)}{1-p^{j*}(r)}\right) dr \int_0^1 d^{-1}p^{*j}(r)dr \\ &= \int_0^1 \log\left(\frac{p^{j*}(r)}{1-p^{j*}(r)}\right) dr,\end{aligned}$$

Similarly, since the density of R^j is $(1-d)^{-1}(1-p^{*j}(r))$ we have that

$$\mathbb{E}Z^j < \int_0^1 \log\left(\frac{p^{j*}(r)}{1-p^{j*}(r)}\right) dr$$

and so $\mathbb{E}Z^j < \mathbb{E}\widetilde{Z}^j$. Since Z^j and $\widetilde{Z}^j$ are bounded, their moment generating functions exist and we can apply Cramer's Theorem [15] and the theory of large deviations. For any $\mathbb{E}Z \leq z \leq \mathbb{E}\widetilde{Z}$, there are smooth functions $\eta(z)$ and $\widetilde{\eta}(z)$ such that

$$\frac{1}{J}\log\left(\mathbb{P}(\frac{1}{J}\sum_j Z^j \geq z)\right) \to \eta(z)$$

and

$$\frac{1}{J}\log\left(\mathbb{P}(\frac{1}{J}\sum_j \widetilde{Z}^j \leq z)\right) \to \widetilde{\eta}(z).$$

Both η and $\widetilde{\eta}$ are smooth functions and $\eta(\mathbb{E}Z) = \widetilde{\eta}(\mathbb{E}\widetilde{Z}) = 0$, and since η is strictly decreasing and $\widetilde{\eta}$ is strictly increasing on the interval $(\mathbb{E}Z, \mathbb{E}\widetilde{Z})$, there exists $\mathbb{E}Z \leq z_0 \leq \mathbb{E}\widetilde{Z}$ such that $\eta(z_0) = \widetilde{\eta}(z_0)$. We use this threshold to analyze $\mathcal{L}_{\zeta',n,J}$. Let $\mathcal{A}_d$ denote the genes with the dn highest values of ζ_i' so

$$\mathcal{L}_{\zeta',n,J} = \frac{1}{n}\mathbb{E}\sum_{i\in\mathcal{A}_d} 1_{B_i^c} + \frac{1}{n}\mathbb{E}\sum_{i\in\mathcal{A}_d^c} 1_{B_i}.$$

Since the number of non-DE genes classified DE must equal the number of DE genes classified non-DE we in fact have,

$$\mathcal{L}_{\zeta',n,J} = \frac{2}{n}\mathbb{E}\sum_{i\in\mathcal{A}_d} 1_{B_i^c} = \frac{2}{n}\mathbb{E}\sum_{i\in\mathcal{A}_d^c} 1_{B_i}.$$

For some fixed y let $M_y = \{i : \zeta_i' > y\}$. Then since $\mathcal{A}_d$ is defined as the genes with the largest dn values of ζ_i' either $M_y \subseteq \mathcal{A}_d$ or $M_y^c \subseteq \mathcal{A}_p^d$. Then either

$$\sum_{i \in \mathcal{A}_d} 1_{B_i^c} \leq \sum_{i \in M_y} 1_{B_i^c}$$

or

$$\sum_{i \in \mathcal{A}_d^c} 1_{B_i} \leq \sum_{i \in M_y^c} 1_{B_i}$$

and so for any y,

$$\begin{aligned} \mathcal{L}_{\zeta',n,J} &\leq \frac{2}{n}\mathbb{E}\sum_{i \in M_y} 1_{B_i^c} + \frac{2}{n}\mathbb{E}\sum_{i \in M_y^c} 1_{B_i} \\ &= 2\mathbb{P}(B_i, \zeta_i' \leq y) + 2\mathbb{P}(B_i^c, \zeta_i' > y) \\ &= 2d\mathbb{P}(\zeta_i' \leq y \mid B_i) + 2(1-d)\mathbb{P}(\zeta_i' > y \mid B_i^c). \end{aligned} \tag{3.15}$$

Taking the threshold

$$y_0 = \frac{\left(\frac{1-d}{d}\right)^{J-1} \exp(Jz_0)}{1 + \left(\frac{1-d}{d}\right)^{J-1} \exp(Jz_0)} \tag{3.16}$$

we have that

$$\begin{aligned} \lim_n \mathcal{L}_{\zeta',n,J} &\leq \lim_n 2d\mathbb{P}(\zeta_i' \leq y_0 \mid B_i) + 2(1-d)\mathbb{P}(\zeta_i' > y \mid B_i^c) \\ &= 2d\mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} Z^j \leq z_0) + 2(1-d)\mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} Z^j \geq z_0) \\ &\leq \exp\left(J\eta(z_0) + o(J)\right). \end{aligned}$$

For the other direction we have that for any y, either

$$\sum_{i \in \mathcal{A}_d} 1_{B_i^c} \geq \sum_{i \in M_y} 1_{B_i^c}$$

or

$$\sum_{i \in \mathcal{A}_d^c} 1_{B_i} \geq \sum_{i \in M_y^c} 1_{B_i}.$$

It follows that

$$\begin{aligned}\lim_n \mathcal{L}_{\zeta',n,J} &\geq \lim_n \frac{1}{n}\mathbb{E}\min\Big\{\sum_{i\in M_{y_0}} 1_{B_i^c}, \sum_{i\in M_{y_0}^c} 1_{B_i}\}\\ &= \min\Big\{\mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} Z^j \leq z_0), (1-d)\mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} Z^j \geq z_0)\}\\ &= \exp\left(J\eta(z_0)+o(J)\right). \qquad (3.17)\end{aligned}$$

Hence with $\rho = \eta(z_0)$ we have that

$$\lim_J \frac{1}{J}\log \mathcal{L}_{\text{Bayes},J} = \rho.$$

Proof of the Main Result

Proof of Theorem 3.2.1.

We are now ready to establish the asymptotic loss rate $\mathcal{L}_{\text{PR}}$ of our classifier $\mathcal{C}_{\text{PR}}$ and establish the main theorem. Now fix $\epsilon > 0$. Recalling Propositions 3.6.1 and 3.6.3 we have that

$$\max_r |p_n^j(r/n) - p^{*j}(r/n)| \to 0, \quad \max_r |q_n^j(r/n) - q^{*j,J}(r/n)|$$

in probability as $n \to \infty$. By Proposition 3.6.5 we have that

$$\sup_x |p^{j*}(x) - q^{*j,J}(x)| \to 0$$

as $J \to \infty$. Altogether, by the triangle inequality, this implies that for any $\delta > 0$ for there exists $J(\delta)$ such that for all $J \geq J(\delta)$ we have that

$$\lim_n \mathbb{P}\left[\max_r |q_n^j(r/n) - p^{*j}(r/n)| \geq \delta\right] \to 0.$$

We can choose $J'(\delta)$ large enough such if $\mathcal{D}$ is the event

$$\mathcal{D} := \left\{\sup_r \left|\log\left(\frac{q_n^j(r)}{1-q_n^j(r)}\right) - \log\left(\frac{p^{j*}(r)}{1-p^{j*}(r)}\right)\right| < \delta\right\}$$

then for all $J \geq J'(\delta)$,

$$\lim_n \mathbb{P}[\mathcal{D}] = 1. \qquad (3.18)$$

We may pick $\delta > 0$ small enough such that

$$\eta(z-\delta) \leq \eta(z_0)+\epsilon, \qquad \widetilde{\eta}(z+\delta) \leq \eta(z_0)+\epsilon.$$

As $\mathcal{C}_{\text{PR}}$ involves ranking the genes according to ζ_i and selecting the dn largest, by the same argument as (3.15) we have that

$$\mathcal{L}_{\text{PR},n,J} \leq 2d\mathbb{P}(\zeta_i \leq y_0 \mid B_i) + 2(1-d)\mathbb{P}(\zeta_i > y_0 \mid B_i^c). \tag{3.19}$$

where y_0 is defined as in (3.16). Now

$$\begin{aligned}
&\limsup_n \mathbb{P}(\zeta_i \leq y_0 \mid B_i) \\
&= \limsup_n \mathbb{P}\biggl(\frac{1}{J}\sum_{j=1}^{J} \log\Bigl(\frac{q_n^j(R_i^j)}{1-q_n^j(R_i^j)}\Bigr) > z_0 \mid B_i\biggr) \\
&\leq \limsup_n \mathbb{P}\biggl(\frac{1}{J}\sum_{j=1}^{J} \log\Bigl(\frac{p^{j*}(R_i^j)}{1-p^{j*}(R_i^j)}\Bigr) > z_0 + \delta \mid B_i\biggr) + \mathbb{P}[\mathcal{D}^c] \\
&\leq \limsup_n \mathbb{P}\biggl(\frac{1}{J}\sum_{j=1}^{J} \widetilde{Z}^j > z_0 + \delta\biggr) \\
&= \exp\Bigl(\widetilde{\eta}(z_0+\delta)J + o(J)\Bigr).
\end{aligned} \tag{3.20}$$

where the first equality is by manipulating ζ_i and y_0, the first inequality is by the definition of $\mathcal{D}$, the second is by equation (3.18) and the fact that conditional on B_i that $\frac{1}{J}\sum_{j=1}^{J}\log\Bigl(\frac{p^{j*}(R_i^j)}{1-p^{j*}(R_i^j)}\Bigr)$ is distributed as $\frac{1}{J}\sum_{j=1}^{J}\widetilde{Z}^j$. The final equality follows from the fact that $\widetilde{\eta}$ is the large deviation rate function $\widetilde{Z}^j$. We similarly have that

$$\limsup_n \mathbb{P}(\zeta_i \geq y_0 \mid B_i^c) \leq \exp\Bigl(\eta(z_0-\delta)J + o(J)\Bigr). \tag{3.21}$$

Substituting equations (3.20) and (3.21) into (3.19) we have that

$$\limsup_n \mathcal{L}_{\text{PR},n,J} = \exp\Bigl(\widetilde{\eta}(z_0+\delta)J + o(J)\Bigr) + \exp\Bigl(\eta(z_0-\delta)J + o(J)\Bigr),$$

and hence we have that

$$\lim_{J\to\infty}\limsup_n \frac{1}{J}\log(\mathcal{L}_{\text{PR},n,J}) \leq \eta(z_0) + \epsilon.$$

As this holds for all $\epsilon > 0$ we have that

$$\lim_{J\to\infty}\limsup_n \frac{1}{J}\log(\mathcal{L}_{\text{PR},n,J}) \leq \eta(z_0) = \rho,$$

the same as the optimal Bayesian rate which completes the proof.

Sub-optimality of alternative methods

The Borda method aggregates ranks, scoring genes according to

$$\sum_{j=1}^{J} -\frac{1}{n} R_i^j$$

and selecting the dn genes with the highest scores. Similarly, the approach of [37] scores genes according to the sum of the truncated ranks,

$$\sum_{j=1}^{J} -\min\{\frac{1}{n} R_i^j, \tau\}.$$

Both of these classifiers are examples of a more general approach of what we will call a *generalized rank based* (GRB) classifier. Such a classifier will take a bounded continuous function $g : [0,1] \to \mathbb{R}$, rank genes according to the score

$$\sum_{j=1}^{J} g(\frac{1}{n} R_i^j)$$

and select the dn genes with the highest scores. When the lists are identically distributed and $p(r) = p^{*j}(r)$ then the classifier $\mathcal{C}_{\zeta'}$ is an element of this class with

$$g_\star(r) = \log(\frac{p(r)}{1-p(r)}). \tag{3.22}$$

In the following theorem we will show that, up to linear transforms, the only asymptotically optimal GRB classifier is $\mathcal{C}_{\zeta'}$.

Theorem 3.6.8. *Let $\mathcal{L}_{g,n,J}$ be the misclassification rate of a generalized rank based classifier with function $g(r)$. If $g(r)$ is not of the form*

$$g(r) = a g_\star(r) + b$$

for some $a, b \in \mathbb{R}$ then

$$\lim_{J\to\infty} \limsup_{n} \frac{1}{J} \log(\mathcal{L}_{g,n,J}) > \rho. \tag{3.23}$$

In particular, since the classifiers of Borda and truncated Borda are not chosen according to the Bayesian log-odds ratio, the classifier $\mathcal{L}_{\mathrm{PR},n,J}$ has an asymptotically lower misclassification rate.

Proof. As in Section 3.6 let R and $\widetilde{R}$ be random variables with CDFs, $H(x)$ and $\widetilde{H}(x)$ respectively and let R^j and $\widetilde{R}^j$ denote independent copies of these distributions. Any reasonable function g must have that $\mathbb{E}g(\widetilde{R}) > \mathbb{E}g(R)$. Indeed suppose that $\mathbb{E}g(\widetilde{R}) < \mathbb{E}g(R)$ then by the law of large number,

$$\frac{1}{J}\sum_{j=1}^{J} g(R^j) \to \mathbb{E}g(R), \qquad \frac{1}{J}\sum_{j=1}^{J} g(\widetilde{R}^j) \to \mathbb{E}g(\widetilde{R})$$

almost surely as $J \to \infty$ and so

$$\lim_{J\to\infty} \limsup_{n} \mathcal{L}_{g,n,J} \to 1,$$

that is the misclassification rate tends to 1 as the number of lists tends to infinity and equation (3.23) holds trivially as $\rho < 0$. If $\mathbb{E}g(\widetilde{R}) = \mathbb{E}g(R)$ then set $\sigma^2 = \mathrm{Var}(g(R)), \widetilde{\sigma}^2 = \mathrm{Var}(g(\widetilde{R}))$. Then by the Central Limit Theorem

$$\frac{1}{\sqrt{J}}\sum_{j=1}^{J}(g(R^j) - \mathbb{E}g(R)) \to N(0,\sigma^2), \quad \frac{1}{\sqrt{J}}\sum_{j=1}^{J}(g(\widetilde{R}^j) - \mathbb{E}g(\widetilde{R})) \to N(0,\widetilde{\sigma}^2)$$

in distribution as $J \to \infty$. Choose some z large enough such that

$$(1-d)\mathbb{P}(N(0,1) > z/\sigma) + d\mathbb{P}(N(0,1) > z/\widetilde{\sigma}) = \alpha < d.$$

Then the fraction of genes with score greater than $J\mathbb{E}g(R) + z\sqrt{J}$ converges to α. So if n and J are large enough, we will have that all genes with score at least $J\mathbb{E}g(R) + z\sqrt{J}$ are selected by the classifier. The number of non-DE genes with score above $J\mathbb{E}g(R) + z\sqrt{J}$ is asymptotically $dn\mathbb{P}(N(0,1) > z/\widetilde{\sigma})$ and so a constant fraction of genes are misclassified and so

$$\limsup_{J\to\infty} \limsup_{n} \mathcal{L}_{g,n,J} > 0$$

and hence

$$\lim_{J\to\infty} \limsup_{n} \frac{1}{J}\log(\mathcal{L}_{g,n,J}) = 0 > \rho.$$

Thus it is sufficient to consider the case $\mathbb{E}g(\widetilde{R}) > \mathbb{E}g(R)$. We will analyze this using the theory of large deviations described in Appendix A.2. By Cramer's Theorem there exists $\tau(x) = \tau_g(x)$ such that for $x > \mathbb{E}g(R)$,

$$\tau(x) = \lim_{J} \frac{1}{J}\log\mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} g(R^j) > x)$$

where

$$\tau(x) = \inf_{\theta > 0} \log(\mathbb{E}(\exp(\theta g(R)))) - x\theta.$$

Let $\theta_x = \theta_{x,g}$ be the unique θ achieving the infimum such that

$$\tau(x) = \log(\mathbb{E}(\exp(\theta_x g(R)))) - x\theta_x.$$

Equivalently, if μ is the measure of R on $[0,1]$ and $\mu_{g,\theta}$ is the tilted measure defined by the Radon-Nikodym derivative

$$\frac{d\mu_{g,\theta}(r)}{d\mu(r)} = \frac{e^{\theta g(r)}}{\mathbb{E}(\exp(\theta g(R)))}$$

then we have that

$$\tau(x) = -H(\mu_{g,\theta_x}|\mu),$$

the relative entropy of μ_{g,θ_x} with respect to μ. Moreover,

$$\int_0^1 g(r) d\mu_{g,\theta_x} = x$$

and

$$\tau(x) = -H(\mu_{g,\theta_x}|\mu) = -\inf_{\mu' : \int_0^1 g(r) d\mu' \geq x} H(\mu'|\mu) \tag{3.24}$$

where μ_{g,θ_x} is the unique measure to achieve the infimum. Similarly there exists $\widetilde{\tau}(x)$ such that for $x < \mathbb{E}g(\widetilde{R})$,

$$\begin{aligned} \widetilde{\tau}(x) &= \lim_J \log \mathbb{P}(\frac{1}{J}\sum_{j=1}^{J} g(\widetilde{R}^j) < x) \\ &= \inf_{\theta > 0} \log(\mathbb{E}(\exp(-\theta g(\widetilde{R})))) + \theta x. \end{aligned}$$

Let $x_0 \in (\mathbb{E}g(R), \mathbb{E}g(\widetilde{R}))$ be chosen such that

$$\tau(x_0) = \widetilde{\tau}(x_0).$$

Similarly to the analysis yielding equation (3.17) we have that

$$\lim_{J\to\infty} \limsup_n \frac{1}{J} \log(\mathcal{L}_{g,n,J}) = \tau(x_0) = -H(\mu_{g,\theta_x}|\mu) = -H(\widetilde{\mu}_{g,-\widetilde{\theta}_x}|\widetilde{\mu}).$$

Comparing to Section 3.6 have that $\eta(x) = \tau_{g_\star}(x)$ and the optimal asymptotic misclassification rate is

$$\rho = \tau_{g_\star}(z_0) = -H(\mu_{g_\star,\theta_\star}|\mu),$$

where $\theta_\star := \theta_{g_\star, z_0}$. Similarly we can write $\widetilde{\eta}(x) = \widetilde{\tau}_{g_\star}(x)$ and

$$\rho = \widetilde{\tau}_{g_\star}(z_0) = -H(\widetilde{\mu}_{g_\star, -\widetilde{\theta}_\star} | \widetilde{\mu}).$$

We claim that in fact

$$\mu_{g_\star, \theta_\star} = \widetilde{\mu}_{g_\star, \widetilde{\theta}_\star}. \tag{3.25}$$

Since by Proposition 3.6.1 the probability that the gene ranked r is DE with probability asymptotically $p(r/n)$ we have that

$$\frac{d\mu}{dr} = \frac{1}{1-d}(1 - p(r)), \qquad \frac{d\widetilde{\mu}}{dr} = \frac{1}{d} p(r).$$

Furthermore as

$$g_\star(r) = \log(p(r)) - \log(1 - p(r))$$

we have that

$$\frac{d\mu_{g_\star, \theta}}{dr} = \frac{1}{Z}(p(r))^\theta (1 - p(r))^{1-\theta}, \qquad \frac{d\widetilde{\mu}_{g_\star, -\theta}}{dr} = \frac{1}{\widetilde{Z}}(p(r))^{1-\theta}(1 - p(r))^\theta.$$

where $Z, \widetilde{Z}$ are normalizing constants. Since

$$\int_0^1 g_\star(r) d\mu_{g_\star, \theta_\star}(r) = \int_0^1 g_\star(r) d\widetilde{\mu}_{g_\star, -\widetilde{\theta}_\star}(r) = z_0,$$

and $\int_0^1 g_\star(r) d\mu_{g_\star, \theta}(r)$ is strictly increasing in θ it follows that equation (3.25) holds and the measures are equal.

