2010

Hidden Markov models for simultaneous testing of multiple gene sets and adaptive and dynamic adaptive procedures for false discovery rate control and estimation

Kun Liang
Iowa State University

Hidden Markov models for simultaneous testing of multiple gene sets and adaptive and dynamic adaptive procedures for false discovery rate control and estimation

by

Kun Liang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Dan Nettleton, Major Professor
Song Xi Chen
Jack C. M. Dekkers
Kenneth J. Koehler
Peng Liu

Iowa State University

Ames, Iowa

2010

To my family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my advisor Dan Nettleton from the bottom of my heart. Looking back, it has been my tremendous fortune to have Dan supervise me. He is knowledgeable and kind to everyone, including his students. I learned so many things from working with him. His dedication to research and ability for attention to detail always keep me humble. I also should thank for his belief in my capacity, which I didn't realize I possess.

I thank Drs. Song Xi Chen, Jack Dekkers, Ken Koehler and Peng Liu for serving on my committee and offering me many professional and personal advices.

I am also grateful to Drs. Douglas Bonett, Mack Shelley and Stephen Vardeman. Their dedication to mentoring, teaching and research has always been a constant inspiration for me.

I thank my wife, Yingli, for her unconditional support along the way.

CHAPTER 1. General introduction

1.1 Introduction

With a long history in statistical literature, the topic of multiple testing has generated reviving interest in recent years. The wide availability of large and complex modern datasets, e.g., datasets from genomics, medical imaging, and astronomy among others, is the driving force behind the surging research of multiple testing issues. Nowadays, scientists routinely test thousands of hypotheses simultaneously, and how to make sound and efficient inferences for these hypotheses has never been more important. This chapter briefly overviews the main components of this dissertation, including multiple testing, the false discovery rate (FDR), the adaptive FDR methods and gene categories/sets testing.

1.2 Multiple Testing

A bit more than a decade ago, a multiple testing problem would mean to test at most dozens of hypotheses simultaneously, e.g., the classical problem in linear models for testing which means are different from each other after rejecting the grand null hypothesis that all means are equal. With the advent of many modern scientific measurement tools, we are facing a flood of data and questions behind them. A poster child of modern multiple testing problem is the analysis of microarray data.

Microarrays measure thousands of gene expressions simultaneously to discover genes that are differentially expressed (DE) across different treatment conditions. As a more concrete example, suppose we are using microarrays to measure $m = 20,000$ gene expressions for each of a group of n_1 healthy people and a group of n_2 sick people. If we are interested in finding the genes that are differentially expressed between the two groups, we can test the null hypotheses $H_0^{(i)} : \mu_1^{(i)} = \mu_2^{(i)}$ for $i = 1, \ldots, m$, where $\mu_1^{(i)}$ is the mean expression level of gene i for the healthy people and $\mu_2^{(i)}$ for the sick people. A certain test, e.g., a two-sample t-test, can be used to test each of the $H_0^{(i)}$s and produces a test statistic (a t-statistic or a p-value) for each $H_0^{(i)}$. Then decisions can be made based on the corresponding test statistics (typically the p-values). Given a threshold for significance c, hypotheses with p-values no larger than c are declared significant. This raises the issue of how to adjust the threshold c to the multiplicity of simultaneous hypotheses testing. Historically, many researchers have simply ignored the issue and made no adjustment, and some even advocated this practice. Under modern large-scale simultaneous hypotheses testing situations, no adjustment to multiplicity is clearly not a good practice. Continue with the microarray example and suppose that 1,100 out of the total 20,000 p-values are no larger than 0.05, the commonly used threshold for hypothesis testing if no correction is made for multiple testing. If we choose c to be 0.05, the corresponding 1,100 genes would be declared significant. However, if all 20,000 genes are independent and equivalently expressed, the p-values will each follow the uniform(0,1) distribution, and we would expect $20,000 \cdot 0.05 = 1,000$ p-values to be no larger than 0.05. Thus, the 1,100 "significant" genes could simply happen by chance. Furthermore, if there are 1,100 genes that are truly differentially expressed, we would still expect the number of true null p-values that are no larger than 0.05 to be $(20,000 - 1,100) \cdot 0.05 = 945$, i.e., the majority of the 1,100 "significant" genes.

Different methods to control the number of unwanted false discoveries are introduced in the following section.

1.3 False Discover Rate

One long-standing alternative to the no multiple testing correction is to control the familywise error rate (FWER). To formally introduce the FWER, let us look at a possible result from a multiple testing procedure. Consider the problem of testing simultaneously m null hypotheses, of which m_0 are true nulls. Suppose the p-values associated with the m null hypotheses are $p_1, \ldots, p_m$, respectively; and let $p_{(1)} \leq \ldots \leq p_{(m)}$ denote these p-values in ascending order. According to some threshold for significance c, we reject R hypotheses whose p-values are no larger than c. The result of the multiple testing can be summarized in Table 1.1. Also note that V is the number of type I errors (false positive results) among the total R rejections, and T is the number of type II errors (false negative results).

Table 1.1: An outcome of a multiple-testing procedure.

	Accept null hypothesis	Reject null hypothesis	Total
Null true	U	V	m_0
Alternative true	T	S	m_1
	W	R	m

Then the FWER is defined as

$$\text{FWER} = \Pr(V \geq 1).$$

Commonly used methods to control the FWER in a multiple testing scenario are Bonferroni's method and Holm's method (Holm, 1979). If we want to control FWER at level α, Bonferroni's method will suggest a threshold of α/m, which could be a severe

penalty when m is large. Holm's method offers a slightly more powerful procedure than Bonferroni's method while controlling the FWER at the same level. With $p_{(0)} \equiv 0$, the threshold of Holm's method is $p_{(k)}$, where $k = \max\{j : p_{(l)} \leq \alpha/(m - l + 1)\ \forall l \leq j\}$. Note that Holm's method is a so-called step-down procedure because it scans the p-values in an ascending order and stops just before the first time a certain condition is not met. However, when m is large, the control of FWER becomes a very conservative practice because the penalty is roughly linear in m. Furthermore, the FWER is the probability of making at least one type I error, and thus, it is not a desired quantity to control for large-scale exploratory scientific endeavors. For example, in microarray experiments, scientists will be content to endure the existence of some type I errors so long as the proportion of the type I errors among total rejections remains small.

In a ground-breaking work, Benjamini and Hochberg (1995) (BH) introduced the concept of the false discovery rate (FDR). The FDR is defined as

$$\text{FDR} = \mathbf{E}\left[\frac{V}{R \vee 1}\right].$$

BH also proposed a linear step-up procedure (the BH procedure) to determine a rejection region that can guarantee the FDR to be below a certain threshold α under certain independence conditions. With $p_{(0)} \equiv 0$, the threshold of the BH procedure is $p_{(k)}$, where $k = \max\{j : p_{(j)} \leq j\alpha/m\}$. The BH procedure is a so-called step-up procedure because it can be viewed as scanning the p-values in a descending order and stopping when the first time a certain condition is met. The BH procedure is a conservative method if $m_0 < m$ because it controls FDR at level $\pi_0\alpha$, where $\pi_0 = m_0/m$. In a later work, Benjamini and Yekutieli (2001) showed the BH procedure also controls the FDR at level $\pi_0\alpha$ under certain positive dependence conditions. Intuitively, if we use the BH procedure at the α/π_0 level then the FDR will be controlled at level α. This suggests the development of adaptive methods that adapt to the value of π_0.

1.4 Adaptive FDR Methods

Knowledge of π_0, or equivalently m_0, is very important for improving the performance of FDR controlling procedures. If π_0 or m_0 were given to us by an "oracle", the BH step-up procedure with target level setting at α/π_0 will control the FDR at level α under certain independence or positive dependence conditions. Storey (2002) and Storey et al. (2004) introduced the first adaptive procedure that is theoretically proven to control FDR under certain independence conditions. Their proposed estimator for π_0 is

$$\hat{\pi}_0(\lambda) = \frac{\#\{p_i > \lambda\} + 1}{(1-\lambda)m} = \frac{m - R(\lambda) + 1}{(1-\lambda)m},$$

where $\lambda \in [0, 1)$ is a parameter to be specified and $R(\lambda)$ is the $\#\{p_i \leq \lambda\}$. It can be easily shown that $\hat{\pi}_0(\lambda)$ is a conservative estimator for π_0. Naturally, the next question is how to choose an appropriate or an optimal λ. It is easy to see that the larger the λ, the smaller the conservative bias of $\hat{\pi}_0(\lambda)$ but the bigger the variance. This is a classical bias versus variance situation. Storey (2002) and Storey et al. (2004) proposed a λ estimation procedure based on a bootstrap method. However, the procedure has been shown to be anti-conservative in subsequent simulation studies (Black, 2004; Nettleton et al., 2008).

Since the original paper of Benjamini and Hochberg (1995), many adaptive procedures have been proposed from different perspectives and using different methodologies. As it turns out, many of the procedures that have been theoretically proven to control FDR can be formulated as special cases of the Storey's λ procedure with their λ chosen individually. We call this group of procedures the *fixed adaptive* procedures. For this group of procedures, the λs are pre-specified before the data are seen. Common choices for λs are some constants (e.g., 0.5) or some functions of α, the target

FDR level.