Now suppose that (3.22) does not hold. Let

$$x_\star = \int_0^1 g(r) d\mu_{g_\star, \theta_\star}$$

be the expected value of $g(r)$ under the measure $\mu_{g_\star, \theta_\star}$. We will assume without loss of generality that $x_\star \geq x_0$, the case of $x_\star \leq x_0$ will follow similarly. Now note that $\mu_{g, \theta_{x_0}} \neq \mu_{g_\star, \theta_\star}$ since g and $g_\star$ are not linear combinations of each other so the reweighed measures must be different. By equation (3.24) since $\int_0^1 g(r) d\mu_{g_\star, \theta_\star}(r) \geq x_0$,

$$\tau(x_0) = -H(\mu_{g, \theta_{x_0}} | \mu) > -H(\mu_{g_\star, \theta_\star} | \mu) = \rho$$

as $\mu_{g, \theta_{x_0}}$ is the unique minimizer of $\inf_{\mu' : \int_0^1 g(r) d\mu'(r) \leq x} H(\mu' | \mu)$. Hence we have that

$$\lim_{J \to \infty} \limsup_n \frac{1}{J} \log(\mathcal{L}_{g,n,J}) = \tau(x_0) > \rho,$$

which completes the proof. □

Chapter 4

Simulation Study

4.1 Metrics for Performance Evaluation

ROC Curve

The receiver-operating-characteristic (ROC) curve is one of the most popular graphical devices for assessing the overall performance of a classifier. For a binary classification system an ROC curve gives a graphical representation of the relationship between the sensitivity and the specificity of a classifier. *Sensitivity* and *Specificity* are defined as the following [19]:

$$\text{Sensitivity} = \frac{\#True\ Positive}{\#Positive}$$

and

$$\text{Specificity} = \frac{\#True\ Negative}{\#Negative}.$$

An ROC curve plots sensitivity against 1-specificity for a classifier when different threshold values are used for the classification (see the example in Section 4.4).

For two randomly chosen elements, one from the positive class and the other from the negative class, the area under the ROC (AUC) gives an estimate of the probability that the classifier thinks the positive item is more likely to belong to the positive class than the negative element [24]. Thus, when comparing multiple classifiers one often prefers the classifier with the largest AUC.

We will explain more in details about interpreting results with an ROC curve in the simulation result section.

PR Curve

The precision-recall (PR) curve is another graphical device that is commonly used to assess the overall performance of a classifier. *Precision* is defined as:

$$\text{Precision} = \frac{\#True\ Positive}{\#Significants}.$$

and *Recall* is equivalent to sensitivity:

$$\text{Recall} = \text{Sensitivity}.$$

Note that *precision* is $(1-FDR)$ for the FDR that we discussed in Section 2.1. As mentioned in Section 2.1 controlling FDR has its practical appeal since biologists are often willing to tolerate some amount of error. In addition, when the positive class has a much smaller proportion than the negative class in the dataset, the PR curve also gives a better presentation than the ROC curve, especially when one is more concerned with identifying the positive than the negative class correctly. Again, we will discuss this in more details in in Section 4.4.

One important point to note is that a classifier with performance that dominates over other classifiers on an ROC curve will also dominates over other classifiers on the PR curve, and vice versa [11]. Thus, when one classifier does consistently better than other classifiers for all threshold values on one of these

4.2 Implementation of the PR-Ranker Algorithm

Let n be the total number of genes, J be the number of lists, and d be the number of DE genes among the n genes. Let r_i^j be the rank for gene i on list j, and let h_i^j denoted the indicator that the gene i on list j is provisionally classified as DE according to the aggregated ranking of all the lists but list j. Also, let A_r^j be the event that the gene ranked r on list j is DE. Finally, let w be the arm length of the window for the smoothing step (see Section 4.4 about how to select a window size).

PR-Ranker Algorithm

Step 1 **Calculate** h_i^j:

Step 1.a For $j = 1, 2, \ldots, J$, aggregate the ranks from all the lists, except from list j, by summing up the ranks for each gene across the lists;

Step 1.b For each list j, rank the genes according to the aggregated ranks obtained in Step 1.a so that the gene with the smallest aggregated rank being the first; ties are resolved randomly;

Step 1.c For each gene i, $i = 1, 2, \ldots, n$, on list j, assign 1 to h_i^j if the gene is ranked among the top d genes in the rank list obtained in Step 1.b.

Step 2 **Estimate** $\mathbb{P}(A_r^j)$ **with smoothing**

Step 2.a For gene i on list j, estimate $\mathbb{P}(A_r^j)$ by averaging h_i^j for all the genes whose original rank on list j is no more than w away from the original rank of gene i on list j.

Step 2.b Adjust $\widehat{\mathbb{P}}(A_r^j)$: If $\widehat{\mathbb{P}}(A_r^j) \leq 0.01$, replace $\widehat{\mathbb{P}}(A_r^j)$ with 0.01; If $\widehat{\mathbb{P}}(A_r^j) \geq 0.99$, replace it with 0.99; otherwise, keep the original $\widehat{\mathbb{P}}(A_r^j)$ value.

Step 3 **Estimate the sum of the log ratios:** For each i, calculate $\Sigma_{j=1}^{J} log(\frac{\widehat{\mathbb{P}}(A_r^j)}{1-\widehat{\mathbb{P}}(A_r^j)})$.

Step 4 **Final ranking:** Rank genes according to the estimated sum obtained in Step 3 so that the gene with the largest estimated sum is ranked the first.

4.3 Simulation Data

Most part of our simulation analysis is based on the following setup with some variations.

We assume that there are J independent rank lists, each with n genes. The values that we will consider for n are $2000, 4000, 6000, 8000$, and 10000, and for J are $4, 6, 8$, and 10. Unless noted otherwise, we set the number of iteration to be 500 for each case that we study. On each list the gene expressions are i.i.d. observations drawn from two classes, one for the DE and one for the non-DE genes. There are 4 replicates for the control group and 4 for the treatment group. The percentage of DE genes for each list is $d = 0.05$ and d remains constant across the lists. Unless stated otherwise, the same set of genes remain being DE throughout all the lists. For non-DE genes we simulate their expressions under both control and treatment conditions from the standard Normal distribution $N(0, 1)$. For DE genes we simulate their expressions under the control condition from the standard Normal distribution $N(0, 1)$; however, under the treatment condition the expressions of the DE genes are simulated from $N(\tilde{\mu}_{ij}, \tilde{\sigma}^2_{ij})$ for gene i on list j where i is taken from the set of the indices for DE genes; here the mean $\tilde{\mu}_{ij}$ is drawn from the $uniform$ distribution over $[-3, -.5] \cup [.5, 3]$, and the variance $\tilde{\sigma}^2_{ij}$ is drawn from $\frac{1}{1.75}|\tilde{\mu}_{ij}|inverseGamma(2.4, 1.4)$, where $inverseGamma(2.4, 1.4)$ is the inverse-gamma distribution with shape parameter 2.4 and scale parameter 1.4; note that this inverse-gamma distribution has mean $\frac{1.4}{2-1} = 1$.

The setup described above was inspired by the simulation setup in [9], except that in [9] there is an additional step to generate the mean expressions for the non-DE genes for the control and treatment groups; however, these means will cancel each other out in the calculation of the differential expression between the treatment and the control groups since both groups have the same mean for the non-DE genes. This is why we skip the step of generating expression means for the non-DE genes here. In addition, we added the factor $\frac{1}{1.75}|\tilde{\mu}_{ij}|$ for the distribution of the variance $\tilde{\sigma}^2_{ij}$ to mimic the biological relationship between the mean difference and the variance of the differential expressions; such pattern is often observed in gene expression data (see Figure 5.2 in Section 5.3 for an example). The purpose of having $\frac{1}{1.75}$ is to rescale the mean of $\tilde{\sigma}^2_{ij}$ back to 1 so that our setup would be more consistent with the setup used in [9].

We will refer to the setup described above as the *base-case* in the rest of the sections in this chapter since other cases that we will considered in the following sections are variations of the base-case.

4.4 Results

Performance in the Base-case

Figures 4.1 (ROC curves) and 4.2 (PR curves) show the performance of the three classifiers, Borda (solid black), SF (dashed black) and PR-Ranker (solid green), for the base-case. We vary the number of genes ($n = 2000, 4000, 6000, 8000, 10000$) for different list numbers (J = 4, 6, 8, 10). On both panels of plots we see that our classifier dominates over the other two classifiers. When the number of lists is small SF performs better than Borda even though both of them perform worse than our classifier. As the number of lists increases the performance of SF and Borda converges; however, both methods fall short compared to our method. In the cases where there are 10 lists, all three classifiers perform well and their performance is almost indistinguishable; however, compared to the other two classifiers, our classifier maintains a lower FDR when the sensitivity is close to 1 (see the right column of the PR curve plots in Figure 4.2), and a higher sensitivity when the specificity is close to 1 (see the right column of the ROC curve plots in Figure 4.1).

To interpret the information shown on the ROC and PR curves, we use the upper-left plot ($n = 2000, J = 4$) on each penal for illustration.

To produce the ROC curve we evaluated the performance of the classifiers for each of the thresholds $1, 2, 3, \ldots, 2000$; e.g., when the threshold was 1, each classifier would identify the top ranked gene chosen by the classifier as DE and the rest of the genes as non-DE; then similarly, when the threshold was 2, each classifier would identify the top two genes as DE and so on. For a point on the ROC curve the x- and the y-coordinates correspond to the fraction of false positives (FP) among all the negatives and the fraction of true positives (TP) among all the positives, respectively, for a particular threshold value. Since the total numbers of positives and negatives remain constant for all threshold values and since $FWER$ can be approximated with $\frac{\text{\# FP}}{\text{\# genes}} \approx \frac{\text{\# FP}}{\text{\# True Negatives}}$ when the fraction of the positives in the population is very small, an ROC curve can be used to estimate the percentage of TP being captured by a classifier for a certain amount of FWER being tolerated. For instance, on the upper-left plot of Figure 4.1 we see that, on average, in order for our method to identify about 90% of the DE genes we would misclassify about 5% of the non-DE genes as DE. This 5% might not seem much at the first glance. However, note that there are $2000 \times .05 = 100$ DE genes and $2000 - 100 = 1900$ non-DE genes in the sample. Thus, this means that in order to discover 90 DE genes we will also misclassify about $1900 \times .05 = 95$ non-DE genes as DE. This means that if we would like to provide the biologist a list of genes that would cover about 90% of the DE genes, the list would contain about 50% non-DE genes. As we can see in this example, an ROC curve does not take into account the imbalance in sizes of the two classes and could depict a misleading optimistic picture. Because of this, one often prefers using the PR curve to evaluate the performance of a classifier. As mentioned in Section 4.1, Precision $= (1 - FDR)$. Thus, if one is more concerned with the FDR associated with the classifier, a PR curve will give a better graphical representation for the analysis.

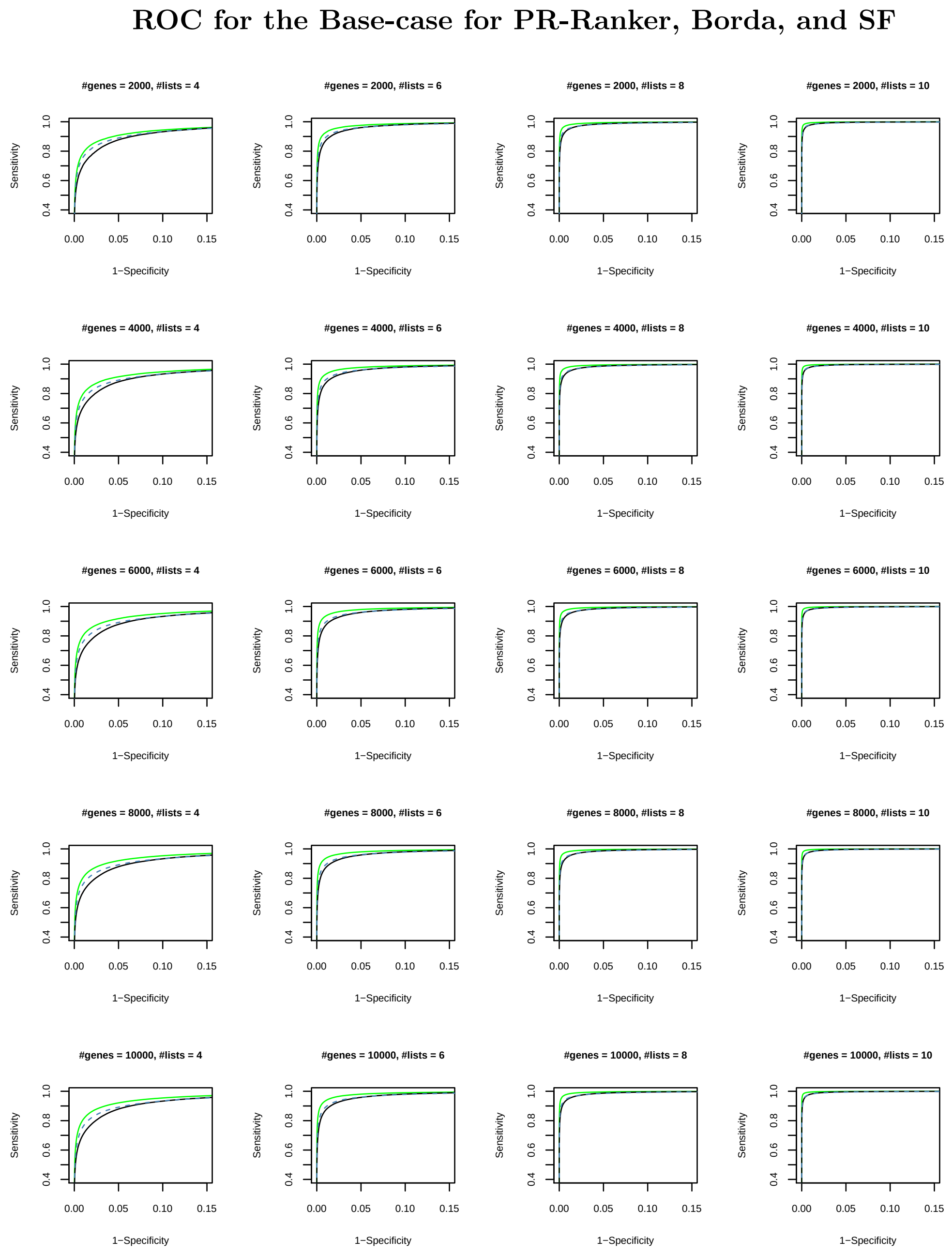

Figure 4.1: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

We now look at the PR curves on the upper-left panel of Figure 4.2. For a point on the PR curve the x- and the y-coordinates correspond to the fraction of TP among all the positives and the fraction of TP among all the significants, respectively, for a particular threshold value. On the upper-left panel of Figure 4.2 we see that for our proposed classifier, PR-Ranker, when the x-value of the PR curve is about 60% the y-value is at least 90%; however, the y-value decreases quickly for bigger x-values and particularly for x bigger than 80%. This means that if one were happy to discover about only 60% of the DE genes our classifier would do pretty well in this case and would make no more than 10% errors (i.e., we can keep the FDR less than 10%) among the genes that we identified; however, the price of identifying the remaining 40% of the DE genes becomes higher and higher, in terms of FDR.

PR curves for the Base-case for PR-Ranker, Borda, and SF

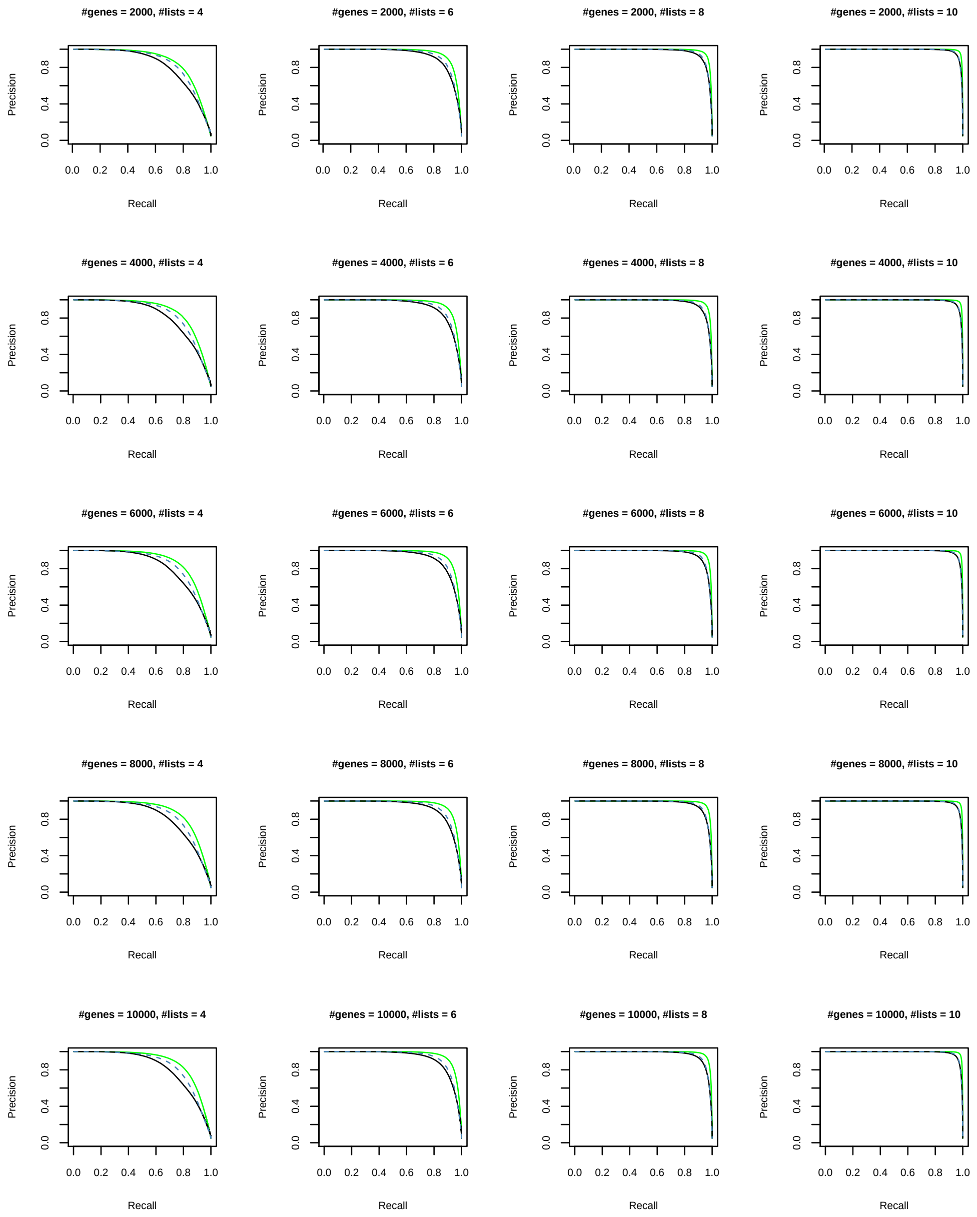

Figure 4.2: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

In summary, our classifier, PR-Ranker, outperforms Borda and SF in the 20 cases that we considered. As the number of lists increases the performance of all three classifiers improves. In the case when there are 10 lists, our classifier wins only by a small margin.

Lastly, even though the ROC curve is a popular choice in applications, as we can see this example, there is a practical appeal of the PR curve. Because of this we will use the PR curve, instead of the ROC curve, for the remaining of the simulation analysis, even though the ROC curve measures something that are more directly close to what our method is trying to optimize (i.e., the total number of false negatives and positives; see the theoretical development of our classifier in Chapter 3).

Asymptotic Loss

In this section we will use simulations to give a demonstration of the main theorem that we proved in Chapter 3. Recall that Theorem 3.2.1 says that our classifier has an asymptotic loss that satisfies the following relationship:

$$\lim_{J\to\infty}\lim_{n\to\infty}\frac{1}{J}\log(\mathcal{L}_{\text{Rank},n,J}) = \rho.$$

An immediate result is that for large n and J

$$\log(\mathcal{L}_{\text{PR},n,J}) \approx J\rho;$$

i.e., the log-loss of our classifier can be approximated by a linear function of the number of lists with slope ρ when n and J are large. In Section 3.6 we showed that any other GRB classifier (except the ones that are equivalent to ours with some positive scale change; i.e., the ones that produce the same ranking results as ours) will have a ρ value bigger than ours.

In this section we will study the asymptotic behaviors of the three classifiers and verify the result of Theorem 3.2.1. To do so we select an unrealistically large value for n, the number of genes, and let $n = 40,000$; then, we calculate the log average misclassification rate for each classifier for the cases when there are $J = 3, 4, \ldots, 30$ lists. The misclassification rate is defined by the total percentage of the false positives and negatives when the classifier calls the top d ranked genes significant, where d is the true number of DE genes.

We will show that asymptotically

- the logarithmic loss of our classifier is always smaller compared to that of Borda (which is a GRB classifier);

- both Borda and our classifier have log-loss values that are a linear function of the number of the lists J.

Although the SF classifier does not fall directly into the category of GRB classifiers (since the statistic that SF uses to rank is not a linear function of some functions of the ranks; see the definition of GRB classifier in Section 3.6) , we will include the log estimated loss for SF here as well for the comparison.

Figure 4.3 plots the log average misclassification rate v.s. the number of lists for each of the classifiers. We superimpose a least-square regression line on the points to emphasize the linear trend of the points for each classifier. Note that the slope of each line is the estimated value for ρ for the associate classifier. As shown on Figure 4.3 our classifier has a slope clearly smaller than that for the other two classifiers. This is consistent with the result stated in Theorem 3.2.1.

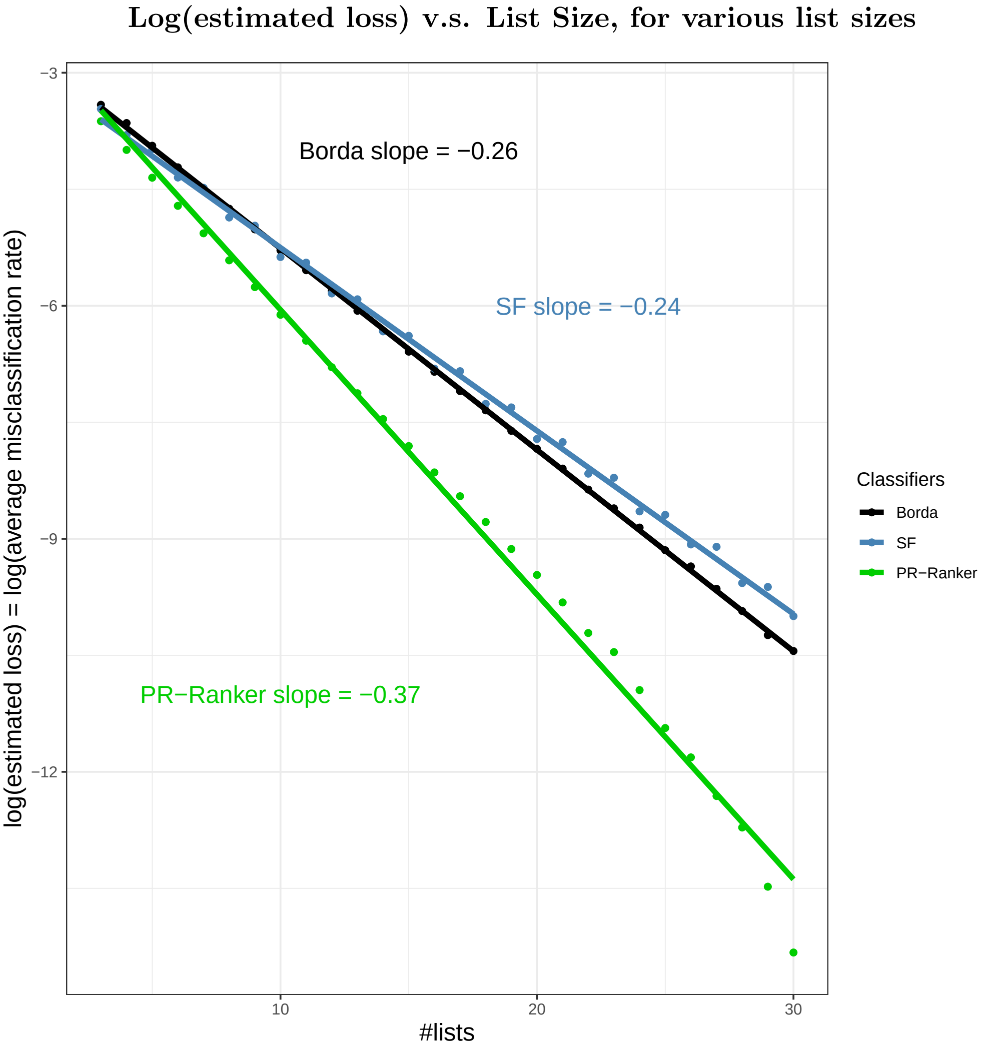

Figure 4.3: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

Robustness in Presence of Corrupted Lists

In this section we study the robustness of the classifiers in the presence of corrupted lists. For each iteration of the simulation we modify the setup in the base-case and include a fraction, b, of corrupted lists where $b = 0, 0.25, 0.5$.

In Figure 4.4 the first row of the plots shows the performance of the classifiers when are no corrupted lists, the second row shows the case when $\frac{1}{4}$ of the lists are corrupted, and the third row is for the case when half of the lists are corrupted. We fix n, the number of genes, to be 5000 for this part of the simulation; then, we generate the corrupted rank lists from the discreet uniform distribution $unif\{1, 2, ...;n\}$.

As we can see on Figure 4.4 that all three classifiers perform worse when part of the data is corrupted than when there is no corruption; in addition, the performance of the classifiers decreases as the fraction of the corrupted lists increases. However, our classifier still performs respectably better than the other two classifiers, especially with a sufficient number of lists. For example, when there are 16 lists, even with half of them being corrupted our classifier is still able to identify over 80% of the DE genes with high precision.

PR curves for the case when 0, 25% or 50% of the lists are corrupted

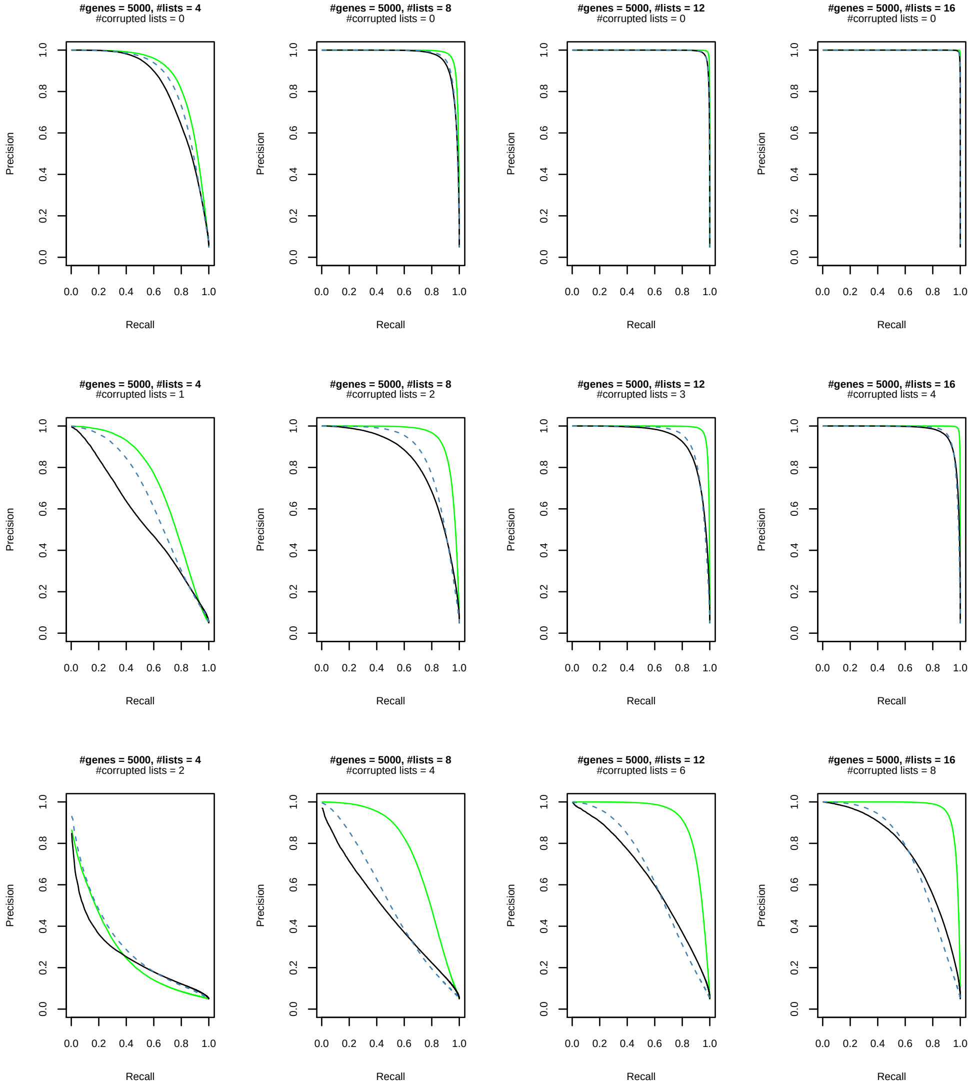

Figure 4.4: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

To understand why our classifier performs almost as badly as the other two classifiers when there are only 4 lists with half of them being corrupted, let's recall the steps in our algorithm in Section 4.2. The ability of our classifier depends on how well we can estimate $\mathbb{P}(A_r^j)$, the probability that the gene ranked r on list j is DE. In addition, the quality of this estimate depends on two factors: the quality of h_i^j in Step 1 and the smoothing procedure in Step 2 of the algorithm.

For a good list, say, list j, h_i^j is calculated based on the ranks in the good lists plus the random noise contributed from the corrupted lists. Because the groupings for the smoothing are of reasonable quality for list j (since list j has reasonable original rankings), $\widehat{\mathbb{P}}(A_r^j)$ is a estimator for $\mathbb{P}(A_r^j)$ with large noise coming from the corrupted lists.

On the other hand, for a corrupted list, say, list j', because the members in a group for the smoothing are collected randomly due to the random nature of the ranks in a corrupted list, all genes are expected to have similar values for $\mathbb{P}(A_r^j)$. As a result when we calculate the sum of the log ratio in Step 3 of the algorithm, the corrupted lists do not play their parts much in discriminating the genes since all genes will have similar estimated $\mathbb{P}(A_r^j)$ values.

Therefore, when the number of good lists is small, with a high proportion of corrupted lists the estimate of the sum of the log ratio becomes highly variable; an example of this be seen on the lower-left plot (the one for 4 lists with 50% corrupted ones) of Figure 4.4.

To further demonstrate the aforementioned concept we look at an extreme case when there are 75% corrupted lists. From Figure 4.5 we see that our classifier does terribly when there is only one good list among four lists; indeed, our classifier behaves as a classifier that just randomly uniformly selects a gene to classify as DE. To understand this behavior note that there is only one good list in this case, and for this good list $\mathbb{P}(A_r^j)$ was calculated based on three corrupted lists that are generated from the uniform distribution. Therefore, roughly speaking in this case our classifier gathers information from 4 random lists and this explains the poor performance of our classifier.

PR curves for comparisons when 75% of the lists are corrupted
Total #lists = 4, 8, 12, 16.

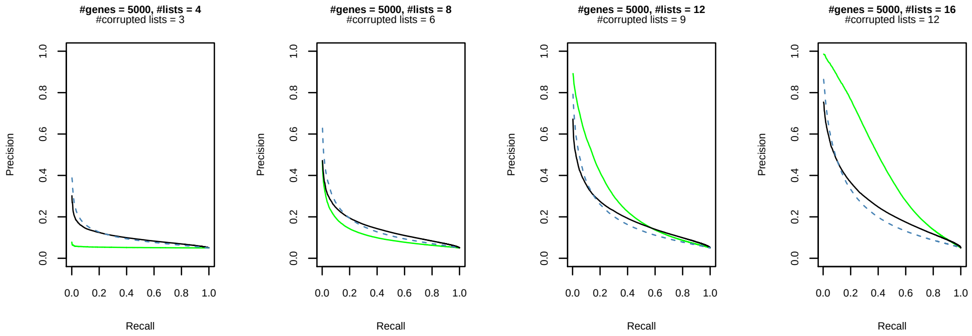

Figure 4.5: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

Note that even in the extreme case when there are 75% corrupted lists, the performance of our method gradually increases as the number of lists increases. In fact, our classifier does considerably better than the other two classifier when there are 16 lists in total and with only 4 good lists. This is because in this case our method weighs the information from the good lists more (since the corrupted lists do not play their parts much in discriminating the genes). In Figure 4.6 we show that our classifier's performance continues improving at a higher rate than the other two classifiers as the number of lists grows.

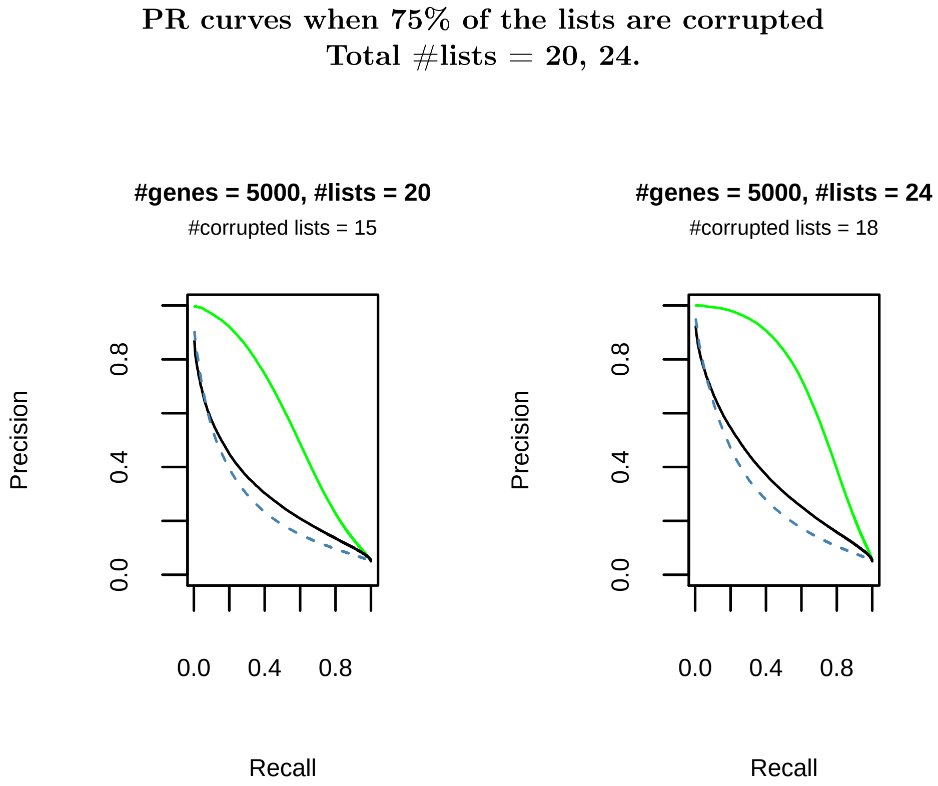

Figure 4.6: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

In summary, we show in this section that compared to Borda and SF our classifier is much more robust in the presence of corrupted lists. In addition, the performance of our classifier grows at a noticeably faster rate than the other two methods as the number of lists grows. A strong strength of our classifier is that it down-weighs the information from corrupted lists.

Effect of Variation of the Signal Strength

In this section we study the behavior of our classifier in the case when the strength of the signal varies among the lists. We will consider two scenarios. In the first case, we dampen the signal by changing the distribution for $\tilde{\mu}_{ij}$ from $uniform\{[-3,-.5]\cup[.5,3]\}$ to $uniform\{\frac{1}{2}[-3,-.5]\cup\frac{1}{2}[.5,3]\}$. In the second case, we vary the strength of signal across the lists and draw $\tilde{\mu}_{ij}$ from $uniform\{\frac{j}{J}[-3,-.5]\cup\frac{j}{J}[.5,3]\}$ for $j=1,2,\ldots,J$. Note that on average the two cases have similar strength of signal (since $\frac{1}{J}\sum_{j=1}^{J}\frac{j}{J}=\frac{J(J+1)}{2J^2}=\frac{J+1}{2J}\approx\frac{1}{2}$ for J that is not too small).

In Figure 4.7 when the signal strength is fixed we see that our classifier performs worse than the other two when the number of genes n and the number of lists J are small. With 8 lists or more our classifier performs similarly as the other two.