More interestingly, there is a group of procedures can be viewed as special cases of the Storey's λ procedure with their λ dynamically selected, i.e., the λs are determined through some data-dependent procedures. We call this group of procedures the *dynamic adaptive* procedures. Examples include Benjamini and Hochberg (2000) method, Storey's bootstrap method, and Nettleton et al. (2006) method among others. However, no theoretical result is available for the dynamic adaptive procedures. In Chapter 2, we show that a class of these dynamic adaptive methods provides conservative point estimation of π_0 and FDR.

1.5 Gene Set Testing

An important challenge facing researchers is how to interpret and report the results from high throughput transcriptome experiments, for example, microarray and RNA-seq experiments. A routine analysis, e.g., a two sample t-test for each gene on a microarray, can produce a list of genes that are declared to be differentially expressed across conditions. The length of the DE gene list can run up to a few thousand, and this makes the interpretation and reporting of the results a challenging task. To interpret the results of such an analysis, researchers study the characteristics of the genes on the DE list as known from past research. Known characteristics of genes may include the molecular function of a gene, the biological process in which the gene operates, or the component of the cell in which the gene product is known to be found. Such information is formalized in the ontologies developed as part of the Gene Ontology (GO) project (Ashburner et al., 2000).

Rather than testing individual genes for differential expression, it has become a common practice for scientists to test whether some predefined gene categories/sets

are differentially expressed. Many statistical methods have been proposed for this purpose. Among them, many of the early developed ones are based on test statistics derived from individual genes. These methods have subsequently been reviewed and criticized on statistical grounds by many authors. The criticism can be summarized as follows: First, the majority of these methods assume the unrealistic assumption of gene independence, e.g., the methods using Fisher's exact test or its variants for testing independence between the memberships to the DE gene list and to a certain gene set; Second, most of these methods are based on the competition between genes within and outside a gene set, which may not be the main interest of biological research; Third, tests based on single gene statistics do not process power to detect gene set multivariate distributional differences across conditions that are beyond marginal differences. In recent years, a viable alternative has been rapidly developed. Many authors have proposed methods to test multivariate gene set differences. The multivariate tests avoid the unrealistic assumption of gene independence and are potentially more powerful than the individual gene tests combined.

1.6 Multiple Gene Sets Testing

Unless a researcher tests only one pre-specified gene set, the multiplicity arising from testing multiple gene sets should be accounted for. Many microarray and RNA-seq experiments are exploratory in nature, and thus, a systematic way of testing multiple gene sets is needed. Furthermore, testing gene sets derived from Gene Ontology is equivalent to testing hundreds of null hypotheses that correspond to nodes in a directed acyclic graph. The logical relationships among the nodes in the graph imply that only some configurations of true and false null hypotheses are possible. We show in Chapter 3 and 4 how to take advantage of these logical restrictions and improve

inferences. In a sense, we are adapting to the underlying dependence structure of null hypotheses.

1.7 Organization

The core idea of the adaptive multiple testing strategies are seen throughout this thesis. The second chapter develops a series of theoretical results for the adaptive FDR control methods through an empirical processes point of view. In particular, we study the finite sample and asymptotic properties of the methods that adapt to the proportion of true null hypotheses.

The third chapter introduces a hidden Markov model on the underlying null hypotheses when testing multiple gene sets from Gene Ontology. We develop a Markov chain Monte Carlo scheme to estimate the posterior probability of each null hypotheses being false and show in simulation that our method is superior to other existing methods.

The fourth chapter explores more efficient implementation based on the framework introduced in Chapter 3. Through the use of a hidden Markov tree model and the deterministic annealing EM algorithm, we develop a more inferentially powerful and computationally efficient method. The thesis concludes with Chapter 5, which also gives an overview of future work.

Bibliography

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al. (2000). Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57(1):289–300.

Benjamini, Y. and Hochberg, Y. (2000). On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25(1):60.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 29(4):1165–1188.

Black, M. (2004). A note on the adaptive control of false discovery rates. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 297–304.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(65-70):1979.

Nettleton, D., Hwang, J., Caldo, R., and Wise, R. (2006). Estimating the number of true null hypotheses from a histogram of p values. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(3):337–356.

Nettleton, D., Recknor, J., and Reecy, J. M. (2008). Identification of differentially expressed gene categories in microarray studies using nonparametric multivariate analysis. *Bioinformatics*, 24(2):192–201.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 64(3):479–498.

Storey, J., Taylor, J., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187–205.

CHAPTER 2. Adaptive and Dynamic Adaptive Procedures for False Discovery Rate Control and Estimation

A paper to be submitted to *Journal of the Royal Statistical Society Series B*

Kun Liang and Dan Nettleton

Department of Statistics

Iowa State University, Ames, IA 50011

email: liangkun@iastate.edu

Abstract

Storey (2002) advocated the approach of conservative point estimation of false discovery rate (FDR) and showed that it is closely related to the classical Benjamini and Hochberg (1995) procedure that controls FDR. Storey et al. (2004) further proved that a related procedure strongly controls FDR through the elegant use of empirical processes. Since then, many FDR control procedures have been developed. In light of these new developments, we continue Storey's work on the conservative point estimation of FDR. We first give a corrected version of the finite sample proof for conservative point estimation of FDR. Then we survey existing theoretically proven FDR control procedures and show that some of these procedures can be cast as plug-in procedures with their estimators of the proportion of true null hypotheses (π_0) more conservative

than that of Storey et al. (2004). Thus, their FDR control properties can be easily proved under conditions weaker than those previously considered. More importantly, we established a condition under which a dynamic adaptive procedure will lead to conservative π_0 and FDR estimators. This result covers some important procedures and will guide the future development of adaptive procedures. Finally, we extend the asymptotic results of Storey et al. (2004) to a larger class of procedures.

KEY WORDS: Empirical processes; Martingales; Multiple testing; Optional Stopping Theorem; Simultaneous inference; Stopping time.

2.1 Introduction

Multiple testing has generated a surging interest in recent years due to the wide availability of large and complex modern data sets. The traditional familywise error rate (FWER) is often considered to be conservative, especially when investigations are exploratory. Benjamini and Hochberg (1995) (BH) introduced the concept of false discovery rate (FDR), which has since spurred much research in statistics.

Consider the problem of testing simultaneously m null hypotheses, of which m_0 are true nulls. Suppose the p-values associated with the m null hypotheses are $p_1, \ldots, p_m$, respectively; and let $p_{(1)} \leq \ldots \leq p_{(m)}$ denote these p-values in ascending order. According to a certain threshold for significance c, suppose we reject R hypotheses whose p-values are no larger than c. The result of the multiple testing is summarized in Table 1.1.

FDR can be roughly understood as the expected proportion of false discoveries (type I errors) among all the discoveries (rejected null hypotheses). BH formally defined FDR as $\mathbf{E}\left[V/(R \vee 1)\right]$ and proposed a linear step-up procedure (the BH pro-

Table 2.1: An outcome of a multiple-testing procedure.

	Accept null hypothesis	Reject null hypothesis	Total
Null true	$m_0 - V$	V	m_0
Alternative true	$m_1 - S$	S	m_1
	$m - R$	R	m

cedure) to determine a rejection region that can guarantee the FDR to be below a certain threshold α. With $p_{(0)} \equiv 0$, the linear step-up procedure sets the threshold for significance c at $p_{(k)}$, where $k = \max\{i : p_{(i)} \leq i\alpha/m\}$.

Rather than searching for a p-value threshold that can guarantee FDR control at a specified level α, Storey (2002) proposed estimating the FDR for a fixed rejection region and provided a family of conservative point estimators. Using the tools of empirical processes, Storey et al. (2004) showed that the estimation approach can be used to control FDR when m is fixed and finite and is more powerful than the original step-up procedure of BH. The power gain behind Storey's estimation approach lies in its ability to adapt to the proportion of the true null hypotheses, which we define as $\pi_0 = m_0/m$. BH showed that their procedure controls FDR at level $\pi_0\alpha$ under an independence condition. In a later work, Benjamini and Yekutieli (2001) showed that the BH procedure controls FDR under a special positive dependence condition, again, at level $\pi_0\alpha$. Thus, the BH procedure is conservative when $\pi_0 < 1$, and it is easy to see that if the BH procedure is used at level α/π_0 then the FDR will be controlled at level α under various conditions. This suggests the use of an adaptive procedure that depends on an estimate of π_0.

Definition 1. *(Adaptive linear step-up procedure)*

Step 1. Compute $\hat{\pi}_0$.

Step 2. If $\hat{\pi}_0 = 0$ *reject all hypotheses; otherwise, test the hypotheses using the BH*

linear step-up procedure at level $\alpha/\hat{\pi}_0$.

Notice that the original BH procedure is a special case of this procedure in which $\hat{\pi}_0$ is set to 1 all the time. The estimation of π_0, or equivalently m_0, can also be used in adaptive FWER control (Hochberg and Benjamini, 1990) and is useful in many other applications. For more discussion of the importance of π_0 estimation, readers can refer to Section 2.2 of Benjamini et al. (2006).

The π_0 and FDR estimators of Storey (2002) and Storey et al. (2004) can be described as follows. Using the notation in Table 2.1, we define, for $t \in [0, 1]$, the following empirical processes:

$$\begin{aligned} V(t) &= \#\{\text{null } p_i : p_i \leq t\}, \\ S(t) &= \#\{\text{alternative } p_i : p_i \leq t\}, \\ R(t) &= V(t) + S(t). \end{aligned}$$

Also define FDR(t), the FDR when rejecting all null hypotheses with p-values $\leq$ t as

$$\text{FDR}(t) = \mathbf{E}\left[\frac{V(t)}{R(t) \vee 1}\right].$$

Storey (2002) proposed

$$\hat{\pi}_0(\lambda) = \frac{\#\{p_i > \lambda\}}{(1-\lambda)m} = \frac{m - R(\lambda)}{(1-\lambda)m}$$

as an estimator for π_0, where λ is a tuning parameter in $[0, 1)$ to be specified. The numerator of $\hat{\pi}_0(\lambda)$ is the number of p-values larger than λ, and the denominator is the expected number of p-values larger than λ when all hypotheses are true nulls. It is easy to show that $\hat{\pi}_0(\lambda)$ is a conservative estimator for π_0. The corresponding plug-in estimator for FDR(t) is defined as

$$\widehat{\text{FDR}}_\lambda(t) = \frac{m\hat{\pi}_0(\lambda)t}{R(t) \vee 1},$$

which was proved to be a conservative estimator for FDR(t) under various independence conditions (Storey 2002 and Storey et al. 2004). To use the $\hat{\pi}_0(\lambda)$ in the adaptive linear step-up procedure, it is a good practice to bound it away from zero. Storey et al. (2004) proposed a slightly altered estimator of π_0,

$$\hat{\pi}_0^*(\lambda) = \frac{m - R(\lambda) + 1}{(1-\lambda)m},$$

and also limited the significance threshold to be $\leq \lambda$ to obtain

$$\widehat{\mathrm{FDR}}_\lambda^*(t) = \begin{cases} \frac{m\hat{\pi}_0^*(\lambda)t}{R(t)\vee 1} & t \leq \lambda \\ 1 & t > \lambda \end{cases}$$

as an estimate of FDR(t). For any function F defined on $[0,1]$, define the step-up thresholding function as

$$t_\alpha[F] = \sup\{0 \leq t \leq 1 : F(t) \leq \alpha\}.$$

Storey et al. (2004) showed that thresholding at $t_\alpha\left[\widehat{\mathrm{FDR}}_\lambda^*\right]$ can control FDR at level α. We will refer to this procedure as Storey's λ procedure. Storey's λ procedure is the first adaptive procedure theoretically proven to strongly control the FDR, i.e., control FDR for any $\pi_0 \in [0,1]$. Storey et al. (2004) also showed that the procedure based on $t_\alpha\left[\widehat{\mathrm{FDR}}_\lambda\right]$ is equivalent to the BH procedure with α replaced by $\alpha/\hat{\pi}_0(\lambda)$. Thus, the FDR control approach of BH and the FDR estimation approach are closely related.

The above results hold for any fixed $\lambda \in [0,1)$, and thus, Storey (2002) and Storey et al. (2004) have proposed a family of conservative estimators of π_0 and FDR(t) and FDR control procedures indexed by the tuning parameter λ. Small values of λ typically yield high bias and low variance estimators, while large values of λ produce estimators with lower bias but higher variance. Arbitrary fixed λ values have been used; for example, a popular choice is to let $\lambda = 1/2$. Other choices of fixed λ can be

functions of α, the target FDR level. We will refer to the class of adaptive procedures that use predetermined tuning parameters as the *fixed adaptive* procedures. That is, the fixed adaptive procedures specify their tuning parameters without the use of data.

A different strategy is to use the data to select the tuning parameter value. If a histogram of p-values is examined, a decreasing trend in the density is often observed. Intuitively, it makes sense to select a value of λ so that the observed distribution of p-values greater than λ is roughly uniform on the interval $(\lambda, 1)$. Thus, it is natural to use the data to select λ, and many data-dependent methods have been proposed. For example, Storey (2002) and Storey et al. (2004) recommended using a bootstrap method to select λ from a finite candidate set. Other data dependent methods for choosing λ include Benjamini and Hochberg (2000), Storey and Tibshirani (2003) and Nettleton et al. (2006) among many others. We refer to this class of procedures that use data to dynamically choose tuning parameters as the *dynamic adaptive* procedures. However, no theoretical result is available for the dynamic adaptive procedures, and the choice of λ remains non-trivial.

Many new adaptive procedures have been proposed and theoretically shown to control FDR since the work of Storey et al. (2004). For some recent examples, see Benjamini et al. (2006), Blanchard and Roquain (2009) and Gavrilov et al. (2009). Through simulation, these authors all found that some versions of Storey's λ procedure are still the most powerful procedures within the class of procedures that have been theoretically proven to control FDR. Furthermore, through simulation, Blanchard and Roquain (2009) also found that a certain Storey's λ procedure ($\lambda = \alpha$) performed the best overall among proven FDR control procedures under independence and dependence conditions.

However, upon close inspection, we found the finite sample proof of conservative

point estimation in Storey et al. (2004) to be technically flawed (for details see the beginning of the Appendix). Thus, in Section 2.2 we give a correct proof in Theorem 1 and 2. Though all of the aforementioned FDR control procedures have been proven to hold the FDR level individually, we show that some of them can be formulated as adaptive linear step-up procedures with π_0 estimators in the form of – but more conservative than – $\hat{\pi}_0^*(\lambda)$. We show in Corollary 2 of Section 2.2.1 that these more conservative procedures control the FDR under more relaxed conditions than the conditions under which they were previously proven.

More importantly, we consider a class of procedures that can be thought of as dynamically selecting the λ for their plug-in estimators $\hat{\pi}_0^*(\lambda)$. Examples include Benjamini and Hochberg (2000), Storey et al. (2004), Nettleton et al. (2006) and Gavrilov et al. (2009). In Theorem 5 of Section 2.2.2, we are able to show for the first time that some of these procedures or their slight modifications lead to conservative point estimation of $\mathrm{FDR}(t)$. During the process, we also shed light on what constitutes a good procedure for selecting λ.

In Section 2.3, we also extend the asymptotic results of Storey et al. (2004) to a larger class of dynamic adaptive procedures in Theorem 6. The paper concludes with a discussion in Section 2.4. All technical proofs are in the Appendix.

2.2 Finite Sample Results

Most of the multiple testing procedures assume all the test statistics are independent; we call this condition the *total independence condition*. Storey et al. (2004) only assumes the test statistics corresponding to the true null hypotheses are independent; more specifically, true null statistics are independent among themselves and with alternative statistics. We call this condition the *null independence condition*. The

procedures of Benjamini and Hochberg (1995) and Storey et al. (2004) are the only two procedures that have been proven to control FDR under the null independence condition. In general, we always assume the null independence condition and that the p-values from tests of true null hypotheses each follows the Uniform$(0, 1)$ distribution.

Our strategy for proving that $\widehat{\text{FDR}}_\lambda(t)$ is a conservative point estimator of $\text{FDR}(t)$ is a two step approach. We first show that an oracle FDR estimator is a conservative point estimator of $\text{FDR}(t)$, then we show $\widehat{\text{FDR}}_\lambda(t)$ is more conservative than the oracle FDR estimator. Under the unrealistic condition that π_0 is known, we define the oracle FDR estimator as

$$\widehat{\text{FDR}}_{OR}(t) = \frac{m\pi_0 t}{R(t) \vee 1}.$$

Theorem 1. *Suppose that the null independence condition holds and that each p-value from a test of a true null hypothesis is uniformly distributed on (0,1), then*

$$\mathbf{E}[\widehat{\text{FDR}}_{OR}(t)] \geqslant \text{FDR}(t).$$

Theorem 2. *Under the conditions of Theorem 1,*

$$\mathbf{E}[\widehat{\text{FDR}}_\lambda(t)] \geqslant \mathbf{E}[\widehat{\text{FDR}}_{OR}(t)]$$

for all $t \in [0, 1]$ *and a fixed* $\lambda \in [0, 1)$.

Because the oracle FDR estimator is a conservative estimator for $\text{FDR}(t)$ by Theorem 1, a sufficient condition for an estimator to be a conservative estimator of $\text{FDR}(t)$ is that it be more conservative than the oracle FDR estimator. By Theorem 2, $\widehat{\text{FDR}}_\lambda(t)$ is one such estimator. Thus, we have the following corollary.

Corollary 1. *For any fixed* $\lambda \in [0, 1)$, $\widehat{\text{FDR}}_\lambda(t)$ *is a conservative estimator of* $\text{FDR}(t)$ *under the condition of Theorem 1.*

It is trivial to show that the augmented estimator $\widehat{\text{FDR}}^*_\lambda(t)$ is also a conservative estimator of FDR(t).

Now we restate the theorem that guarantees FDR control for Storey's λ procedure.

Theorem 3. *Under the conditions of Theorem 1,*

$$\text{FDR}\left(t_\alpha\left[\widehat{\text{FDR}}^*_\lambda(t)\right]\right) \leq \alpha$$

Proof. See the proof of Theorem 3 in Storey et al. (2004).

With the exception of Storey's λ procedure, all the adaptive procedures that have been proven to control FDR have been proved only under the total independence condition. However, it is straightforward to prove FDR control for many adaptive procedures under the weaker null independence condition through the use of the following corollary.

Corollary 2. *Suppose that an adaptive FDR estimator is*

$$\widehat{\text{FDR}}'_\lambda(t) = \begin{cases} \frac{m\hat{\pi}'_0 t}{R(t)\vee 1} & t \leq \lambda \\ 1 & t > \lambda \end{cases}$$

with $\hat{\pi}'_0 \geq \hat{\pi}^*_0(\lambda)$ *for some* $\lambda \in [0,1)$. *Then*

$$\text{FDR}\left(t_\alpha\left[\widehat{\text{FDR}}'_\lambda\right]\right) \leq \alpha$$

under the conditions of Theorem 1.