PR curves for the case when the strength of the signal is halved with fixed strength of singal for all lists

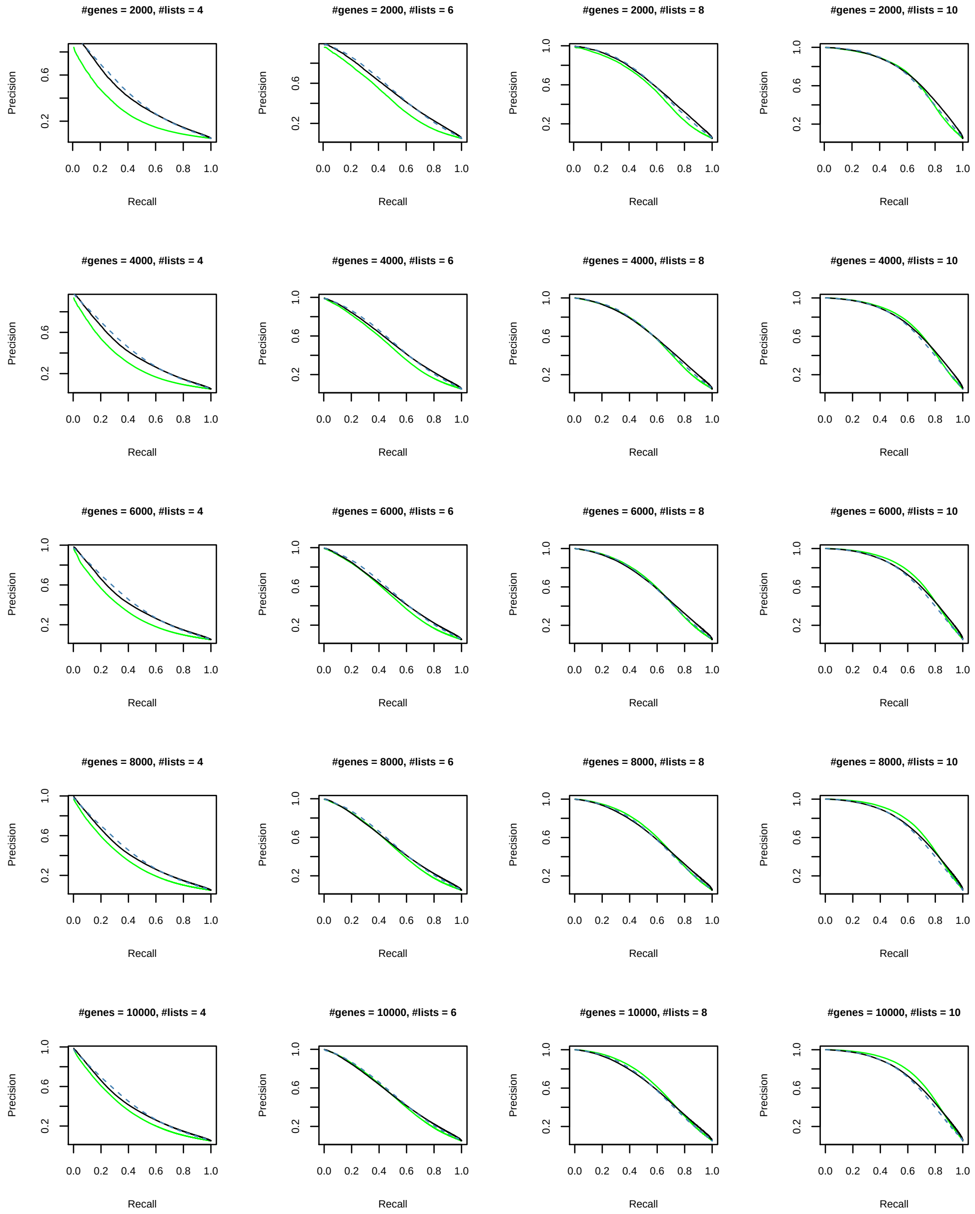

Figure 4.7: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

On the other hand, when the signal strength varies Figure 4.8 shows that our classifier improves significantly over the other two classifiers as the number of lists grows. The reasoning behind this phenomenon is similar to what was explained in the previous section (Section 4.4); because our method downweighs information from lists of low quality, our classifier performs more superiorly when the strength of the signal varies among the lists.

PR curves for the case when the strength of the signal is weaken and varies among the lists

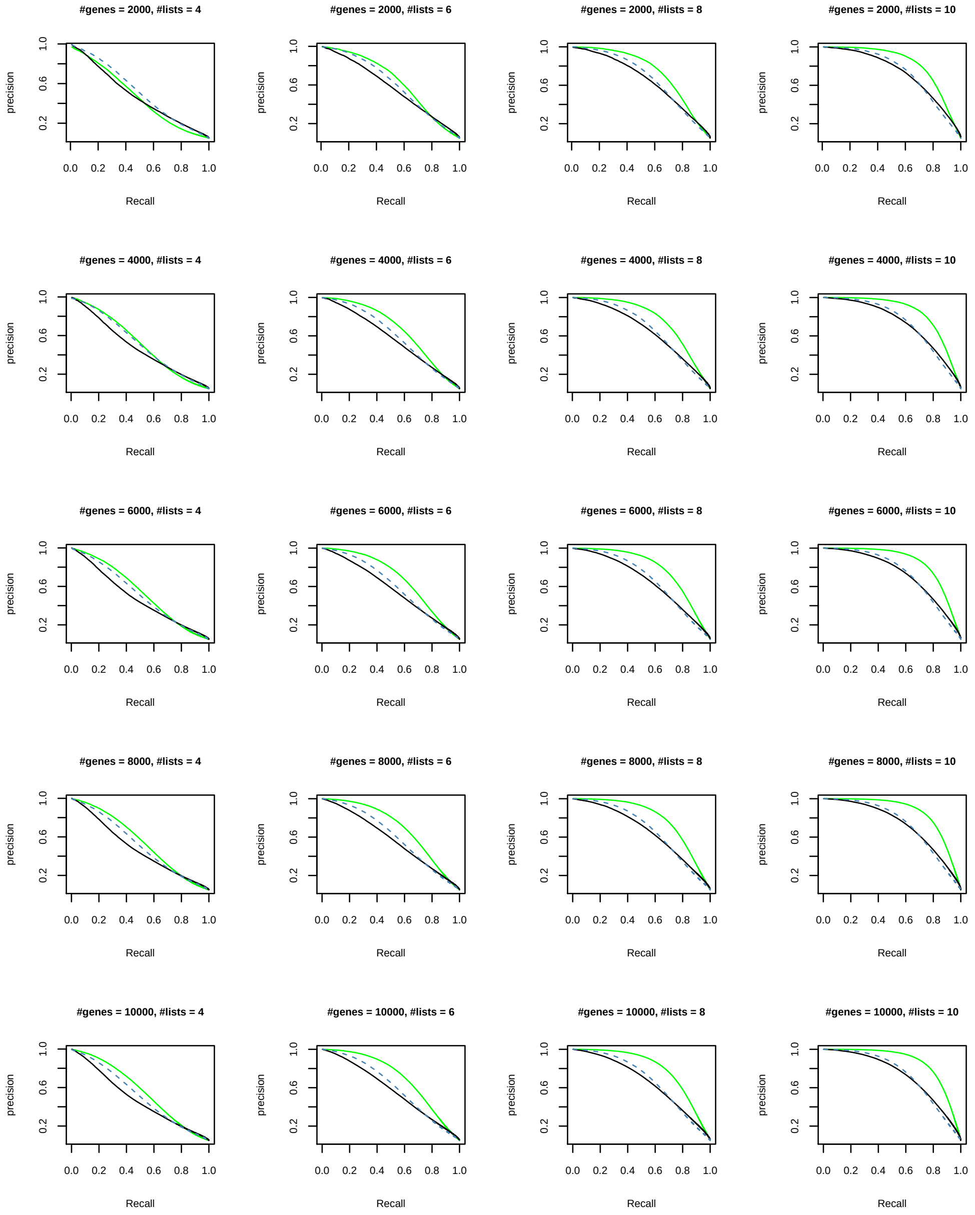

Figure 4.8: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

Robustness when d is unknown

In the development of our classifier (Chapter 3) we assumed that the value of d, the true number of DE genes, is known. We also use this piece of information in our algorithm to provisionally classify genes to be DE. In this section we will investigate the behavior of our classifier when one uses an estimated value for d. We consider the cases when the estimate of d is αd, where $\alpha = 0.5, 1, 2, 3, 4, 5$.

In Figure 4.9 we see that with different factor of d our classifier performs quite similarly. In addition, with 10 lists the performance of the classifier is almost indistinguishable with different choices of α.

More importantly in Figure 4.10 where we zoom in on the plots we see that the quality of our classifier decreases slightly as the estimate on d deviates from the true value of d. We looked at the higher values of α (up to 15) and this pattern continues consistently.

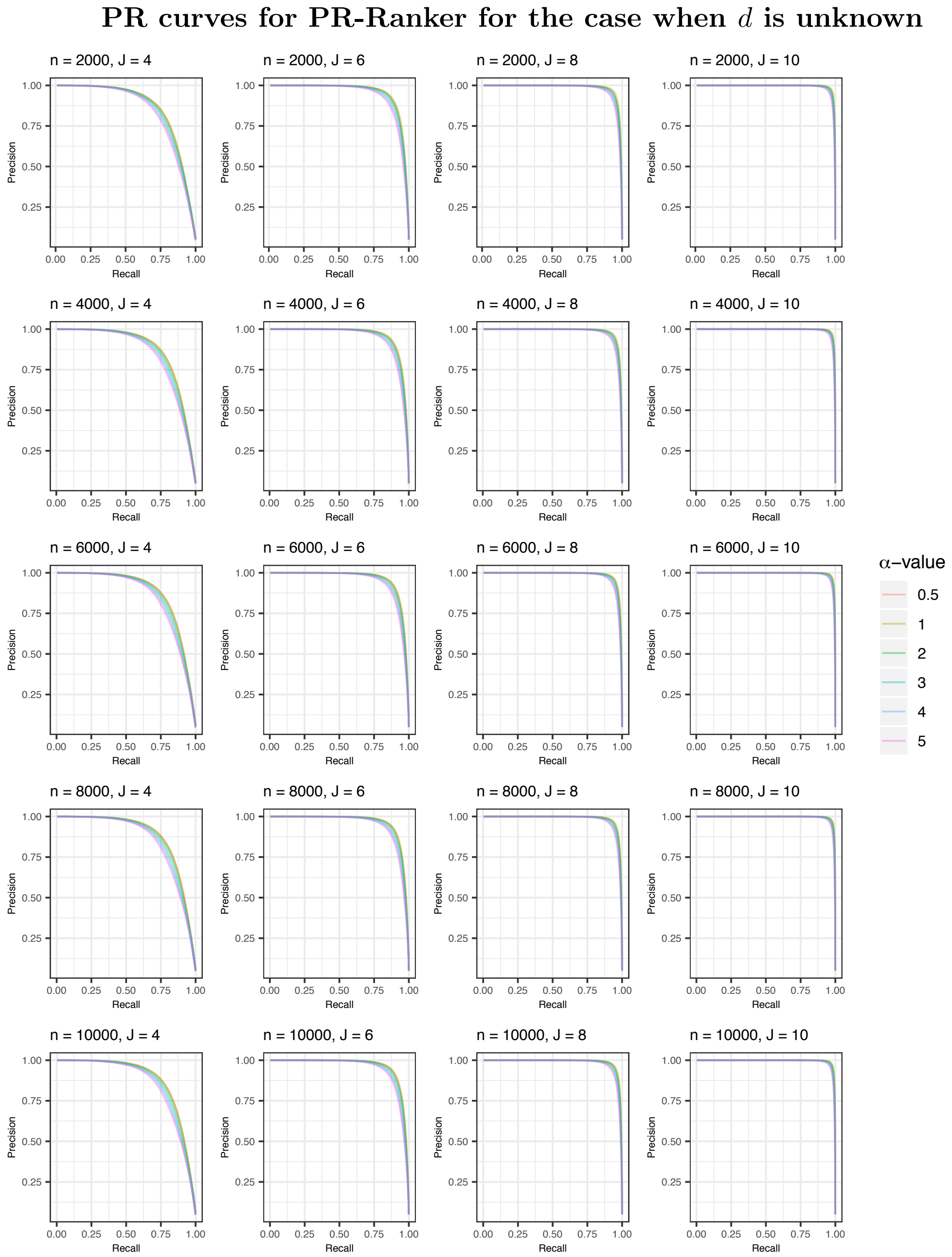

Figure 4.9: Performance of our classifier when αd is used as the cutoff for classifying provisionally DE genes, where $\alpha = 0.5, 1, 2, 3, 4, 5$; $d = .05$ is the true fraction of the DE genes in the data.

PR curves for PR-Ranker for the case when d is unknown (zoomed-in version)

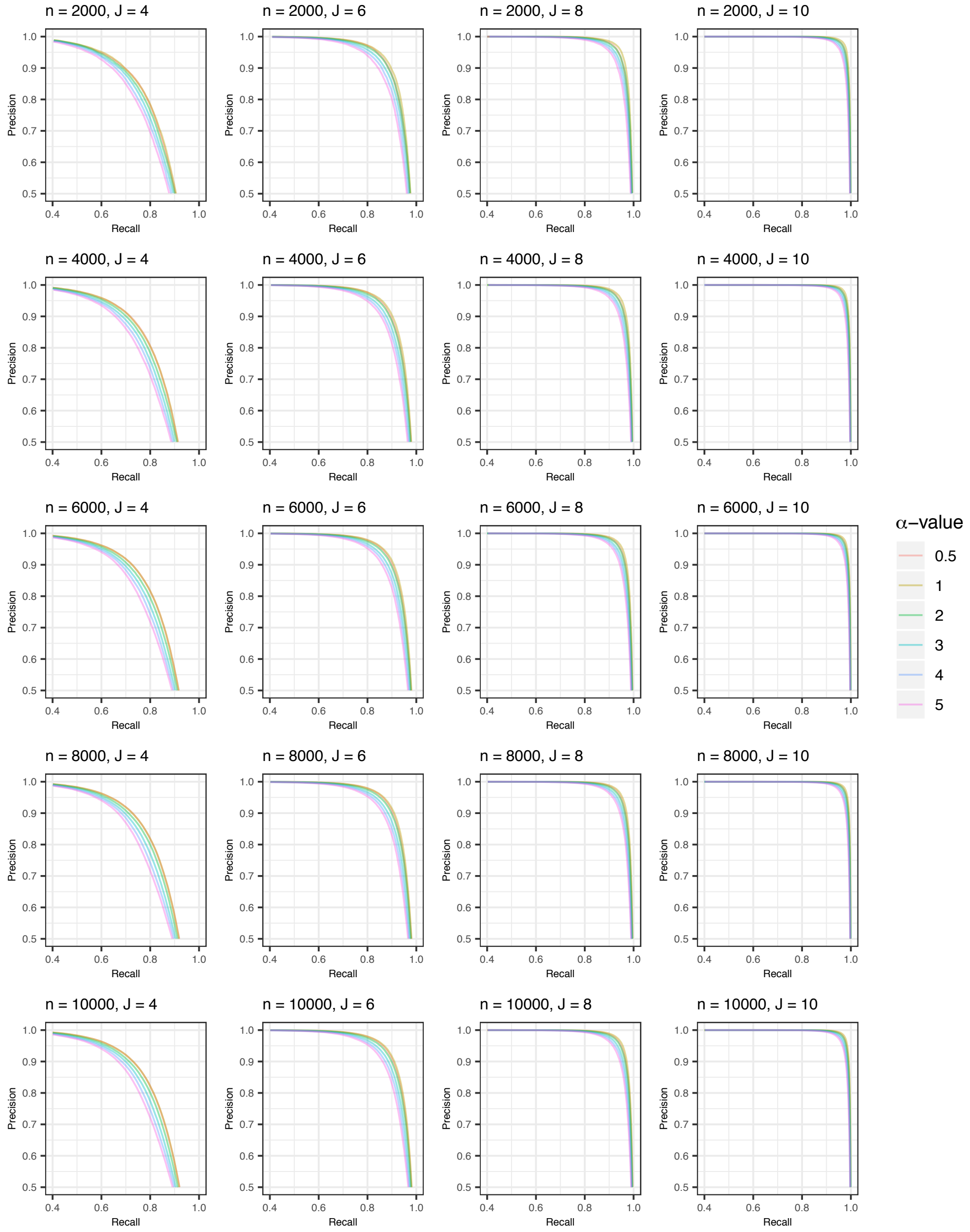

Figure 4.10: Performance of our classifier when αd is used as the cutoff for classifying provisionally DE genes, where $\alpha = 0.5, 1, 2, 3, 4, 5$; $d = .05$ is the true fraction of the DE genes in the data.

Improving Borda with Truncation is Cutoff Dependent

There is a suggestion that one can improve the quality of the aggregated ranking by adding a preprocessing step to use the top-k lists of the original rank lists [37]. This means to truncate the ranks on the original rank lists by replacing all rank values bigger than k with $k+1$ before the aggregation. The motivation behind this preprocessing step is that the top ranks are more reliable and the data usually becomes noisy for the larger ranks [23, 37, 38]. We explore this approach in this section.

Figure 4.11 shows the performance of Borda with top-k lists where $k = \beta d, \beta = 0.5, 1, 2, 3, 4, 5$. We see that the performance of Borda improves as β becomes bigger. However, from Figure 4.12 we see that the performance of Borda starts to decline for β bigger than 7.