A heuristic proof is as follows. It is reasonable to assume FDR(t) is non-decreasing in t. Storey (2003) showed that under a mixture model, $\text{FDR}(t) = \frac{\pi_0 t}{G(t)}$, where $G(t)$ is the marginal distribution for p-values. It is straightforward to show that FDR(t) is increasing in t if G is a concave function. Then because $\hat{\pi}'_0 \geq \hat{\pi}^*_0(\lambda)$, $\widehat{\text{FDR}}'_\lambda(t) \geq \widehat{\text{FDR}}^*_\lambda(t)$ for any $t \in [0,1]$, and $t_\alpha\left[\widehat{\text{FDR}}'_\lambda\right] \leq t_\alpha\left[\widehat{\text{FDR}}^*_\lambda\right]$. Thus,

$$\text{FDR}\left(t_\alpha\left[\widehat{\text{FDR}}'_\lambda\right]\right) \leq \text{FDR}\left(t_\alpha\left[\widehat{\text{FDR}}^*_\lambda\right]\right) \leq \alpha.$$

A more technical proof which does not require that $\mathrm{FDR}(t)$ is non-decreasing is presented in the Appendix.

For the completeness of results and to illustrate the connection between the FDR control approach and the FDR estimation approach, we restate the following lemma, which appeared as Lemma 2 in Storey et al. (2004).

Lemma 1. *In general, the p-value step-up method $t_\alpha(\widehat{\mathrm{FDR}}_\lambda)$ with plug-in estimator $\hat{\pi}_0(\lambda)$ is equivalent to the BH procedure with α replaced by $\alpha/\hat{\pi}_0(\lambda)$.*

Storey et al. (2004) remarked that $t_\alpha(\widehat{\mathrm{FDR}}^*_\lambda)$ is essentially equivalent to $t_\alpha(\widehat{\mathrm{FDR}}_\lambda)$ in practice. Thus, Storey's λ procedure is essentially an adaptive linear step-up procedure. Surprisingly, many of the procedures that have been proved to control FDR can be cast in the form of Storey's λ procedure, and some can be shown to be more conservative than Storey's λ procedure, for some $\lambda \in [0, 1)$. Thus, the FDR control properties of such procedures can be guaranteed by our results as we demonstrate in the next subsection.

2.2.1 More Conservative Estimators

Many so-called two-stage procedures perform rejections according to some algorithms at a first stage, then the results of the first stage are used to obtain estimates of π_0 to plug in the final linear step-up procedure. These two-stage procedures can be formulated as variations of Storey's λ procedure, and their FDR controlling properties can be guaranteed by our results under conditions weaker than those previously considered.

The first example is proposed by Benjamini et al. (2006) as a two-stage procedure. Blanchard and Roquain (2009) pointed out that the procedure is equivalent to an

adaptive linear step-up procedure with π_0 estimator defined as

$$\hat{\pi}_0^{BKY}(\lambda) = \frac{m - R_{BH}(\lambda) + 1}{(1-\lambda)m},$$

where $R_{BH}(\lambda)$ is the number of rejections using the standard BH procedure at level $\lambda = \alpha/(1+\alpha)$. This is a modified version of Benjamini et al. (2006) with the numerator increased by one, but the difference is minor. $R_{BH}(\lambda) \leq R(\lambda)$ because the BH procedure finds $k = \max\{i : p_{(i)} \leq i\lambda/m\}$ and uses $p_{(k)}$ (which is guaranteed to be $\leq \lambda$ by the definition of k) as the significance threshold. Thus, $\hat{\pi}_0^{BKY}(\lambda) \geq \hat{\pi}_0^*(\lambda)$.

The second example estimator is proposed by Blanchard and Roquain (2009) to improve upon the Benjamini et al. (2006) procedure. Their procedure is equivalent to an adaptive linear step-up procedure with π_0 estimator defined as

$$\hat{\pi}_0^{BR}(\lambda) = \frac{m - R'(\lambda) + 1}{(1-\lambda)m},$$

where $R'(\lambda)$ is the number of rejections that result from using an adaptive one-stage step-up procedure at level $\lambda \in (0,1)$; the authors' recommended value for λ is α. In their first stage, instead of using the linear step-up threshold $i\alpha/m$, they reject all null hypotheses for which $p_i \leq p_{(k)}$, where $k = \max\{i : p_{(i)} \leq \min[(1-\lambda)\frac{\alpha i}{m-i+1}, \lambda]\}$. The authors showed that $\hat{\pi}_0^{BR}(\lambda)$ is less conservative than $\hat{\pi}_0^{BKY}(\lambda)$. It is straightforward to see that in their first stage, no p-value $> \lambda$ will be rejected. Thus, $R'(\lambda) \leq R(\lambda)$, and $\hat{\pi}_0^{BR}(\lambda)$ is more conservative than $\hat{\pi}_0^*(\lambda)$.

By corollary 2, the above two procedures can control the FDR at any specified level under the null independence condition. Note that the above two procedures have been proven previously to control FDR only under the total independence condition.

The results of this section show that with the same λ, Storey's λ procedure dominates the FDR control procedures based on $\hat{\pi}_0^{BKY}(\lambda)$ and $\hat{\pi}_0^{BR}(\lambda)$. Under the null independence condition, these procedures are inadmissible with respect to power for

any FDR level $\alpha \in (0,1)$. This partly explains why Storey's λ procedure has been found repeatedly through simulation to be the most powerful among the theoretically proven procedures. Yet, criteria for selecting an appropriate λ remain unsettled.

2.2.2 Dynamically Choosing λ

As the examples of Section 2.2.1 illustrate, many π_0 estimators can be cast in the form of $\hat{\pi}_0^*(\lambda)$ and be shown to be more conservative than $\hat{\pi}_0^*(\lambda)$ with the same λ. The challenge remains to find an appropriate λ, and further, an optimal λ. There are many proposed procedures that can be formulated as Storey's λ procedure with λ dynamically selected. We will first give the theoretical results, then go through a list of proposed procedures to show that either they or their slight modifications provide conservative point estimation for FDR(t) for all meaningful values of t.

We prove the theoretical results in this subsection using a martingale method. We observe the crucial fact that if the random variable λ is a stopping time with respect to a certain filtration, then the Optional Stopping Theorem can be used to prove the conservativeness of some dynamic adaptive procedures. Recall the empirical processes $V(t), S(t)$ and $R(t)$ as defined in Section 2.1. The FDR control proof in Storey et al. (2004) utilized a related martingale, where the "time" (i.e., the threshold t) ran backwards (in the direction of from 1 to 0). Here we view the time as running forward, and note that $\left\{\frac{m_0 - V(t)}{1-t} : t \in [0,1)\right\}$ is a martingale. This elementary fact is presented in the following lemma. The lemma is easy to verify, and the proof is omitted. Interested readers can refer to Karlin and Taylor (1975) for more discussion of martingales.

Lemma 2. *Under the conditions of Theorem 1,* $\left\{\frac{m_0 - V(s)}{1-s} : s \in [0,1)\right\}$ *is a martingale with time running forward with respect to the filtration* $\mathcal{F}_s = \sigma(1_{\{p_i \le u\}}, 0 \le u \le s,\ i =$

$1, \ldots, m)$, *i.e., for* $s \leq u < 1$, $\mathbf{E}\left[\frac{m_0 - V(u)}{1-u} | \mathcal{F}_s\right] = \frac{m_0 - V(s)}{1-s}$.

Lemma 3. *If* $\left\{\frac{m_0 - V(s)}{1-s} : s \in [0,1)\right\}$ *is a martingale and* λ *a stopping time with respect to* $\{\mathcal{F}_s : s \in [0,1)\}$ *and* λ *is bounded away from 1, i.e.,* $\lambda \leq \tau < 1$ *with probability 1 for some constant* $\tau \in (0,1)$, *then* $\mathbf{E}\left[\frac{m_0 - V(\lambda)}{1-\lambda}\right] = \frac{m_0 - V(0)}{1-0} = m_0$.

Lemma 3 is adapted from Theorem 8.1 (continuous time optional stopping theorem) of Karlin and Taylor (1975) on page 320. The stopping time λ must be bounded away from 1 to satisfy the uniform integrability condition of Karlin and Taylor's Theorem 8.1. For a stopping time λ with respect to $\{\mathcal{F}_s : s \in [0,1)\}$ and some constant $\tau \in (0,1)$, we can let $\lambda^* = \lambda \wedge \tau$. Then λ^* is also a stopping time with respect to $\{\mathcal{F}_s : s \in [0,1)\}$, and λ^* is bounded away from 1. If we choose τ to be close to 1, the λ^* is essentially λ and makes little or no difference in practice. The following lemma is a generalization of the Theorem 8.1, where the stopping time λ is guaranteed to be no less than a certain threshold κ.

Lemma 4. *If* $\left\{\frac{m_0 - V(s)}{1-s} : s \in [0,1)\right\}$ *is a martingale and* λ *a stopping time with respect to* $\{\mathcal{F}_s : s \in [0,1)\}$ *and* λ *is bounded away from 0 and 1, i.e.,* $0 < \kappa \leq \lambda \leq \tau < 1$ *with probability 1 for some constants* κ *and* $\tau \in (0,1)$, *then for any* $t \in [0, \kappa]$, $\mathbf{E}\left[\frac{m_0 - V(\lambda)}{1-\lambda} \middle| \mathcal{F}_t\right] = \frac{m_0 - V(t)}{1-t}$.

First, we can use Lemma 3 to show that if λ is an appropriate stopping time, $\hat{\pi}_0(\lambda)$ provides a conservative estimator of π_0.

Theorem 4. *Under the conditions of Theorem 1, if* λ *is a stopping time with respect to* $\{\mathcal{F}_s : s \in [0,1)\}$ *and is bounded away from 1, then*

$$\mathbf{E}[\hat{\pi}_0(\lambda)] \geqslant \pi_0.$$

Second, we use Lemma 4 to show that if λ is an appropriate stopping time, $\widehat{\mathrm{FDR}}_\lambda(t)$ provides a conservative estimator for $\mathrm{FDR}(t)$.

Theorem 5. *Under the conditions of Theorem 1, if λ is a stopping time with respect to $\{\mathcal{F}_s : s \in [0,1)\}$ and is bounded away from 0 and 1, i.e., $0 < \kappa \leq \lambda \leq \tau < 1$ with probability 1 for some constant κ and $\tau \in (0,1)$, then for all $t \in [0,\kappa]$,*

$$\mathbf{E}[\widehat{\mathrm{FDR}}_\lambda(t)] \geqslant \mathrm{FDR}(t).$$

Plainly speaking, the condition that λ is a stopping time with respect to $\{\mathcal{F}_s : s \in [0,1)\}$ means that, for any $s \in [0,1)$, it must be possible to determine whether or not $\lambda \leq s$, given all the p-values that are $\leq s$. In some sense, Theorem 5 is a generalization of Corollary 1 under the null independence condition. This is because a fixed $\lambda \in [\kappa, 1)$ is a stopping time that satisfies the conditions of Theorem 5. On the other hand, Corollary 1 applies to all $t \in (0,1)$ while Theorem 5 only proves the conservativeness of $\widehat{\mathrm{FDR}}_\lambda(t)$ for all $t \leq \kappa$. However, the condition that $0 < \kappa \leq \lambda$ for some constant $\kappa \in (0,1)$ is a sufficient condition not a necessary condition for $\widehat{\mathrm{FDR}}_\lambda(t)$ to be conservative. Theorem 4 shows that $\hat{\pi}_0(\lambda)$ is conservative, and thus, $\widehat{\mathrm{FDR}}_\lambda(t)$ with $\hat{\pi}_0(\lambda)$ plugged in is expected to be conservative for the whole range of t. Furthermore, carefully chosen κs can be used to modify any existing stopping time λ to a stopping time that satisfies the conditions of Theorem 5 and cover all meaningful values of t. For example, we can set $\kappa = \alpha$, the pre-specified FDR level, and let $\lambda^* = \lambda \vee \kappa$. Then λ^* is also a stopping time with respect to $\{\mathcal{F}_s : s \in [0,1)\}$, and $\widehat{\mathrm{FDR}}_\lambda(t)$ is conservative for all $t \in [0,\alpha]$. Benjamini and Hochberg (1995) motivated the FDR as a middle ground between no multiple testing adjustment (thresholding at α) and severe penalty of FWER (thresholding around α/m), so it is sensible to only look at those p-values that are $\leq \alpha$ for the purpose of FDR control at level α.

Thus, Theorem 5 can be used to give conservative point estimation of FDR(t) for all meaningful values of p-values.

Now we go through a list of procedures that dynamically search for λ. The first estimator is an interesting one and the earliest proposed dynamic adaptive procedure. Benjamini and Hochberg (2000) proposed a π_0 estimator as

$$\hat{\pi}_0^{BH}(k) = \frac{m-k+1}{(1-p_{(k)})m},$$

where the search for the final k starts from 2 and stops when the first time $\hat{\pi}_0^{BH}(k) > \hat{\pi}_0^{BH}(k-1)$. We call this procedure the *BH00* procedure and the final k chosen as k^*. The estimator has a nice graphical interpretation as described by Schweder and Spjøtvoll (1982). This estimator was only shown through simulation to provide FDR control in Benjamini and Hochberg (2000). Benjamini et al. (2006) proved that an adaptive linear step-up procedure with $\hat{\pi}_0^{BH}(k)$ as its π_0 estimator provides FDR control for any fixed $k \in \{1, \ldots, m\}$ and recommended choosing $k = \lfloor \frac{m}{2} \rfloor$ so that $p_{(k)}$ is approximately the median of the p-values. In general, $\hat{\pi}_0^{BH}(k) = \hat{\pi}_0^*(\lambda)$, where $\lambda = p_{(k)}$. Thus, $\hat{\pi}_0^{BH}(k)$ is of the form $\hat{\pi}_0^*(\lambda)$ where λ only takes on existing p-values and thus can be thought of as a quantile version of the $\hat{\pi}_0^*(\lambda)$. It is easy to verify that $p_{(k)}$ for k fixed and $p_{(k^*)}$ are stopping times, and thus, both π_0 estimators are conservative and their corresponding FDR(t) estimators will provide conservative point estimation for FDR(t). This is the first theoretical result for the original Benjamini and Hochberg (2000) version of dynamic adaptive procedure.

Mosig et al. (2001) proposed an iterative algorithm for estimating the proportion of true null hypotheses from a histogram of p-values. Nettleton et al. (2006) derived the limit of the algorithm when the histogram is evenly spaced and showed that the estimator can be characterized as a version of Storey's λ estimator with the λ dynamically chosen. The commonly used histogram-based method searches for a value

of λ on a finite equal-distance grid between 0 and 1, which is the set up used in the examples of Mosig et al. (2001) and the derivation of Nettleton et al. (2006). Suppose the interval $(0, 1]$ is partitioned into B equal-length bins numbered from 1 to B. Let $\lambda_i = \frac{i-1}{B}$ for $i \in \{1, \ldots, B+1\}$ so that the ith bin is $(\lambda_i, \lambda_{i+1}]$ for $i \in \{1, \ldots, B\}$. Define the bin count for bin i, n_i, as the number of p-values falling into the ith bin, i.e., $n_i = \#\{p_j : p_j \in (\lambda_i, \lambda_{i+1}]\}$. With the notation of $R(t)$ as defined in Section 2.1, denote the tail average of the bin count from ith bin to Bth bin as $\bar{n}_{i:B} = \frac{m - R(\lambda_i)}{B-i+1}$. Then Storey's λ estimator for π_0 at λ_i is $\hat{\pi}_0(\lambda_i) = B\bar{n}_{i:B}/m$. Nettleton et al. (2006) showed that the iterative estimator of Mosig et al. (2001) converges to $\hat{\pi}_0(\lambda_{\mathrm{I}})$ where $\mathrm{I} = \min\{i : n_i \leq \bar{n}_{i:B}\}$, i.e., with λ chosen as the left boundary of the first bin whose bin count is less than or equal to the tail average.

However, λ_{I} is not a stopping time because the random variable n_i is not measurable with respect to $\mathcal{F}_{\lambda_{\mathrm{i}}}$ for any $i \in \{1, \ldots, B\}$; in other words, n_i being the number of p-values in $(\lambda_i, \lambda_{i+1}]$ is not determinable from the p-values no larger than λ_i. However, it is easy to see that $\lambda_{\mathrm{I}^*} = \lambda_{\mathrm{I}+1}$ is a stopping time with respect to $\{\mathcal{F}_s : s \in [0, 1)\}$. Thus, to apply Theorem 4 and 5 to the histogram-based estimator, we should move one step further to the right and chose the λ_i that is the right boundary of the first bin whose bin count is less than or equal to the tail average. It also can be shown that λ_{I^*} is the first λ_i where $\hat{\pi}_0(\lambda_i) \geq \hat{\pi}_0(\lambda_{i-1})$. In the case when no such λ_i exists, we use the largest $\lambda_i < 1$, i.e., we choose $\lambda = 1 - 1/B$. We will call this modified version the *right-boundary* procedure. The procedure is similar in spirit to the Benjamini and Hochberg (2000) dynamic adaptive procedure based on $\hat{\pi}_0^{BH}(k)$. The major difference between the methods is the choice of the candidate λ set. However, we will show that the BH00 procedure and a histogram-based procedure are essentially the same when we allow histogram bins to adapt to the p-values in Section 2.4. As an interesting

comparison for now, we look at the case when p-values are discrete and reside on a uniformly distributed grid between 0 and 1. For example, p-values can be computed by permutation tests and rest on a equal-distance grid between 0 and 1. Suppose we set up the right-boundary procedure with bins that coincide with the grid. It can be shown the BH00 procedure will stop no later than the right-boundary procedure and could lead to over-conservativeness. Benjamini and Hochberg (2000) also remarked on the potential conservativeness of their method when p-values are discrete.