PR curves: performance of Borda when truncating rank lists at βd , where $\beta = 0.5, 1, 2, 3, 4, 5$

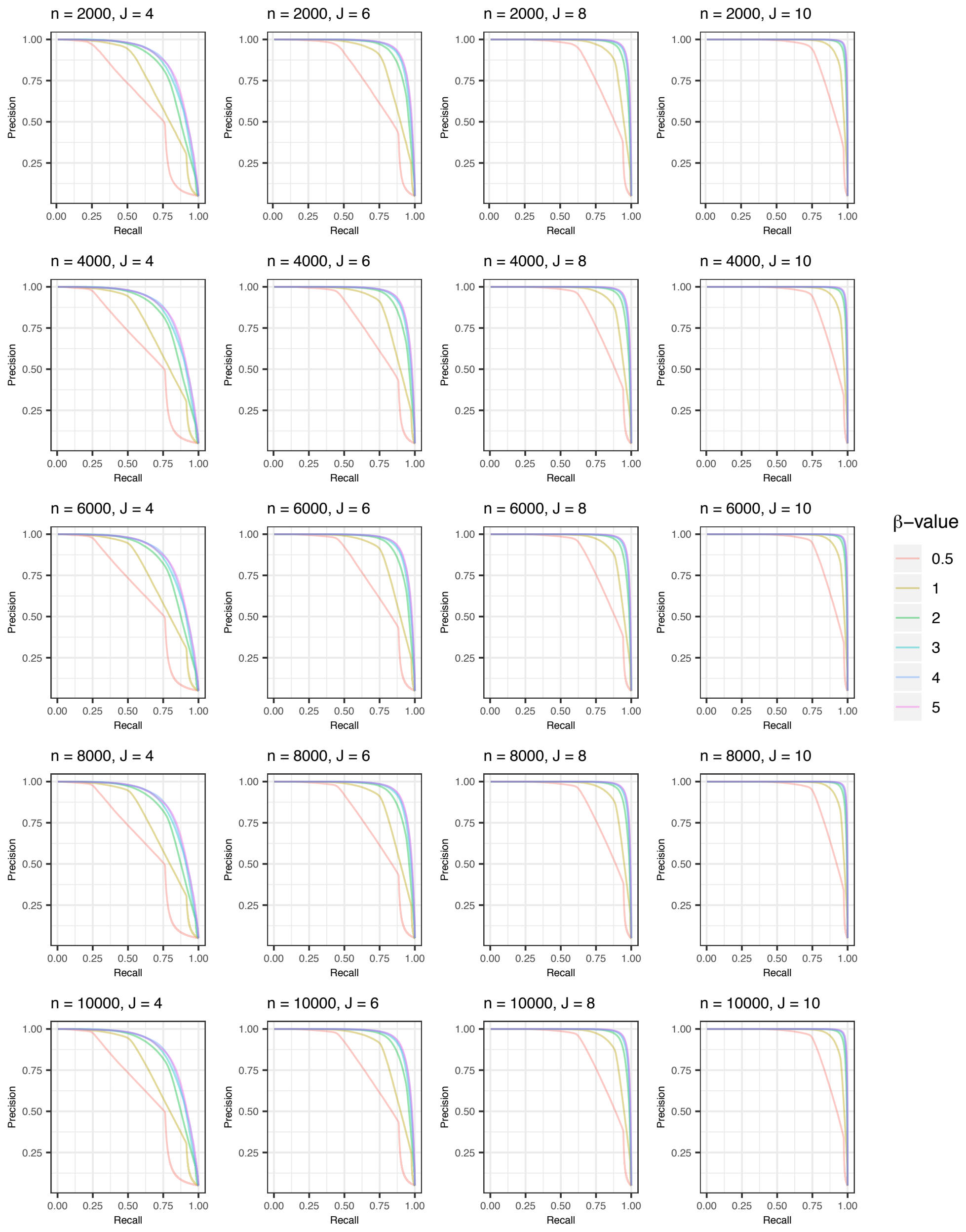

Figure 4.11: Performance of Borda with truncation at βd , where $\beta = 0.5, 1, 2, 3, 4, 5$ and $d = 0.05$ is the true fraction of the DE genes in the data

PR curves: performance of Borda with various truncation values, where $\beta = 5, 7, 9, 11, 13$

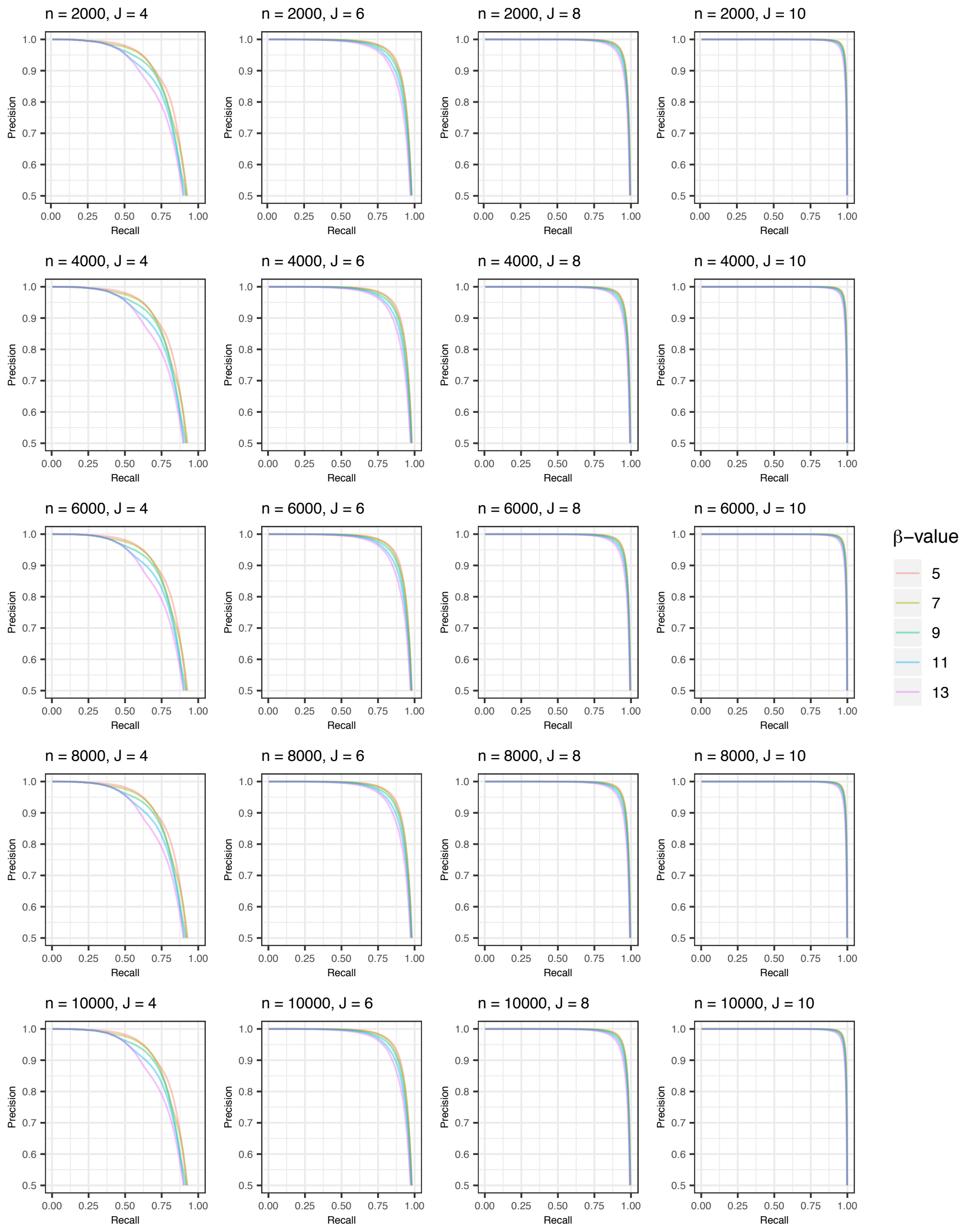

Figure 4.12: Performance of Borda with truncation at βd where $\beta = 5, 7, 9, 11, 13$ and $d = 0.05$ is the true fraction of the DE genes in the data

From the result above one might think that using a big value for β would be a good choice for the truncation in general. However, we now investigate the effect of the truncation factor β further with a different underlying distribution and show that Borda behaves quite differently in this case. We modify the base-distribution and amplify the signal by changing the differential expression mean from $\tilde{\mu}_{ij} \sim uniform\{[-3,-.5] \cup [.5,3]\}$ to $2\tilde{\mu}_{ij}$ while keeping the variance the same as before. Then, for each of the lists we randomly select half of the DE genes and draw their expression values from the distribution for the non-DE genes. This means that we boost the signals for the DE genes in general but randomly mask the signals of half of the DE genes for each list. Figure 4.13 shows the performance of Borda under this new distribution. Note that now the smaller values of β are the better choices for Borda's performance. The β values, such as 5 and 7, that are good for the previous distribution now weaken the performance of the classifier drastically.

PR curves: performance of truncated Borda with the new distribution

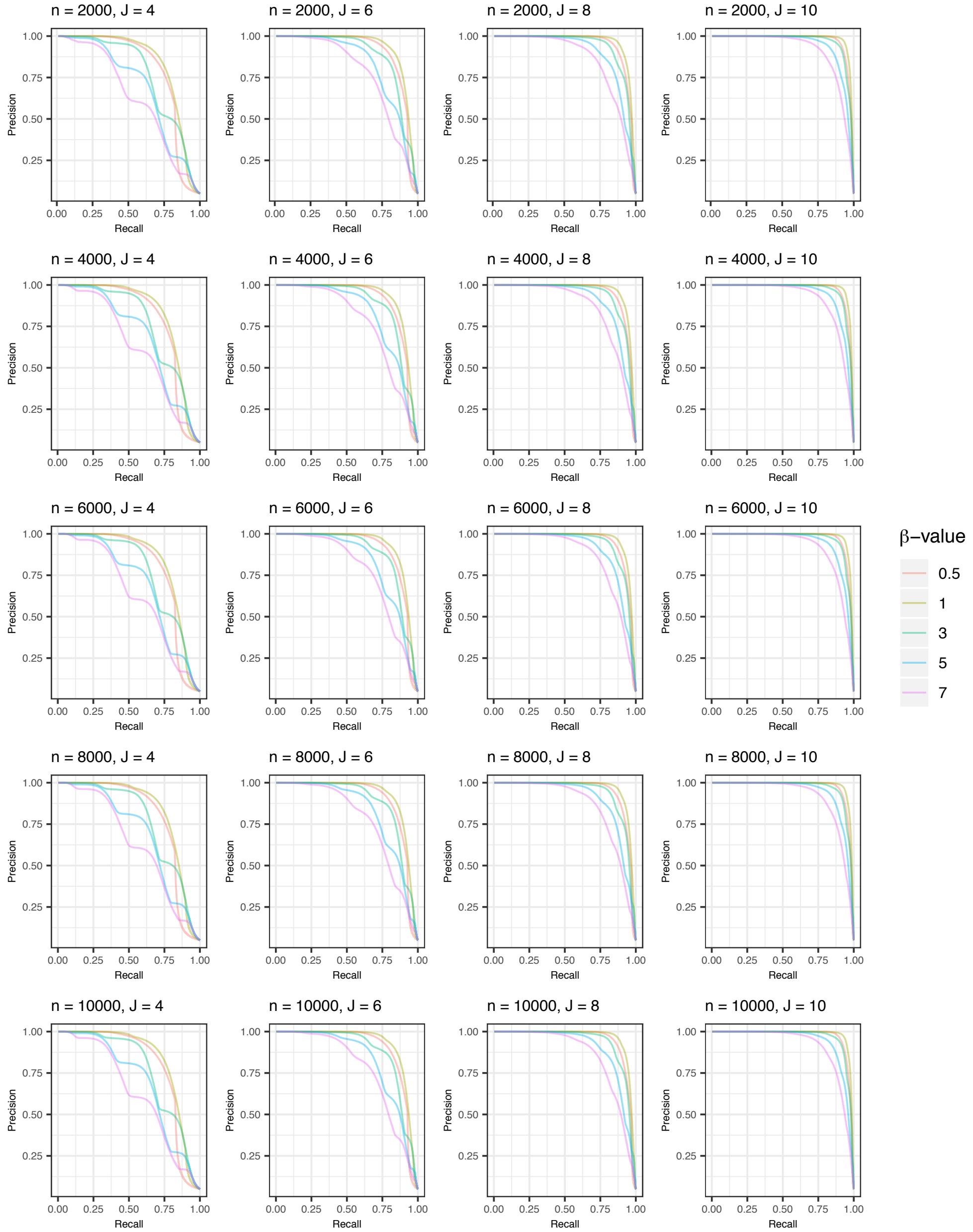

Figure 4.13: Performance of Borda in the case where the signals are stronger in general but for each list the signals of 50% of the DE genes are masked

Figure 4.14 shows the performance of the classifier, SF, with the new distribution. Note that different values of β affects the performance of the classifier vastly. Furthermore, in some cases (e.g., when the number of lists is 4 or 6) no β value gives a result (in terms of precision) that dominates over all other β values.

Figure 4.15 shows the performance of our classifier with the new distribution. Note that the value of α (as in αd, the cutoff for classifying provisionally DE genes) does not affect the performance of the classifier as much as how β affects the performance of Borda and SF. In addition, it is worth noting that the choice of α becomes less important as the number of the lists grows; for 10 lists the performance of the classifier becomes practically indistinguishable for various choices of the value for α.

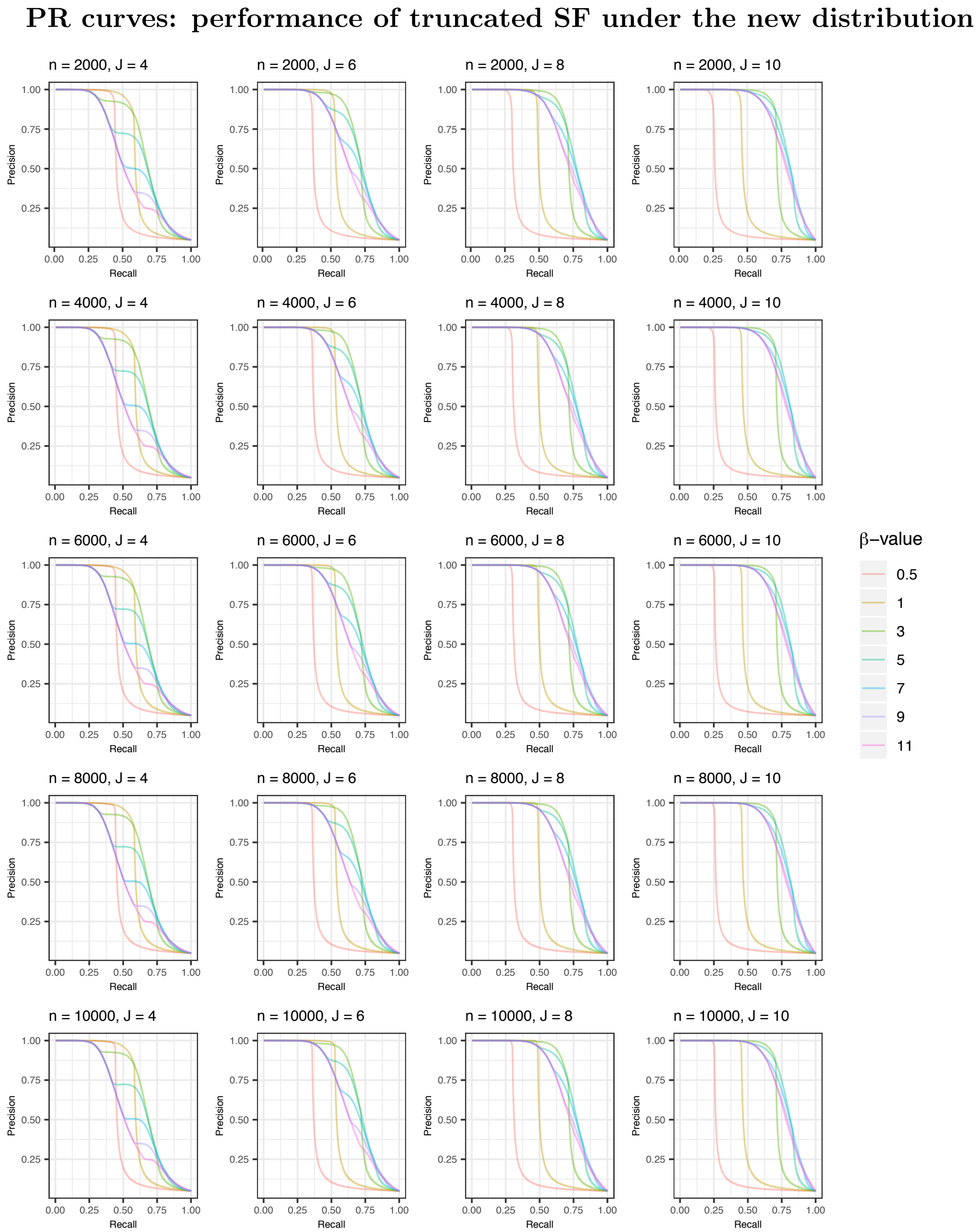

Figure 4.14: Performance of SF in the case where the signals are stronger in general but for each list the signals of 50% of the DE genes are masked

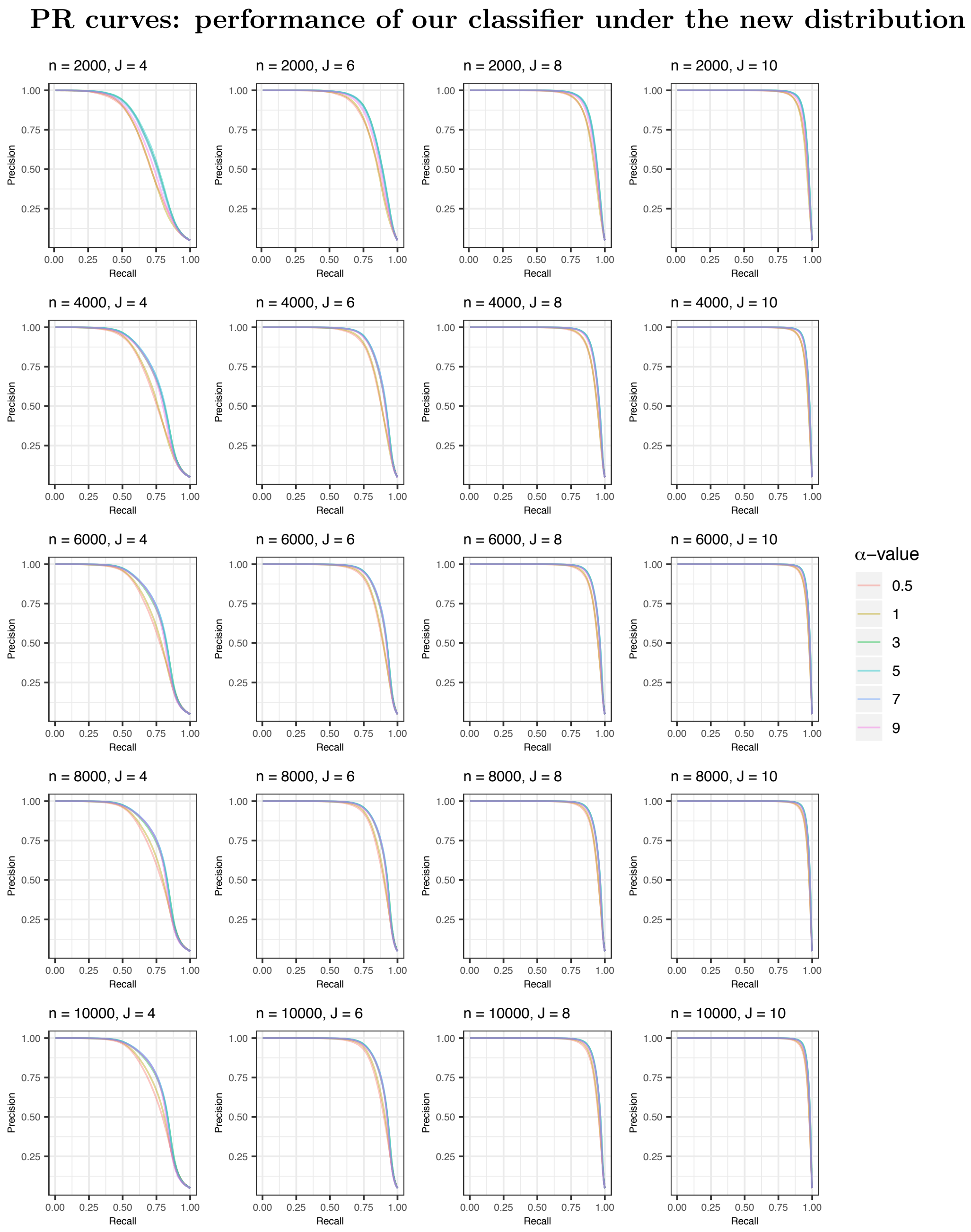

Figure 4.15: Performance of our classifier in the case where the signals are stronger in general but for each list the signals of 50% of the DE genes are masked; αd is used as the cutoff for classifying provisionally DE genes, where $\alpha = 0.5, 1, 3, 5, 7, 9$

In summary, this section we study the behavior of the Borda and SF classifiers when they are applied to datasets with truncated ranks. This truncation procedure was suggested by previous literature to aim at improving data quality. However, in our analysis we see that the performance of the classifiers are sensitive to the choice of the cutoff for the truncation and that different underlying distributions require different values for the truncation in order for the classifiers to achieve the optimal performance; in particular, some choice of the value for truncation drastically weakens the performance of Borda and SF. This could be problematic when the desired factor for d, where d is the true number of DE genes, is large. For example, if the desired cutoff is $6d$, an over- or under- estimate on d by a factor of 2 will change the cutoff for the truncation to be $3d$ or $12d$ and such big change is likely to affect the performance of the classifiers. In addition, since the underlying distribution of the gene expressions is usually unknown, it is difficult to decide on the best cutoff value to use for the truncation as the optimal choice of the cutoff value depends on the underlying distribution. In contrast, our classifier behaves in a stable manner among the difference choices for the cutoff value used to classify genes as provisionally DE. More importantly, for our classifier the choice for the cutoff value becomes practically insignificant for the performance of the classifier when the number of the lists is large.