The condition that λ is a stopping time with respect to $\{\mathcal{F}_s : s \in [0,1)\}$ is a very generous condition. Any procedure that searches for a value of λ in the direction from 0 to 1 can be a candidate. For example, step-down FDR methods that sequentially assess the significance of p-value in the direction from 0 to 1 may naturally produce stopping times with respect to $\{\mathcal{F}_s : s \in [0,1)\}$. Consider the multi-stage step-down procedure first proposed in Benjamini et al. (2006). This procedure was subsequently studied by Gavrilov et al. (2009), who were able to prove its FDR control property. The proposed step-down procedure computes $\widehat{\mathrm{FDR}}_\lambda^{GBS}(t)$ with

$$\hat{\pi}_0^{GBS}(t) = \frac{m - R_{BH}(t) + 1}{(1-t)m},$$

i.e., instead of fixing a λ beforehand it sets $\lambda = t$ at each t. The procedure uses $p_{(k)}$ as a p-value threshold for significance, where

$$k = \max\{i : \widehat{\mathrm{FDR}}_\lambda^{GBS}(p_{(j)}) \le \alpha,\ j = 1, \ldots, i\}.$$

We refer to this procedure as the GBS procedure. The procedure can be viewed as trying to choose λ and a threshold for FDR control simultaneously. Note that $p_{(k)}$ is not a stopping time, but $\lambda^* \equiv p_{(k^*)}$ is, where $k^* = k+1 = \min\{i : \widehat{\mathrm{FDR}}_\lambda^{GBS}(p_{(i)}) > \alpha\}$. We call this procedure the *one-step GBS* procedure. Following the results of Theorem 4 and 5, $\hat{\pi}^*(\lambda^*)$ and $\widehat{\mathrm{FDR}}_{\lambda^*}^*(t)$ are conservative estimators of π_0 and $\mathrm{FDR}(t)$, respectively.

However, being forced to find the threshold for FDR control and λ simultaneously, the procedure may not allow the search for λ to fully adapt to the data. Thus, it is expected the procedure is more conservative than the BH00 procedure and the right-boundary procedure. Furthermore, a level α FDR controlling threshold c is usually much less than α when m is large. Finally, the GBS procedure is a step-down procedure, which is more conservative than a step-up procedure with the same thresholding values. So in practice, the GBS procedure is likely to be the most conservative procedure among the adaptive FDR control methods we have considered in this paper.

In general, a dynamic λ procedure will adapt to the data and provide less conservative π_0 and FDR(t) estimators than a fixed λ procedure with a small λ. For a fixed λ procedure with a large λ, the variances of corresponding π_0 and FDR(t) estimators are expected to be large. In contrast, dynamic adaptive procedures may be able to avoid both high bias and high variance. As an intuitive illustration, the right-boundary procedure is most likely to stop when the histogram starts to flatten out, a good indication that the most of p-values from the right hand side are coming from the true null uniform(0,1) distribution. Thus, a dynamic λ procedure is likely to be a good solution to the trade off of bias versus variance when choosing λ.

2.3 Asymptotic Results

Though all the above theories before Section 2.2.2 hold for arbitrary fixed $\lambda \in [0, 1)$, a λ must be chosen. Storey (2002) and Storey et al. (2004) recommended using a bootstrap method to select the λ. The procedure can be summarized as follows: For a candidate set, say $\Lambda = \{0, 0.05, 0.10, \ldots, 0.95\}$, which are the left boundaries of bins when $B = 20$, calculate $\hat{\pi}_0(\lambda)$ for each $\lambda \in \Lambda$. Then the mean squared error of each $\hat{\pi}_0(\lambda)$ is approximated as $\widehat{\mathrm{MSE}}(\lambda) = \frac{1}{J}\sum_{j=1}^{J}\left(\hat{\pi}_0^{*j}(\lambda) - \hat{\pi}_0^{*}\right)^2$, where $\hat{\pi}_0^{*} = \min_{\lambda\in\Lambda}[\hat{\pi}_0(\lambda)]$

and $\hat{\pi}_0^{*j}(\lambda)$ represents the jth of the J with-replacement bootstrap replications of $\hat{\pi}_0(\lambda)$. Then, λ is chosen as $\hat{\lambda} = \text{argmin}_{\lambda \in \Lambda} \left[\widehat{\text{MSE}}(\lambda)\right]$. The authors justified the use of the minimum of all the estimators, $\hat{\pi}_0^*$, as a plug-in estimator of π_0 from the fact that each of the $\hat{\pi}_0(\lambda)$s is positively biased.

There are other procedures that dynamically search for λ. For example, the Mosig et al. (2001) method can be viewed as searching for λ over a predefined grid between $[0, 1]$. On the other hand, the BH00 procedure searches over all the p_is. Storey et al. (2004) stated without proof that the asymptotic results should hold for the bootstrap selected $\hat{\lambda}$. Next we show that the asymptotic results hold for a much broader class of π_0 estimators.

Continue with the set-up of B bins on $(0, 1]$ with n_i being the number of p-values in the ith bin. Let $\hat{\pi}_{0i} = n_i B/m$, the π_0 estimator assuming the p-values in the ith bin all come from tests of true null hypotheses. For all $j \in \{1, \ldots, B\}, \min_i\{n_i\} \leq \bar{n}_{j:B}$, so that $\min_i\{\hat{\pi}_{0i}\} \leq \hat{\pi}_0(\lambda_j)$. Let $\hat{\pi}_0^* = \min_i\{\hat{\pi}_{0i}\}$, the most liberal π_0 estimator based on the B bins set-up. Denote the corresponding most liberal FDR(t) estimator, $\frac{m\hat{\pi}_0^* t}{R(t) \vee 1}$, by $\widehat{\text{FDR}}^*(t)$.

We will assume roughly the same set of assumptions as in Storey et al. (2004) (7)-(9) for our asymptotic results:

$$\lim_{m\to\infty} \frac{V(t)}{m_0} = G_0(t) \ a.s. \ \text{ and } \lim_{m\to\infty} \frac{S(t)}{m_1} = G_1(t) \ a.s. \ \text{ for each } t \in (0, 1], \tag{2.1}$$

where G_0 and G_1 are continuous functions.

$$G_0(t) = t \text{ for each } t \in (0, 1]. \tag{2.2}$$

$$\lim_{m\to\infty} m_0/m \equiv \pi_0 \text{ exists.} \tag{2.3}$$

The only difference between Storey et al. (2004) (7)-(9) and our set of assumptions is our condition (2.2) which is a special case of their condition that assumes $0 < G_0(t) \leq$

t for each $t \in (0, 1]$. Under these conditions, we can show that even the most liberal estimator $\widehat{\text{FDR}}^*(t)$ is simultaneously conservatively consistent for FDR(t), provided the bin number B is finite.

Theorem 6. *Suppose (2.1), (2.2) and (2.3) hold and that the number of bins, B, is finite. Then for each* $\delta > 0$,

$$\lim_{m\to\infty} \inf_{t\geq\delta} \left[\widehat{\text{FDR}}^*(t) - \text{FDR(t)}\right] \geq 0 \text{ and } \lim_{m\to\infty} \inf_{t\geq\delta} \left[\widehat{\text{FDR}}^*(t) - \frac{V(t)}{R(t) \vee 1}\right] \geq 0$$

Corollary 3. *Any* $\widehat{\text{FDR}}_\lambda(t)$ *estimator with* λ *chosen from a finite set of candidates in* $[0, 1)$ *is simultaneously conservatively consistent for* FDR*(t).*

With the above results, it is straightforward to prove that the equivalent of Storey et al. (2004) Theorem 4 also holds for $\widehat{\text{FDR}}_{\lambda^*}(t)$, where λ^* is arbitrarily chosen from Λ, a finite set. We will list the theorem here without proof. Define the pointwise limit of $\widehat{\text{FDR}}_\lambda(t)$ under the assumptions of equations (2.1), (2.2) and (2.3) as

$$\widehat{\text{FDR}}^\infty_\lambda(t) \equiv \frac{\left\{\frac{1-G_0(\lambda)}{1-\lambda}\pi_0 + \frac{1-G_1(\lambda)}{1-\lambda}\pi_1\right\} G_0(t)}{\pi_0 G_0(t) + \pi_1 G_1(t)}.$$

Then $t_\alpha[\widehat{\text{FDR}}_{\lambda^*}(t)]$ asymptotically provides strong control of FDR under the set of assumptions.

Theorem 7. *Suppose that (2.1), (2.2) and (2.3) hold. If there exists a* $t \in (0, 1]$ *such that* $\widehat{\text{FDR}}^\infty_{\lambda^*}(t) < \alpha$, *then*

$$\limsup_{m\to\infty} \text{FDR}\left(t_\alpha\left[\widehat{\text{FDR}}_{\lambda^*}\right]\right) \leq \alpha.$$

Thus, Storey's bootstrap method and the histogram-based method each provide strong control of FDR in the limit and are simultaneously conservatively consistent. Benjamini and Hochberg (2000) method searches over all the p_i's, which is equivalent

to setting $B = m + 1$, and cannot be shown here to have the asymptotic properties of this section.

These asymptotic results, however, need to be taken with caution because asymptotic conservative FDR control does not guarantee that a procedure will perform well for finite m. Black (2004) showed through simulation that the Storey (2002) bootstrap λ selection procedure produces a negative bias for π_0 estimation. Nettleton et al. (2006) also showed that the Storey (2002) bootstrap λ selection procedure exhibited the greatest degree of negative bias among all methods considered. A possible reason is the use of $\hat{\pi}_0^*$ in the place of π_0. Though each of the $\hat{\pi}_0(\lambda)$s, $\lambda \in \Lambda$, is positively biased, the minimum of a number of $\hat{\pi}_0(\lambda)$s is not guaranteed to remain positively biased. Thus, the selection of λ is not a trivial matter, in spite of the asymptotic results. The results in Section 2.2.2 that guarantee the conservative point estimation of FDR(t) for the case of finite m are more useful.