Technical Notes

Choice of the Smoothing Factor for PR-Ranker

The window size $w = n^{\frac{1}{2}}$ was chosen somewhat arbitrary in the proof–the main reason for the choice was to keep the notation straightforward. In fact, the exponent could be replaced with any positive real number that is less than 1 and the proof would still be carried out exactly the same way. We also use this value $\frac{1}{2}$ for our simulation study. Different choices of the window size will affect the performance of the classifier slightly.

In practice, one can use a more advanced method, such as LOWESS (Locally Weighted Scatterplot Smoothing) or cubic smoothing spline, for the smoothing. Packages for these smoothing methods are available in `R`.

In practice we suggest using a more advanced method, such as the LOWESS (Locally Weighted Scatterplot Smoothing) or cubic smoothing spline. To choose a value for smoothing parameter for these more advanced methods, one can look at the data to find a value that gives a monotonic decreasing shape for the probability that we are trying to smooth.

If one prefers to do the smoothing manually with a fixed window size and equal weights, when the list is short, one needs to be careful about that the window for the smoothing is not too large; otherwise, there might be a problem with over-smoothing. In general, we recommend keeping the window size to no more than $\frac{1}{4}$ of the number of the genes (otherwise, we would include more than half of the list of the genes in our smoothing window); i.e.,

$$\frac{n^w}{n} \leq \frac{1}{4}$$

where n is the number of genes, w is the power raised to calculate the arm size of the window size. This means that the window size should satisfy the criterion

$$w \leq 1 - \frac{log(4)}{\log(n)}.$$

4.5 Discussion

Through the simulation analysis in this chapter we see that our method outperforms two well-known rank-based aggregation methods, Borda and SF, under various simulation conditions. While Borda and SF weigh the information in each list equally, our classifier automatically adapt to data quality between lists and down-weighs the information from the lists of lower quality. This property of our classifier is particularly valuable since in practice there maybe a varying level of noise in datasets collected in different experiments and with different technologies.

Chapter 5

Application

In this chapter we apply our algorithm to a set of data collected on 157 placenta samples to identify genes that can potentially serve as biomarkers for preeclampsia. Preeclampsia is the most serious hypertensive disorder of pregnancy and is one of the major causes of maternal deaths. In Western Europe and North America preeclampsia occurs to 2-5% of pregnancies and this figure increases to as high as 18% in some parts of Africa [36, 59].

Although the placenta is considered to be the primary cause of preeclampsia [42], the exact cause of the disorder is believed to be multifactorial; for example, the disorder has higher incident rate among women who are nulliparous, or with pre-existing metabolic, vascular or renal disease [22].

Because of the multifactorial nature of the condition, the heterogeneity in patient samples has led to inconsistent results in past studies [35, 17, 32]; in addition, previous studies have produced results that have been shown to have low sensitivity [35]. Consequently, no consensus has been reached for the list of genes that are associated with the disorder [35, 17, 32].

In this chapter we apply the three classifiers (Borda, SF and PR-Ranker) to a gene expression dataset that is collected from 157 placenta samples and analyze and compare the results produced by the classifiers.

5.1 Data Description

The dataset that we use for the analysis in this chapter is from the study conducted by Leavey et al [35]. There are a total of 157 placenta samples (77 control and 80 treatment samples). The samples were acquired from the patient sample set at the Research Centre for Women's and Infants' Health BioBank (Mount Sinai Hospital, Toronto, Canada). Four tissue biopsies were collected and processed into powder per placenta by the BioBank and mRNA was extracted from the placental samples and sent to the Princess Margaret Genomics Centre (Toronto, Canada) for hybridization against Human Gene 1.0 ST Array chips (Affymetrix). The microarray data was then normalized by using the *Affy* package in R 3.0.1. Some genes were filtered out for quality control. The resulting dataset contains the expressions of 14651 genes. The microarray data can be found in the Gene Expression Omnibus (GEO) database [35].

5.2 Method and Metrics Used

We analyze the performance of the three classifiers (Borda, SF and PR-Ranker) by investigating two properties of the classifiers: consistency and sensitivity.

For consistency, we divide the control replicates into two subsets of similar sizes, and do the same for the treatment replicates. From each pair of control-treatment subsets we sample, without replacement, 4 samples of controls and 4 samples of treatments. Then, a t-statistic is calculated with the 4 control and 4 treatment samples for each gene; after that the 14651 genes are ranked by using their t-statistics, such that, the gene with the biggest absolute value of the t-statistics being ranked first. This creates the rank list for one pseudo study. We then continue drawing samples from the same pair of control-treatment subsets (samples drawn for making the previous lists can be drawn again) to make more pseudo rank lists until we collect the desirable number of pseudo rank lists. The same procedure is repeated for the other pair of control-treatment subsets. We now have two set of pseudo ranking lists and can see whether a classifier will generate similar rank results when applied to these two sets of rank lists.

We use the *Consistency Index* proposed by Kuncheva [34] to assess the consistency of a classifier. Suppose that there are n genes and an algorithm calculates two aggregated rank lists for the two set of rank lists described in the previous paragraph. Let k be any positive integer such that $k = 1, 2, \ldots, n-1$. For a given k there are two set of top-ranked k genes, $\mathcal{A}$ and $\mathcal{B}$, according to the two aggregated lists. The consistency index for the sets $\mathcal{A}$ and $\mathcal{B}$ is defined to be

$$I_s(\mathcal{A}, \mathcal{B}) = \frac{\Psi n - k^2}{k(n-k)},$$

where $\Psi = |\mathcal{A} \cap \mathcal{B}|$ is the cardinality of the intersection of the subsets $\mathcal{A}$ and $\mathcal{B}$. Note that the consistency index has the following properties:

- For a fixed k the consistency index is monotonically increasing with Ψ;
- $-1 \leq I_s(\mathcal{A}, \mathcal{B}) \leq 1$; $I_s(\mathcal{A}, \mathcal{B}) = -1$ when $\Psi = 0$ and $k = \frac{n}{2}$, and $I_s(\mathcal{A}, \mathcal{B}) = 1$ when $\Psi = k$ for all k;
- The consistency index should be around 0 for any two randomly generated independent rank lists since in this case the expected value for Ψ is $\frac{k^2}{n}$ which implies that the expected value of the consistency index is zero.

The procedure to assess the sensitivity of the classifiers will be described in the result section.

5.3 Exploratory Analysis of the Data

The (log transformed) gene expressions in the dataset have identical distributions for the control and treatment groups (see figure 5.1); this is due to normalization. Note that the distributions are slightly right skewed even after taking log_2 transformation for the original gene expression data; this indicates that the original data is even more right skewed.

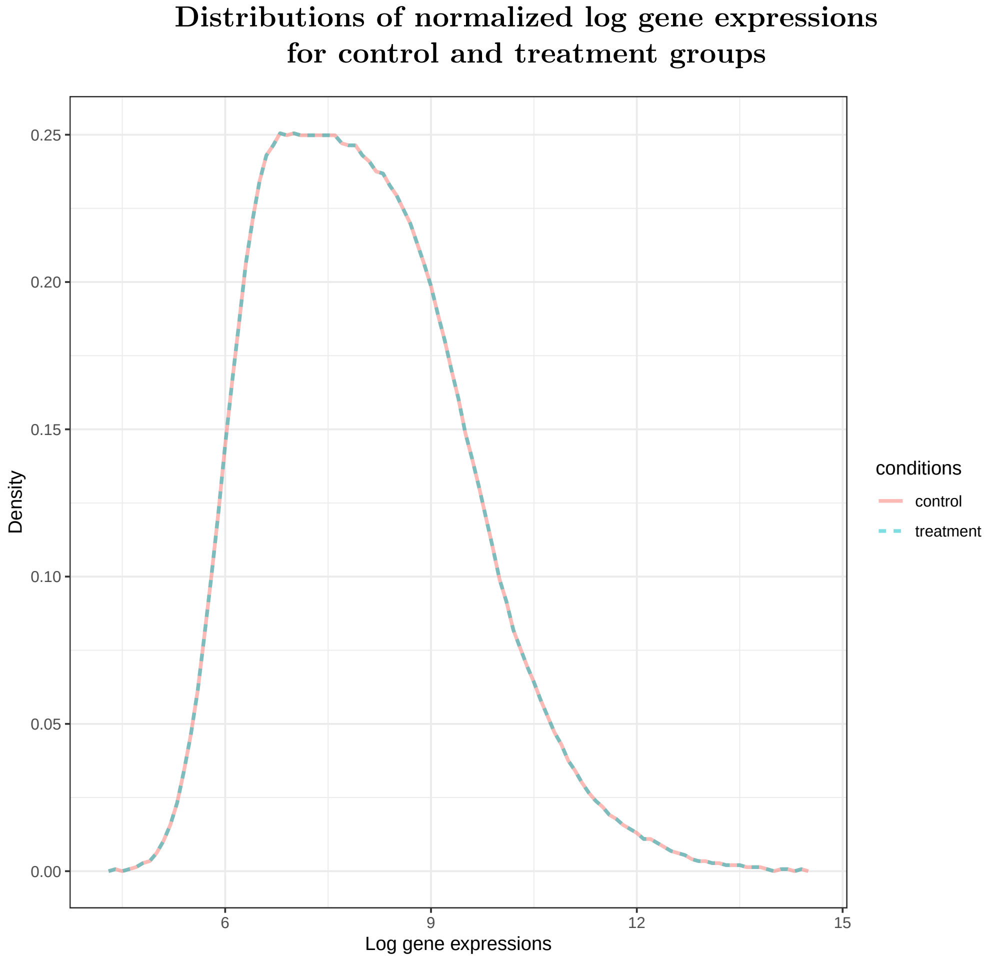

Figure 5.1: Histograms for the distributions of the normalized gene expressions

We also look at the relationship between the estimated SE of the sample mean difference and the difference of the sample means. We superimpose the LOWESS curve of the data on top of the scatterplot to emphasize the positive relationship between the x- and y- variables. We see a positive relationship between the two variables. This validates the reasonable choice of the distributions used in our simulation study.

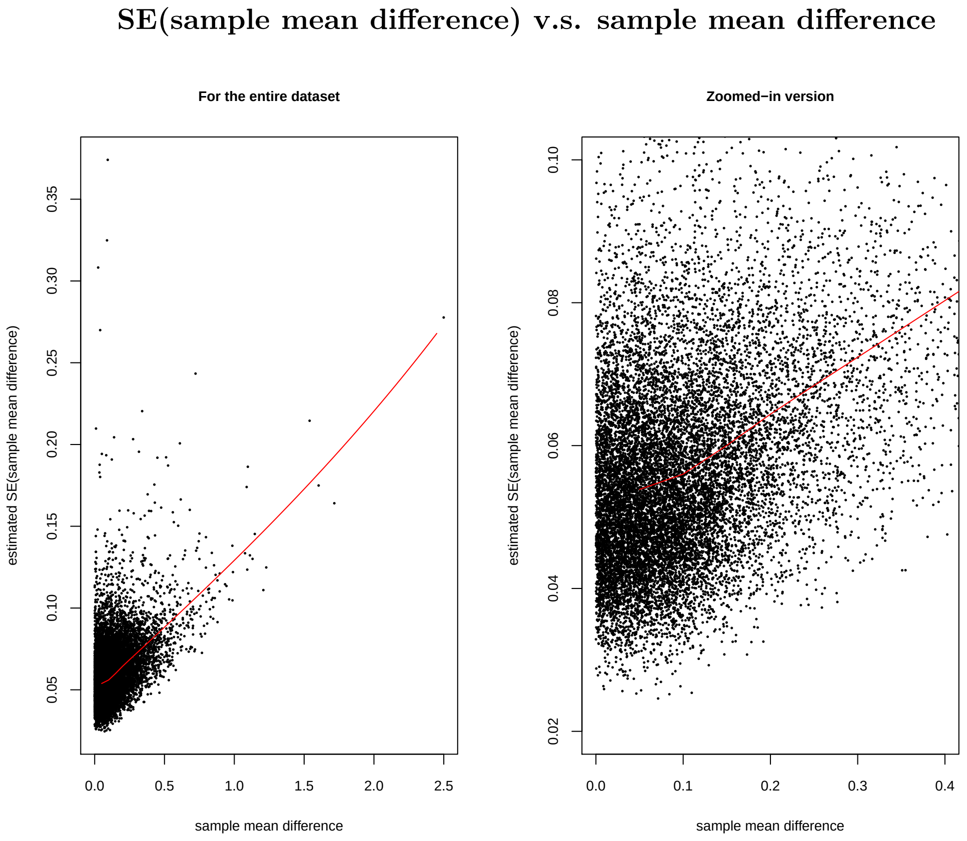

Figure 5.2: Scatterplot for estimated SE v.s. sample mean difference for gene expressions

5.4 Results

Assessing Consistency

As described in Section 5.2 we split the data randomly into two halves and generate J pseudo rank lists from each half. We iterate this process 400 times. The values that we consider for J are 5, 10, 20, and 40. For our classifier we need to select an estimated value for the percent of DE genes. Previous literature suggest the range for this number to be 2.5-10% [8, 55, 39]. Figure 5.3 shows the values of the consistency index for each method when we estimate the percent of DE genes to be 5%; we calculate the consistency index for the top-k subsets for $k \leq 1000$ since the values for k greater than 1000 are not practically interesting.

All three methods improve significantly as the number of lists increases. Similar to the simulation results the relative performance of our method increases with the number of lists, and our method outperforms the other two classifier substantially the number of lists is large.

Consistency Index for the Classifiers

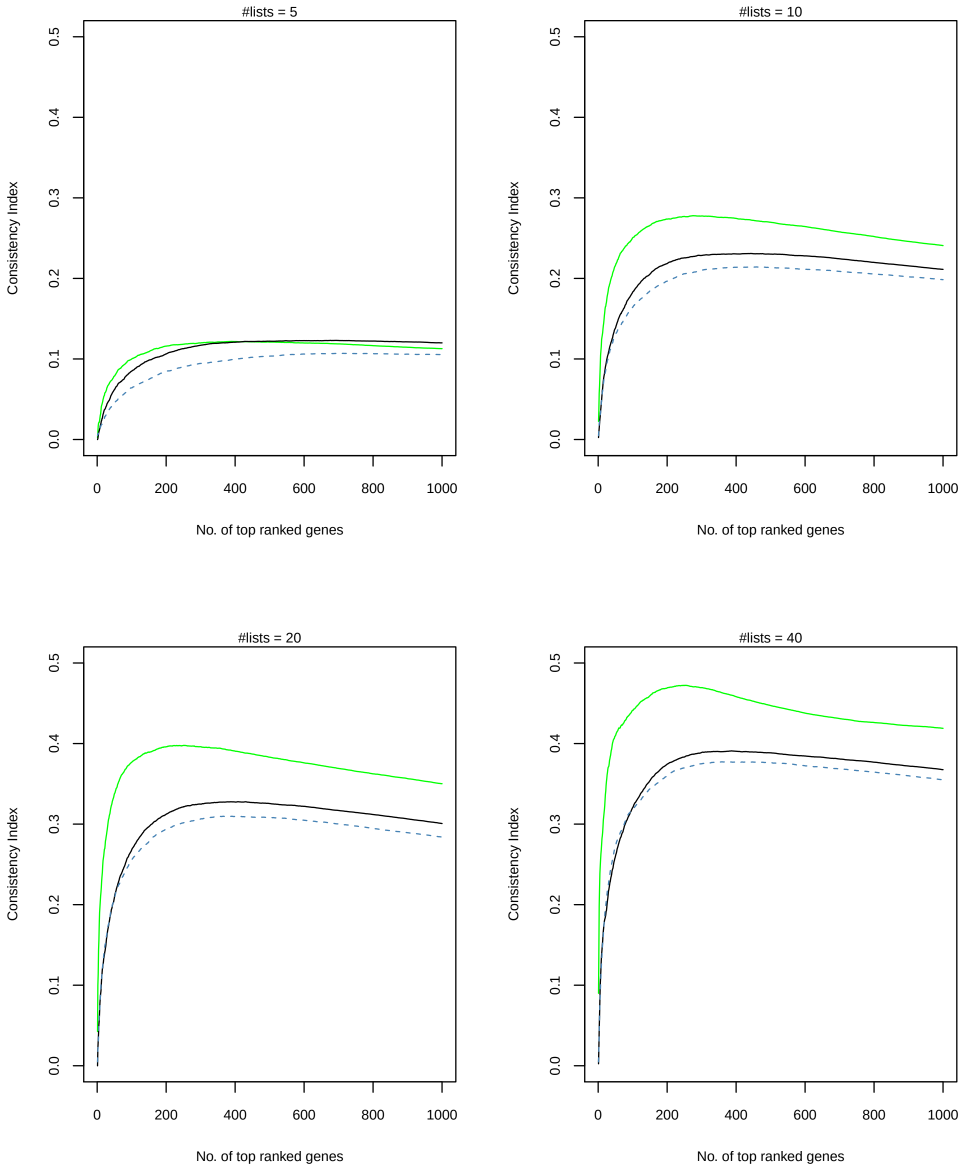

Figure 5.3: Solid green: PR-Ranker, dashed steel blue: SF, solid black: Borda

Figure 5.4 shows the consistency index when 40 lists are used for our method when the value of the estimated fraction of DE genes is 0.025, 0.05 and 0.10. Note that the choice of the estimated value for the fraction of the DE genes does not affect the performance of our classifier much. This shows the robustness of our classifier.