2.4 Discussion

Storey (2002) introduced a family of procedures, indexed by $\lambda \in [0, 1)$, that provide conservatively biased point estimations of FDR(t). Various versions of Storey's λ procedure have been shown through many studies to be the most powerful among the class of the procedures that have been proved to control FDR. But the choice of λ remained unsettled until now.

For any fixed λ, some upward bias in estimating π_0 is expected. The closer λ is to 1, the smaller the bias but the bigger the variance. A better approach is to dynamically search for λ, to let the data speak for themselves. We have identified a sufficient condition of a λ selection method to provide conservative point estimation for FDR(t). The methods with clear motivations for estimating π_0 are the preferable ones,

for example, the BH00 and the right-boundary procedures. Our sufficient condition is general and easy to check, though which method satisfying the condition is the best awaits further investigation.

On the other hand, smaller λ will yield a more positively biased π_0 estimator thus increasing the procedure's ability to tolerate dependence among statistics. Benjamini et al. (2006) and Gavrilov et al. (2009) showed through simulation that Storey's λ procedure with $\lambda = 0.5$ is the most powerful procedure under independence, and they went on to show that their procedures have better FDR control property under a positive dependence situation. However, their procedures are much more conservative adaptive methods compared to Storey's λ procedure with $\lambda = 0.5$. For example, the Benjamini et al. (2006) procedure is shown in Section 2.2.1 to be more conservative than Storey's λ procedure where λ is set to be $\alpha/(1+\alpha)$. Gavrilov et al. (2009)'s procedure is expected to be even more conservative than the Benjamini et al. (2006) procedure. It is an open question whether the ability to tolerate dependence is a property of a procedure or the consequence of its conservativeness. In another study, Blanchard and Roquain (2009) confirmed through simulation that a version of Storey's λ procedure ($\lambda = 1/2$) is the most powerful procedure under independence. Furthermore, they also showed that a certain Storey's λ procedure ($\lambda = \alpha$) is the overall best in term of balance of power and tolerance to positive dependence. Usually α is small, and the common choices are 0.05 or 0.10. Thus, setting $\lambda = \alpha$ gives a relatively conservative estimator for π_0, but this conservativeness pays off in the dependence situation.

Now we show that the BH00 procedure and a version of histogram-based procedure are essentially equivalent, and we propose a new procedure that is a generalization of the both procedures. If we allow flexible histogram bin setup, i.e., the lengths of

bins are not necessarily all equal, the right-boundary procedure can be modified to allow for variable-width bins, and the BH00 and the right-boundary procedures are special cases. Suppose the interval $(0, 1]$ is partitioned into B bins numbered from 1 to B. Consider bin boundaries $\lambda_0 = 0 < \lambda_1 < \ldots < \lambda_{B-1} < \lambda_B = 1$ such that bin i is $(\lambda_{i-1}, \lambda_i]$ for $i = 1, \ldots, B$. Define the π_0 estimator for the ith bin as

$$\hat{\pi}_0(i) = \hat{\pi}_0^*(\lambda_i) = \frac{m - R(\lambda_i) + 1}{(1 - \lambda_i)m}$$

for $i \in \{1, \ldots, B-1\}$. Then the generalized stopping time according to the BH00 procedure is λ_k where

$$k = \min\{2 \leq i \leq B-1 : \hat{\pi}_0(i) > \hat{\pi}_0(i-1)\} \tag{2.4}$$

if such an i exists and $B-1$ otherwise. Let n_i denote the number of p-values falling into the ith bin, i.e., $n_i = \#\{p_j : p_j \in (\lambda_{i-1}, \lambda_i]\}$. Now we extend the stopping time for the right-boundary procedure to the situation where bin lengths are arbitrary. The stopping time for the flexible histogram is λ_k where

$$k = \min\left\{1 \leq i \leq B-1 : \frac{n_i}{\lambda_i - \lambda_{i-1}} \leq \frac{m - R(\lambda_i) + 1}{1 - \lambda_i}\right\} \tag{2.5}$$

if such an i exists and $B-1$ otherwise. This stopping time has the interpretation of the first time the p-value density in ith bin is smaller or equal to the tail p-value density, i.e., the p-value density to the right of λ_i. When setting up the histogram bins, we would want to avoid the situation where there is no p-value in any particular bin, i.e., $n_i = 0$, for any $i \in \{1, \ldots, B-1\}$. This is because the search for λ will automatically stop at such a bin if it is ever reached, and the dynamic adaptive procedure degenerates into a fixed adaptive procedure. If all the p-values are continuous and no two are equal such that $0 < p_{(1)} < \ldots < p_{(m)} < 1$, we can let the flexible histogram adapt to the natural ordered p-values, i.e., let $B = m+1$ and $\lambda_i = p_{(i)}$ for $i = 1, \ldots, m$. Now $n_i = 1$

and $R(\lambda_i) = i$ for $i = 1, \ldots, m$. Then it is straightforward to show that the stopping condition in (2.4), i.e., $\hat{\pi}_0(i) > \hat{\pi}_0(i-1)$, is equivalent to

$$\frac{1}{p_{(i)} - p_{(i-1)}} < \frac{m - i + 1}{1 - p_{(i)}}, \tag{2.6}$$

which is the stopping condition in (2.5) with "$\leq$" replaced by "$<$". When p-values are continuous, the probability that the left hand side in (2.6) equals the right hand side is zero. Thus, the BH00 procedure and the histogram-based procedure are essentially the same.

We propose an adaptive histogram-based procedure that sets the histogram bins at distinct p-values. Let $0 < \mathfrak{p}_{(1)} < \ldots < \mathfrak{p}_{(B-1)} < 1$ be $B-1$ ordered distinct p-values of $p_1, \ldots, p_m$. Let $\lambda_i = \mathfrak{p}_{(i)}$ for $i = 1, \ldots, B-1$ such that $\lambda_0 = 0 < \lambda_1 < \ldots < \lambda_{B-1} < \lambda_B = 1$. We set up a histogram with B bins whose ith bin is $(\lambda_{i-1}, \lambda_i]$ for $i = 1, \ldots, B$. The stopping time is λ_k where

$$k = \min \left\{ 1 \leq i \leq B-1 : \frac{n_i}{\lambda_i - \lambda_{i-1}} < \frac{m - R(\lambda_i) + 1}{1 - \lambda_i} \right\}.$$

This adaptive histogram-based procedure essentially becomes the BH00 procedure when p-values are continuous, but avoids the unnecessary conservativeness of BH00 procedure when handling discrete p-values. Furthermore, this procedure naturally applies to the case in which p-values are discrete but non-uniformly distributed between 0 and 1, a situation which naturally arises in sequential permutation testing. By Theorems 4 and 5, $\hat{\pi}^*(\lambda_k)$ and $\widehat{\mathrm{FDR}}^*_{\lambda_k}(t)$ are conservative estimators of π_0 and $\mathrm{FDR}(t)$, respectively.

2.5 Appendix

Flaw of Theorem 1 in Storey et al. (2004):

At the end of proof, the authors claimed that

$$\begin{aligned} & \mathbf{E}\left[\frac{m_0 t - V(t)}{R(t) \vee 1}\right] \\ \geq \; & \mathbf{E}\left\{\mathbf{E}\left[\frac{\{m_0 t - V(t)\}/(1-t)}{S(t)+V(t)} \mathbf{1}_{\{V(t)\geq 1\}} \Big| S(t)\right]\right\} \\ \geq \; & 0 \end{aligned} \tag{2.7}$$

because the conditional expectation $\mathbf{E}\left[\frac{\{m_0 t - V(t)\}/(1-t)}{S(t)+V(t)} \mathbf{1}_{\{V(t)\geq 1\}} \Big| S(t)\right] \geq 0$ for any $S(t)$ by use of Jensen's inequality. However, for any fixed $S(t)$, $\frac{\{m_0 t - V(t)\}/(1-t)}{S(t)+V(t)} \mathbf{1}_{\{V(t)\geq 1\}}$ is not convex function of $V(t)$ and Jensen's inequality can not be used. Furthermore, as an easy counter example, consider when $m_0 t < 1$, $[m_0 t - V(t)]\mathbf{1}_{\{V(t)\geq 1\}} = 0$ when $V(t) = 0$ and < 0 when $V(t) \geq 1$. Thus, the conditional expectation is negative for all $S(t)$, and (2.7) has to be negative as well.