Consistency Index for Our Classifier

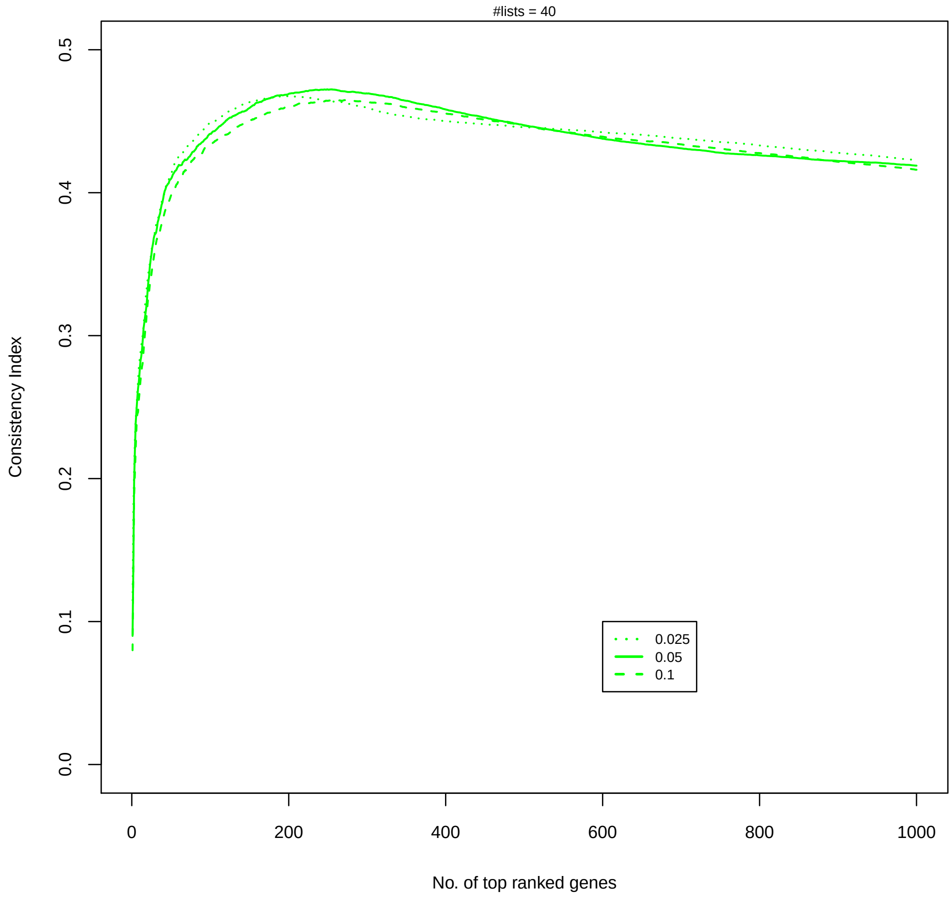

Figure 5.4

Assessing Sensitivity

In this section we generate 40 rank lists, each is done by sampling 4 control and 4 treatment samples from the original dataset; then, we ask each classifier to aggregate the 40 rank lists. We repeat this process 1000 times. We record the genes that show up on the top-20 of the list. Figure 5.5 displays the genes that show up on the top-20 of the aggregated list at least 500 times, i.e., at least half of the time, for each classifier. There are 16 such genes for our classifier, 15 for Borda and 14 for SF. We display these genes along with the frequencies they show up in the top-20 list.

There are 10 genes (FLT1, FSTL3, FLNB, NDRG1, SASH1, TPBG, SH3PXD2A, OCRL, P4HA1, COL17A1) that are are commonly selected by all three methods, and these genes are colored with red color in Figure 5.5.

Among these 10 commonly selected genes, the association to preeclampsia has been biologically corroborated for 7 (FLT1, FSTL3, FLNB, NDRG1, SASH1, SH3PXD2A, COL17A1) of them [40, 20, 2, 49]. There is also evidence that supports the up-regulation of TPBG [43]. We cannot find strong evidence that links OCRL to preeclampsia and there are mixed views on whether P4HA1 is linked to the disorder [30, 25].

We will focus our attention on the 8 commonly selected genes whose associations to preeclampsia are supported by previous publications (i.e., all 10 commonly selected genes, except OCRL and P4HA1). From Figure 5.5 we see that over 80% of the time our classifier ranks 7 of these 8 genes in the top-20. In comparison, Borda captures only 4, and SF only 3, of the 8 genes over 80% of the time. In addition, for 4 (FLT1, FSTL3, FLNB, NDRG1) of the confirmed genes our method selects each of them to be in the top-20 list with an overwhelmingly high probability (over 95% chance); in fact, our classifier selects gene FLT1 100% of the time and FLT1 is well-known to be associated with preeclampsia [40]. In contrast, Borda captures only 3 and SF only 1 of the 8 confirmed genes over 95% of the time. This shows that our method has higher sensitivity compared to the other two methods.

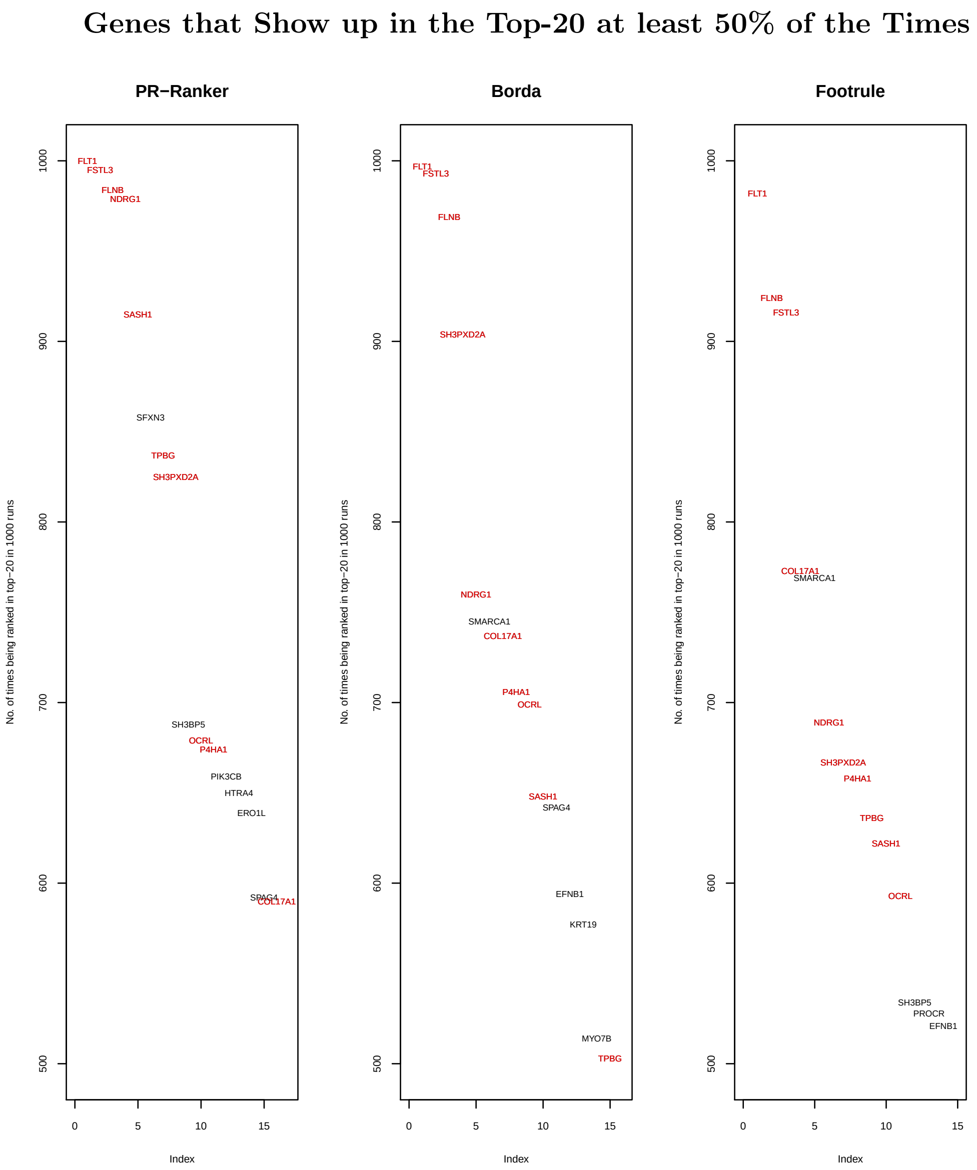

Figure 5.5: Genes that are commonly selected by all three classifiers are in red

Gene ID	Ours	Borda	SF
EFNB1		●	●
ERO1L	●		
HTRA4	●		
KRT19		●	
MYO7B		●	
PIK3CB	●		
PROCR			●
SFXN3	●		
SH3BP5	●		●
SMARCA1		●	●
SPAG4	●	●	

Table 5.1: Top-20 genes that are selected by only some of the classifiers

Additional Findings

Table 5.1 lists the genes that are selected by at least one method but not by all. For example, gene EFNB1 is selected by Borda and SF but not by our classifier.

Four genes (ERO1L, HTRA4, PIK3CB, SFXN3) are detected by our classifier but not by Borda and SF.

An extensive literature links the role of HTRA4 in early onset preeclampsia [62, 27, 61]; early-onset preeclampsia is associated with wide-spread endothelial injury and dysfunction and high levels of HTRA4 has been shown to impede endothelial proliferation and repair [62, 27, 61]. The exact role of gene ERO1L in the Pathogenesis of Preeclampsia is not as well understood as that of HTRA4, but it is believed that the gene plays a role in fetal energy metabolism and is linked to preeclampsia [33]. We cannot find biological experiments that study the role of PIK3CB in preeclampsia; however, the gene was detected in a number of significant pathways in an enrichment analysis for the hypertension disorder [58]. A review of the literature did not find previous links between SFXN3 and preeclampsia.

In this chapter we apply the classifiers of interest to a gene expression dataset to understand the hypertension disorder, preeclampsia. We observe that our classifier produces more consistent results in 400 pseudo-experiments; in particular, the consistency of our classifier grows in a substantially faster rate than the other two classifiers.

In addition, we study the top-ranked genes selected by the classifiers in 1000 pseudo-experiments. 10 genes (FLT1, FSTL3, FLNB, NDRG1, SASH1, TPBG, SH3PXD2A, OCRL, P4HA1, COL17A1) were selected by all classifiers. Except for OCRL and P4HA1, the links to preeclampsia have been established in biological studies. Our classifier shows higher sensitivity since it selects the 8 biologically confirmed genes with noticeably higher frequencies.

Therefore, we conclude that our classifier is more superior than the other two classifier in terms of consistency and sensitivity.

Chapter 6

Conclusion

The capacity of studying the expressions of tens of thousands of genes simultaneously has led to remarkable discoveries in biological sciences. One of the major challenges in gene expression statistical analysis is induced by two intrinsic properties of a typical gene expression dataset: the small number of replicates and the large number of genes. While traditionally gene expression datasets were analyzed one at a time, the opportunity for improving statistical analysis accuracy arose when recently a growing number of gene expression datasets measured under the same sets of biological or experimental conditions have become available. Due to the different technologies that were used in studies it is common that the datasets generated from various studies are not directly comparable. Aggregating study results through rank statistics is a favorable approach because rank statistics are scale invariant, require few distributional assumptions and are more robust in general.

As shown in the Simulation Study Chapter, we see that our classifier, PR-Ranker, has a lower misclassification rate compared to other rank aggregation methods, such as Borda and SF. The strength of our classifier is manifested in situations where some of the rank lists are of low quality. Previous literature has proposed applying a rank truncation procedure on the original rank lists as a preprocessing step to try to improve the quality of the aggregated rank. However, the performance of such procedure is cutoff dependent and the optimal cutoff value for the truncation varies depending on the underlying distribution that generates the ranks. In practice, the underlying distribution that generates the ranks is usually unknown, and in this case truncating the original ranks is likely to cause the performance of the classifier to be less stable. Our classifier aggregates complete rank lists and has the ability of automatically adapting to data quality between lists and down-weighing the information from the lists of lower quality. This property of our classifier is particularly valuable in practice since data quality often varies among different experiments and technologies.

In addition, the algorithm of our classifier is constructed in a such a way that it is well-suited to do a theoretical analysis on and we show in the Theoretical Analysis Chapter that asymptotically our classifier has the lowest misclassification rate in the entire class of the GRB classifiers. Besides that, our classifier has an algorithm that is simple to apply in practice.

Lastly, it was noted in the Theoretical Analysis Chapter that our classifier requires modest distributional assumptions. In the Application Chapter we show in a case where the (log transformed) gene expression data is skewed and departs from the Gaussian distribution that our classifier outperforms the other two classifiers in terms of both consistency and sensitivity.

Bibliography

[1] Ackermann, M., Strimmer, K. *A general modular framework for gene set enrichment analysis.* BMC Bioinformatics 10:47 (February 2009).

[2] Apicella, C, . Ruano, Méhats, C , Miralles, C and Vaiman, D. *The Role of Epigenetics in Placental Development and the Etiology of Preeclampsia.* International Journal of Molecular Sciences. 2019-06-11. (2019)

[3] Barry, W., Nobel, A., Wright, F. *Significance analysis of functional categories in gene expression studies: a structured permutation approach.* Bioinformatics 21:9 (2005). 1943-1949.

[4] Benjamini, Y. and Hochberg, Y. *Controlling the false discovery rate: a practical and powerful approach to multiple testing*, Journal of the Royal Statistical Society Series B. 85: 289-300. (1995).

[5] Benjamini, Y. and Yekutieli, Y. *False discovery rate controlling confidence intervals for selected parameters*, Journal of the American Statistical Association 100: 71-80. (2005).

[6] Borda, JC. *Memoire sur les elections au scrutin.* Histoire de l'Academie des Sciences. (1781).

[7] Bradley, R.A. and Terry, M.A. *Rank analysis of incomplete block designs.* I. Biometrika, 39:324-345. (1952).

[8] Brew O, Sullivan MHF, Woodman A. *Comparison of Normal and Pre-Eclamptic Placental Gene Expression: A Systematic Review with Meta-Analysis.* PLoS ONE 11(8): e0161504. https://doi.org/10.1371/journal.pone.0161504. (2016)

[9] Cai, X., Giannakis, G.B. *Identifying Differentially Expressed Genes in Microarray Experiments With Model-Based Variance Estimation.* IEEE 54:6 (June 2006). 2418-2426.

[10] Davis, B., McDonald, D. *An Elementary Proof of the Local Central Limit Theorem* Journal of Theoretical Probability, Vol. 8, pp. 693-701 (1995).

[11] Davis, J and Goadrich, M. *The relationship between precision-recall and roc curves*. In Proceedings of the 23rd ACM International Conference on Machine learning (ICML'06), pages 233-240. (2006).

[12] DeConde, RP., Hawley, S., Falcon, S., Clegg, N., Knudsen, B., and Etzioni, R. *Combining results of microarray experiments: a rank aggregation approach*. Statistical Applications in Genetics and Molecular Biology. 2006; 5:15.

[13] Dembo, A., and Zeitouni, O *Large deviation techniques and applications*. (1998).

[14] Dudoit,S., Yang,Y.H., Callow,M.J. and Speed,T.P. *Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments*. Statistica Sinica., 12(1), 111-139. (2002).

[15] Durrett, Rick. *Probability: theory and examples* Vol. 49. Cambridge university press. (2019)

[16] Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. *Rank aggregation meth- ods for the web*. Proceedings of the 10th International World Wide Web Conference. 613-622. New York. (2001).

[17] Enquobahrie, DA, Meller, M, Rice, K, Psaty, BM, Siscovick, DS, Williams, MA: *Differential placental gene expression in preeclampsia*. Am J Obstet Gynecol., 199 (5): e1-11. (2008)

[18] Ezkurdia I, Juan D, Rodriguez JM, Frankish A, Diekhans M, Harrow J, Vazquez J, Valencia A, Tress ML. *Multiple evidence strands suggest that there may be as few as 19,000 human protein-coding genes*. Human Molecular Genetics. 23 (22): 5866-78. (November 2014). doi:10.1093/hmg/ddu309. PMC 4204768. PMID 24939910

[19] Fawcett, Tom. *An introduction to ROC analysis* Pattern Recognition Letters, 27, 861-874 (2006) .

[20] Founds, SA, Terhorst, LA, Conrad, KP, Hogge, WA, Jeyabalan, A, Conley, YP: *Gene expression in first trimester preeclampsia placenta*. Biol Res Nurs. 13 (2): 134-139. (2011)

[21] Griffiths, Anthony JF, Miller, Jeffrey H , Suzuki, David T , Lewontin, Richard C, Gelbart, William M.*An Introduction to Genetic Analysis.* New York: W. H. Freeman. 7th edition (2000).

[22] Grill, S, Rusterholz, S, Zanetti-Dallenbach, R, et al., *Potential markers of preeclampsia-a review.* Reproductive Biology and Endocrinology, vol. 7, article 70. (2009).

[23] Hall, P, Schimek, M (2012) *Moderate-Deviation-Based Inference for Random Degeneration in Paired Rank Lists.* Journal of the American Statistical Association, 107:498, 661-672, DOI:10.1080/01621459.2012.682539

[24] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome H. *The elements of statistical learning: data mining, inference, and prediction* 2nd edition (2017).

[25] Highet A.R., Khoda S.M., Buckberry S., Leemaqz S., Bianco-Miotto T., Harrington E. et al. (2015) Hypoxia induced HIF-1/HIF-2 activity alters trophoblast transcriptional regulation and promotes invasion. Eur. J. Cell Biol. 94, 589 10.1016/j.ejcb.2015.10.004

[26] Huang, X. and Pan, W. *Comparing three methods for variance estimation with duplicated high density oligonucleotide arrays.* Funct. Integr. Genomics, vol. 2, pp. 126-133, 2002.

[27] Inagaki, A., Nishizawa, H., Ota, S. et al., *Upregulation of HtrA4 in the placentas of patients with severe pre-eclampsia.* Placenta, vol. 33, no. 11, pp. 919-926. (2012)

[28] Ji, H., Liu, X. Shirley. *Analyzing 'Omics Data Using Hierarchical Models.* Nat Botechnol. April 2010. 28(4):337-340.

[29] Jiang, Z. and Gentleman, R. *Extensions to gene set enrichment.* Bioinformatics 23:3 (2007). 306-313.

[30] Jin X, Xu Z, Cao J, et al. Proteomics analysis of human placenta reveals glutathione metabolism dysfunction as the underlying pathogenesis for preeclampsia. Biochim Biophys Acta. 2017;1865:1207-1214. doi: 10.1016/j.bbapap.2017.07.003.

[31] Kim, S. and Volsky, D. *PAGE: parametric analysis of gene set enrichment.* BMC Bioinformatics 6:144 (June 2005).

[32] Kleinrouweler CE, van Uitert M, Moerland PD, Ris-Stalpers C, van der Post JA, A nk GB. *Differentially expressed genes in the preeclamptic placenta: a systematic review and meta-analysis.* PLoS One. 8:e68991. (2013)

[33] Kobayashi, H. *The Impact of Maternal-Fetal Genetic Conflict Situations on the Pathogenesis of Preeclampsia.* Biochem Genet. 53:223-234. (2015)

[34] Kuncheva, L I, *A stability index for feature selection*, in Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications. Anaheim, CA, USA: ACTA Press, pp. 390-395. (2007).

[35] Leavey, K, Benton, SJ, Grynspan, D, Kingdom, JC, Bainbridge, SA, Cox, BJ. *Unsupervised placental gene expression profiling identifies clinically relevant subclasses of human preeclampsia.* Hypertension, 68:137-47. (2016)

[36] Lie, RT, Rasmussen, S, Brunborg, H, Gjessing, HK, Lie-Nielsen, E, Irgens, LM. *Fetal and maternal contributions to risk of pre-eclampsia: population based study.* BMJ, 316:1343-7. (1998)

[37] Lin, S *Space Oriented Rank-Based Data Integration.* Stat Appl Genet Mol Biol 9:Article20. (2010)

[38] Lin, S., Ding, J. *Integration of ranked lists via cross entropy Monte Carlo with applications to mRNA and microRNA studies.* Biometrics. 2009; 65:9-18.

[39] Liu, JL, Zhang, WQ, Huang, MY. *Transcriptional signature of the decidua in preeclampsia.* Proc Natl Acad Sci USA 115:E5434-E5436. (2018)

[40] McGinnis, R, Steinthorsdottir, V, et al. *Variants in the fetal genome near FLT1 are associated with risk of preeclampsia.* Nat Genet. 49:1255. (2017)

[41] Moore, Lisa D., Le, Thuc T., Fan Guoping. *DNA Methylation and Its Basic Function.* Neuropsychopharmacology. (2013).

[42] Myatt L. Role of placenta in preeclampsia. Endocrine. 2002;19: 103-111. pmid:12583607

[43] Okazaki, S, Sekizawa, A, Purwosunu, Y, Farina, A, Wibowo, N, Okai, T. *Placenta-derived, cellular messenger RNA expression in the maternal blood of preeclamptic women.* Obstet Gynecol 2007; 110: 1130-6.

[44] Opgen-Rhein, Rainer, Strimmer, Korbinian. *Accurate Ranking of Differentially Expressed Genes by a Distribution-Free Shrinkage Approach.* Statistical Applications in Genetics and Molecular Biology (2007) Vol. 6, Issure 1.

[45] Oron, A., Jiang, Z., Gentleman, R. *Gene set enrichment analysis using linear models and diagnostics.* Bioinformatics 24:22 (November 2008). 2586-2591.

[46] Pihur, V., Datta,S., Datta, S. *Finding cancer genes through meta-analysis of microarray experiments: Rank aggregation via the cross entropy algorithm* in 7th International Conference for the Critical Assessment of Microarray Data Analysis, ser. CAMDA'07, Dec. (2007)

[47] Rhodes, D.R., Barrette, T.R., Rubin, M.A., Ghosh,D. and Chinnaiyan,A.M. *Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer.* Cancer Res., 62(15):4427- 4433. (2002)

[48] Rhodes, D.R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Bar- rette, T., Pandey, A. and Chinnaiyan, A.M. *Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression.* Proc Natl Acad Sci USA., 101(25):9309-9314. (2004)

[49] Serebrova, VN, Trifonova, EA, Gabidulina, TV., et al. *Detection of novel genetic markers of susceptibility to preeclampsia based on an analysis of the regulatory genes in the placental tissue.* Mol Biol (Mosk). 50:870-9. (2016)

[50] Šidák, Z. (1967). Rectangular confidence region for the means of multivariate normal distributions. Journal of the American Statistical Association, 62, 626-633.

[51] Storey, J. *A direct approach to false discovery rates*, Journal of the Royal Statistical Society B. 64(3): 479-498. (2002).

[52] Storey, J., Taylor, J. and Siegmund, D. *Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach*, Journal of the Royal Statistical Society, Series B 66: 187-205. (2004).

[53] Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., et al. *Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles.* PNAS 102:43 (October 2005).

[54] Tai, Yu Chuan and Speed, Terence P. *A Multivariate Empirical Bayes Statistic for Replicated Microarray Time Course Data*. Ann. Statist. (2006).

[55] Tejera, E, Bernardes, J, Rebelo, I. *Co-expression network analysis and genetic algorithms for gene prioritization in preeclampsia*. BMC Med Genomics 6:51. (2013)

[56] Tian, L., Greenberg, S., Kong, S., Altschuler, J., Kohane, I., Park, P. *Discovering statistically significant pathways in expression profiling studies.* PNAS 102:38 (September 2005). 12544-13549.

[57] van der Vaart, Aad W. *Asymptotic statistics.* Vol. 3. Cambridge university press. (2000)

[58] Vaiman, D, Miralles, F. *An Integrative Analysis of Pre-eclampsia Based on the Construction of an Extended Composite Network Featuring Protein-Protein Physical Interactions and Transcriptional Relation ships*. Plos One. November 1. (2016)

[59] Villar J, Say L, Gulmezoglu AM, Meraldi M, Lindheimer MD, Betran AP, Piaggio G; *Eclampsia and preeclampsia: a health problem for 2000 years.* In Preeclampsia, Critchly H, MacLean A, Poston L, Walker J, eds. London, RCOG Press, pp 189-207. (2003)

[60] Wald, R., Khoshgoftaar, T., Dittman, D., Awada, W. and Napolitano, A. *An Extensive Comparison of Feature Ranking Aggregation Techniques in Bioinformatics* IEEE IRI 2012, August 8-10, 2012, Las Vegas, Nevada, USA 377 978-1-4673-2284-3/12/ 2012 IEEE

[61] Wang, Y, Lim, R, Nie, G. *HtrA4 may play a major role in inhibiting endothelial repair in pregnancy complication preeclampsia*. Sci Rep. PMID: 30804477 (2019)

[62] Wang, Y., Nie, G. *High levels of HtrA4 observed in preeclamptic circulation drastically alter endothelial gene expression and induce inflammation in human umbilical vein endothelial cells*. Placenta 47, 46-55. (2016).

Appendix A

Appendix

A.1 Proof of Propositions 3.6.1 and 3.6.6

In this section we will establish the proof of Proposition 3.6.1 which will follow from the next lemma.

Let $\mathcal{F}_r^j$ denote the sigma-algebra generated by $\{U_1^j, ..., U_r^j; A_1^j..., A_r^j\}$. Also define the reversed sigma-algebra denoted $\mathcal{F}_r^{j*}$ generated by $\{U_r^j, ..., U_n^j; A_r^j..., A_n^j\}$ where A_r^j is the indicator that the gene ranked r in list j is DE and U_r^j is the r-th order statistic of the t-statistics for list j.

Lemma A.1.1. *The conditional probability of being DE for ranked genes satisfies*

$$\max_{1 \leq r \leq 2n/3} |\mathbb{P}[A_r \mid \mathcal{F}_{r-1}^j] - p^{j*}(r/n)| \to 0 \tag{A.1}$$

$$\max_{n/3 \leq r \leq n} |\mathbb{P}[A_r \mid \mathcal{F}_{r+1}^{j*}] - p^{j*}(r/n)| \to 0 \tag{A.2}$$

in probability as $n \to \infty$.

Proof. We will prove the first of these two equations, the second will follow similarly. Let $S_r = dn - \sum_{\ell=1}^{r-1} 1_{A_\ell}$, that is the number of DE genes associated with genes ranked r through n. Applying Bayes rule we have,

$$\begin{aligned}&\mathbb{P}(A_r|\mathcal{F}^j_{r-1}, U_r \in dt)\\ =&\mathbb{P}(A_r|\mathcal{F}^j_{r-1}, U_r \geq t, U_r \in dt)\\ =&\frac{\mathbb{P}(A_r, U_r \in dt|\mathcal{F}^j_{r-1}, U_r \geq t)}{\mathbb{P}(A_r, U_r \in dt|\mathcal{F}^j_{r-1}, U_r \geq t) + \mathbb{P}(A^c_r, U_r \in dt|\mathcal{F}^j_{r-1}, U_r \geq t)}\\ =&\frac{S_r\widetilde{\phi}(t)/(1-\widetilde{F}(t))dt}{S_r\widetilde{\phi}(t)/(1-\widetilde{F}(t))dt + (n-(r-1)-S_r)\phi(t)/(1-F(t))dt}.\end{aligned}$$

We define the function

$$y(s,t,\alpha) = \frac{s/(1-\widetilde{F}(t))}{s/(1-\widetilde{F}(t)) + (1-\alpha-s)(\phi(t)/\widetilde{\phi}(t))/(1-F(t))}$$

and so

$$\mathbb{P}(A_r|\mathcal{F}_{r-1}, U_r \in dt) = y(S_r/n, t, (r-1)/n).$$

Now, since the variable t is defined on the whole negative real line, we do a change of variables so that $\hat{y}$ is defined on a compact set and is uniformly continuous

$$\hat{y}(s,v,\alpha) = y(s, 1-1/v, \alpha).$$

By our assumptions on the densities, ϕ and $\widetilde{\phi}$ the function $\hat{y}$ is uniformly continuous on the domain

$$\mathcal{D}_\epsilon := \{(s,v,\alpha) : 0 \leq s, v, \alpha \leq 1-\epsilon, s+\alpha \leq 1-\epsilon\}. \tag{A.3}$$

for any $\epsilon > 0$.

Therefore, we have that

$$\begin{aligned}\mathbb{P}(A_r|\mathcal{F}^j_{r-1}) =& \int_{U_{i-1}}^0 \mathbb{P}(A_r, U_r \in dt|\mathcal{F}^j_{r-1})dt\\ =&\int_{U_{i-1}}^0 \mathbb{P}(U_r \in dt|\mathcal{F}^j_{r-1})y(S_r/n, t, (r-1)/n)dt\\ =&\mathbb{E}[y(S_r/n, U_r, (r-1)/n)|\mathcal{F}^j_{r-1})\\ =&\mathbb{E}\left[\hat{y}(S_r/n, \frac{1}{1-U_r}, (r-1)/n)|\mathcal{F}^j_{r-1}\right]\end{aligned}$$

Let G^* denote the CDF of $\frac{1}{1-T^*_i}$ which are bounded random variables with full support on $[0,1]$. By the Glivenko-Cantelli Theorem [15] G^*_n, the ECDF of the T_i^* converges almost surely uniformly to G^* since it is a mixture of two i.i.d. sequences of random variables. Following the approach of the proof in [57] we can also show that $G_n^{*-1} \to G^{*-1}$ almost surely uniformly. It follows that

$$\max_r |\frac{1}{1-U_r} - G^{*-1}(r/n)| \to 0$$

almost surely. In fact a stronger quantitative statement from the large deviation theory for empirical distribution functions says that for any fixed $\delta > 0$, for some $c(\delta) > 0$,

$$\mathbb{P}\left[|\frac{1}{1-U_r} - G^{*-1}(r/n)| > \delta\right] \leq \exp(-c(\delta)n).$$

Thus

$$\mathbb{P}\left[\mathbb{P}\left[|\frac{1}{1-U_r} - G^{*-1}(r/n)| > \delta \mid \mathcal{F}_{r-1}^j\right] > n^{-2}\right] \leq n^2 \exp(-c(\delta)n).$$

and so for any $\delta > 0$,

$$\max_r \mathbb{P}\left[|\frac{1}{1-U_r} - G^{*-1}(r/n)| > \delta \mid \mathcal{F}_{r-1}^j\right] \to 0$$

almost surely. Similarly

$$\max_r \left|\frac{S_r}{n} - (1 - \widetilde{F}(F^{*-1}(r/n)))\right| \to 0$$

almost surely. It follows that

$$\max_r \left|\mathbb{E}\left[\hat{y}(S_r/n, \frac{1}{1-U_r}, (r-1)/n)|\mathcal{F}_{r-1}\right] - \hat{y}(1 - \widetilde{F}(F^{*-1}(r/n)), G^{*-1}(r/n), \frac{r}{n})\right| \to 0.$$

Now

$$\begin{aligned}\hat{y}(1 - \widetilde{F}(F^{*-1}(r/n)), G^{*-1}(r/n), \frac{r}{n}) &= y(1 - \widetilde{F}(F^{*-1}(r/n)), F^{*-1}(r/n), \frac{r}{n}) \\ &= \frac{d\widetilde{\phi}(F^{*-1}(r/n))}{d\widetilde{\phi}(F^{*-1}(r/n)) + (1-d)\phi(F^{*-1}(r/n))} \\ &= p^{j*}(r/n)\end{aligned}$$

which completes the proof. □

Proof of Proposition 3.6.1.
The proof follows by averaging over the conditioning.

Proof of Proposition 3.6.6.

Consider the probability distribution $\mathbb{Q}$ where instead of having exactly dn DE genes, each gene is DE independently with probability d so $N = \sum_i 1_{B_i}$, the total number of DE genes, has distribution $\text{Bin}(n, d)$. Then we have that

$$\mathbb{P}(\cdot) = \mathbb{Q}(\cdot \mid N = dn),$$

that is conditional on there being exactly dn DE genes under $\mathbb{Q}$, it is the same model as the original $\mathbb{P}$. The advantage of $\mathbb{Q}$ is that each gene is independent. Note that under $\mathbb{Q}$ we have that

$$\mathbb{Q}[B_i \mid \mathcal{G}] = \mathbb{Q}[B_i \mid \mathcal{G}_i] = \xi_i.$$

Now

$$\mathbb{P}[B_i \mid \mathcal{G}] = \mathbb{Q}[B_i \mid \mathcal{G}, N = dn]$$

and so if $N_{-i} = N - 1_{B_i}$, the number of DE genes apart from i, then by Bayes rule

$$\begin{aligned}
&\mathbb{Q}[B_i \mid \mathcal{G}, N = dn] \\
&\quad = \frac{\mathbb{Q}[B_i \mid \mathcal{G}]\mathbb{Q}[N = dn, \mid \mathcal{G}, B_i]}{\mathbb{Q}[B_i \mid \mathcal{G}]\mathbb{Q}[N = dn, \mid \mathcal{G}, B_i] + \mathbb{Q}[B_i^c \mid \mathcal{G}]\mathbb{Q}[N = dn, \mid \mathcal{G}, B_i^c]} \\
&\quad = \frac{\xi_i \mathbb{Q}[N_{-i} = dn - 1, \mid \mathcal{G}]}{\xi_i \mathbb{Q}[N_{-i} = dn - 1, \mid \mathcal{G}] + (1 - \xi_i)\mathbb{Q}[N_{-i} = dn, \mid \mathcal{G}]}.
\end{aligned}$$

Hence if we prove that for any $\epsilon > 0$,

$$\mathbb{P}\Big(\Big|\frac{\mathbb{Q}[N_{-i} = dn - 1, \mid \mathcal{G}]}{\mathbb{Q}[N_{-i} = dn, \mid \mathcal{G}]} - 1\Big| > \epsilon\Big) \to 0, \tag{A.4}$$

then we will show that

$$\mathbb{P}(|\mathbb{Q}[B_i \mid \mathcal{G}, N = dn] - \xi_i| > \epsilon) \to 0,$$

which will establish the lemma. Let $\mathcal{B}$ be the σ-algebra generated by the $\{B_i\}_{i=1,\dots,n}$, the information on which genes are differentially expressed. Conditional on $\mathcal{B}$ the ξ_i are conditionally independent. Define

$$\mu := \mathbb{E}[\xi_i \mid B_i^c], \qquad \widetilde{\mu} := \mathbb{E}[\xi_i \mid B_i].$$

Then

$$d = \mathbb{E}[\xi_i] = (1 - d)\mu + d\widetilde{\mu}.$$

and

$$\mathbb{E}[\xi_i \mid \mathcal{B}] = \mu 1_{B_i^c} + \widetilde{\mu} 1_{B_i}.$$

Setting $S := \sum_i \xi_i$ we then have that $\mathbb{E}[S \mid cB] = \sum_i \mathbb{E}[\xi_i \mid \mathcal{B}] = dn$. By the Azuma-Hoeffding inequality for any t,

$$\mathbb{P}\Big(\Big|S - dn\Big| > t\sqrt{n} \mid \mathcal{B}\Big) \leq 2\exp(-t^2/2),$$

and hence unconditionally

$$\mathbb{P}\Big(\Big|S - dn\Big| > t\sqrt{n}\Big) \leq 2\exp(-t^2/2),$$

By our assumption that $0 < c_1 \leq \frac{\phi(t)}{\widetilde{\phi}(t)} < c_2$ we have that for some $\delta > 0$ that $\delta < \xi_i < 1-\delta$ for all i. Setting $\sigma^2 = \sum \xi_i(1-\xi_i)$ we have that $\sigma^2 \geq n\delta(1-\delta)$. Define $S_{-i} = S - \xi_i$ and $\sigma^2_{-i} = \sigma^2 - \xi_i(1-\xi_i)$. Under $\mathbb{Q}$, conditional on $\mathcal{G}$ the 1_{B_i} are independent Bernoulli random variables with probabilities ξ_i. Hence by the Local Central Limit Theorem (see [10] Theorem 1.1) we have that

$$\mathbb{Q}(N_{-i} = k, \mid \mathcal{G}) = \frac{1}{\sqrt{2\pi}\sigma_{-i}} \exp\big(-(k - S_{-1})/(2\sigma^2_{-i})\big) + o(n^{-1/2})$$

as $n \to \infty$. For any fixed t, for large enough n when $|S_{-1} - dn| \leq \sqrt{n}t + 1$ we have that

$$\Big|\frac{\mathbb{Q}[N_{-i} = dn-1, \mid \mathcal{G}]}{\mathbb{Q}[N_{-i} = dn, \mid \mathcal{G}]} - 1\Big| \leq \epsilon$$

Hence

$$\begin{aligned}\limsup_n \mathbb{P}\Big(\Big|\frac{\mathbb{Q}[N_{-i} = dn-1, \mid \mathcal{G}]}{\mathbb{Q}[N_{-i} = dn, \mid \mathcal{G}]} - 1\Big| > \epsilon\Big) &\leq \limsup_n \mathbb{P}(|S_{-1} - dn| > \sqrt{n}t + 1) \\ &\leq \limsup_n \mathbb{P}(|S - dn| > t\sqrt{n}) \\ &\leq \exp(-t^2/2).\end{aligned}$$

This holds for all $t > 0$ which establishes equation (A.4) and hence the lemma.

A.2 Large Deviations and Cramer's Theorem

The proofs of the main theorems make crucial use of the theory of large deviations to establish the optimal error rates for our rank based estimators. Here we recall some basic results, see for instance [13]. Let $X_1, X_2, \ldots$ be a sequence of independent and identically distributed random variables with measure μ. Then Cramer's Theorem gives the probability that the partial sum $S_n = \sum_{i=1}^n X_i$ is much greater than its expected value.

Theorem A.2.1 (Cramer's Theorem). *If for some* $\delta > 0$, $\mathbb{E}[\exp(\delta X_i)] < \infty$ *then there exists a function* $\phi(z)$ *such that for all* $z > \mathbb{E}[X_1]$ *we have that*

$$\phi(z) = \lim_n \frac{1}{n} \log \mathbb{P}[\frac{1}{n} S_n > z] = \inf_{\theta > 0} \log \mathbb{E}[\exp(\theta X_i)] - \theta z < 0. \qquad \text{(A.5)}$$

The large deviation rate function ϕ can be expressed in terms of the *relative entropy*. The relative entropy of μ' with respect to μ is denoted by defined as

$$H(\mu'|\mu) = \int \log \left(\frac{d\mu'(x)}{d\mu(x)} \right) d\mu'(x).$$

The rate function $\phi(x)$ can be written as

$$\phi(z) = - \inf_{\mu' : \int x d\mu'(x) \geq z} H(\mu' \mid \mu)$$

that is the smallest relative entropy over all measures μ' with mean at least z. When the random variable X is almost surely bounded with essential supremum $M < \infty$, for any $\mathbb{E}[X_i] < z < M$ there exists μ' which achieves the supremum and moreover is given by the Radon-Nikodym derivative

$$\frac{d\mu'(r)}{d\mu(r)} = \frac{\exp(\theta_z r)}{\mathbb{E}[\exp(\theta_z X_1)]}$$

where θ_z is the θ that achieves the infimum in (A.5). Since relative entropy is strictly convex this is the unique infimum.