Proof of Theorem 1:

The true null p-values are independently uniform r.v., so $\mathbf{E}(V(t)) = m_0 t$. Then

$$\mathbf{E}[m_0 t - V(t)] = 0,$$

and that is

$$\sum_{j=0}^{m} (m_0 t - j) \Pr(V(t) = j) = 0.$$

Define $b(j) \equiv (m_0 t - j) \Pr(V(t) = j)$, then $\sum_{j=0}^{m} b(j) = 0$. Let $\omega = \lfloor m_0 t \rfloor$, the largest integer that is smaller or equal to $m_0 t$. We have $b(j) \geqslant 0$ for $j \leqslant \omega$, and $b(j) < 0$ for

$j > \omega$.

$$\begin{aligned}
& \mathbf{E}[\widehat{\mathrm{FDR}}_{OR}(t) - \mathrm{FDR}(t)] \\
= \; & \mathbf{E}\left[\frac{m_0 t - V(t)}{R(t) \vee 1}\right] \\
= \; & \mathbf{E}\left\{\mathbf{E}\left[\frac{m_0 t - V(t)}{[S(t)+V(t)] \vee 1}\Big| S(t)\right]\right\}.
\end{aligned}$$

Now, for any $S(t)$

$$\begin{aligned}
& \mathbf{E}\left[\frac{m_0 t - V(t)}{[S(t)+V(t)] \vee 1}\Big| S(t)\right] \\
= \; & \sum_{j=0}^{m} \frac{b(j)}{[S(t)+j] \vee 1} \\
= \; & \sum_{j=0}^{\omega} \frac{b(j)}{[S(t)+j] \vee 1} + \sum_{j=\omega+1}^{m} \frac{b(j)}{[S(t)+j] \vee 1} \\
\geqslant \; & \frac{1}{[S(t)+\omega] \vee 1} \sum_{j=0}^{\omega} b(j) + \frac{1}{S(t)+\omega+1} \sum_{j=\omega+1}^{m} b(j) \\
\geqslant \; & \frac{1}{S(t)+\omega+1} \sum_{j=0}^{m} b(j) \\
= \; & 0.
\end{aligned}$$

Thus, $\mathbf{E}[\widehat{\mathrm{FDR}}_{OR}(t)] - \mathrm{FDR}(t) \geq 0$.

Proof of Theorem 2:

$$\begin{aligned}
& \mathbf{E}[\widehat{\mathrm{FDR}}_{\lambda}(t)] - \mathbf{E}[\widehat{\mathrm{FDR}}_{OR}(t)] \\
= \; & \mathbf{E}\left[\frac{m\hat{\pi}_0(\lambda) t - m\pi_0 t}{R(t) \vee 1}\right] \\
= \; & \mathbf{E}\left[\frac{m\frac{m-R(\lambda)}{(1-\lambda)m} t - m_0 t}{R(t) \vee 1}\right] \\
\geq \; & \mathbf{E}\left[\frac{\frac{m_0 - V(\lambda)}{(1-\lambda)} t - m_0 t}{R(t) \vee 1}\right] \\
= \; & \mathbf{E}\left[\mathbf{E}\left(\frac{\frac{m_0 - V(\lambda)}{(1-\lambda)} t - m_0 t}{R(t) \vee 1}\Big| V(t), S(t)\right)\right]. \qquad (2.8)
\end{aligned}$$

When $t < \lambda$, $\mathbf{E}[m_0 - V(\lambda)|V(t)] = [m_0 - V(t)]\frac{1-\lambda}{1-t}$ because $m_0 - V(\lambda)|V(t) \sim$ Binomial$(m_0 - V(t), \frac{1-\lambda}{1-t})$, then the above equals to

$$\mathbf{E}\left[\frac{\frac{t}{1-t}[m_0 t - V(t)]}{R(t) \vee 1}\right].$$

Next, if $t \geq \lambda$ then $\mathbf{E}[V(\lambda)|V(t)] = V(t)\frac{\lambda}{t}$, and (2.8) $= \mathbf{E}\left[\frac{\frac{\lambda}{1-\lambda}[m_0 t - V(t)]}{R(t)\vee 1}\right]$. Then it is enough to show $\mathbf{E}\left[\frac{m_0 t - V(t)}{R(t)\vee 1}\right] \geq 0$, which follows from the proof of Theorem 1.

The following two Lemmas appeared in Storey et al. (2004) as Lemma 3 and 4. They are needed in the proof of Corollary 2.

Lemma 5. *If the p-values of the m_0 true null hypotheses are independent, then $V(t)/t$ for $0 \leq t < 1$ is a martingale with time running backwards with respect to the filtration $\mathcal{F}'_t = \sigma(1_{\{p_i \leq s\}}, t \leq s \leq 1,\ i = 1, \ldots, m)$, i.e., for $0 < s \leq t$, $\mathbf{E}[V(s)/s|\mathcal{F}'_t] = V(t)/t$.*

Lemma 6. *For $\lambda > 0$, $t_\alpha(\widehat{\mathrm{FDR}}_\lambda)$ is a stopping time with respect to $\mathcal{F}'^{\lambda}_t \equiv \mathcal{F}'_{t\wedge\lambda}$, with time running backwards.*

Proof of Corollary 2. This proof loosely follows the proof of Theorem 3 in Storey et al. (2004). Abbreviate $t_\alpha(\widehat{\mathrm{FDR}}'_\lambda)$ by t^λ_α. Notice that when t is running from 1 to 0, the process $mt/R(t)$ starts from 1 and has only upward jumps. If $R(t^\lambda_\alpha) = 0$, the FDR is zero by definition, and the theorem holds trivially. From now on, we assume $R(t^\lambda_\alpha) \geq 1$. If $\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha$, then $R(t^\lambda_\alpha) = t^\lambda_\alpha m\hat{\pi}'_0/\alpha$. And $V(t)/t$ stopped at t^λ_α is bounded by m/α. Thus,

$$\begin{aligned}
\mathbf{E}\left[\frac{V(t^\lambda_\alpha)}{R(t^\lambda_\alpha)}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right] &\leq \mathbf{E}\left[\frac{\alpha}{\hat{\pi}'_0 m}\frac{V(t^\lambda_\alpha)}{t^\lambda_\alpha}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right] \\
&= \mathbf{E}\left[\frac{\alpha}{\hat{\pi}'_0 m}\mathbf{E}\left[\frac{V(t^\lambda_\alpha)}{t^\lambda_\alpha}\Big|\mathcal{F}_\lambda\right]\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right] \\
&= \mathbf{E}\left[\frac{\alpha}{\hat{\pi}'_0 m}\frac{V(\lambda)}{\lambda}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right].
\end{aligned}$$

When $\widehat{\mathrm{FDR}}'_\lambda(\lambda) < \alpha$, then $t^\lambda_\alpha = \lambda$, $R(t^\lambda_\alpha) > t^\lambda_\alpha m\hat{\pi}'_0/\alpha$ and $V(t)/t$ stopped at t^λ_α is bounded by m/λ. It follows that

$$\mathbf{E}\left[\frac{V(t^\lambda_\alpha)}{R(t^\lambda_\alpha)}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) < \alpha\right] < \mathbf{E}\left[\frac{\alpha}{\hat{\pi}'_0 m}\frac{V(\lambda)}{\lambda}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) < \alpha\right].$$

Then,

$$\begin{aligned}
\mathrm{FDR}(t^\lambda_\alpha) &= \mathbf{E}\left[\frac{V(t^\lambda_\alpha)}{R(t^\lambda_\alpha)}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) < \alpha\right]\Pr\left(\widehat{\mathrm{FDR}}'_\lambda(\lambda) < \alpha\right) + \\
&\quad \mathbf{E}\left[\frac{V(t^\lambda_\alpha)}{R(t^\lambda_\alpha)}\Big|\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right]\Pr\left(\widehat{\mathrm{FDR}}'_\lambda(\lambda) \geq \alpha\right) \\
&\leq \mathbf{E}\left[\frac{\alpha}{m\hat{\pi}'_0}\frac{V(\lambda)}{\lambda}\right] \\
&\leq \mathbf{E}\left[\frac{\alpha}{m\hat{\pi}^*_0(\lambda)}\frac{V(\lambda)}{\lambda}\right] \leq \alpha.
\end{aligned}$$

The last step was shown in the proof of Storey et al. (2004) Theorem 3.

Proof of Theorem 4.

$$\begin{aligned}
\mathbf{E}[\hat{\pi}_0(\lambda)] &= \frac{1}{m}\mathbf{E}\left[\frac{m - R(\lambda)}{(1-\lambda)}\right] \\
&\geq \frac{1}{m}\mathbf{E}\left[\frac{m_0 - V(\lambda)}{(1-\lambda)}\right] \\
&= m_0/m = \pi_0
\end{aligned}$$

by Lemma 3.

Proof of Theorem 5.

By the result of Theorem 1, it is enough to show $\mathbf{E}[\widehat{\mathrm{FDR}}_\lambda(t)] \geq \mathbf{E}[\widehat{\mathrm{FDR}}_{OR}(t)]$. Also

by the derivations in the proof of Theorem 2,

$$\begin{aligned}
& \mathbf{E}[\widehat{\mathrm{FDR}}_\lambda(t)] - \mathbf{E}[\widehat{\mathrm{FDR}}_{OR}(t)] \\
\geq \; & \mathbf{E}\left[\frac{\frac{m_0 - V(\lambda)}{(1-\lambda)}t - m_0 t}{R(t) \vee 1}\right] \\
= \; & \mathbf{E}\left\{\mathbf{E}\left[\frac{\frac{m_0 - V(\lambda)}{(1-\lambda)}t - m_0 t}{R(t) \vee 1}\middle| \mathcal{F}_t\right]\right\} \\
= \; & \mathbf{E}\left[\frac{\frac{t}{1-t}[m_0 t - V(t)]}{R(t) \vee 1}\right] \\
\geq \; & 0.
\end{aligned}$$

The second to the last step follows from Lemma 4 through the use of the Optional Stopping Theorem. The last step is by the proof of Theorem 1.

Proof of Theorem 6. Let $G_{1i} = G_1(\frac{i}{B}) - G_1(\frac{i-1}{B}) \geq 0$, then by conditions (2.1), (2.2) and (2.3), $\forall i \in \{1, \ldots, B\}$,

$$\begin{aligned}
& \lim_{m\to\infty} \frac{n_i}{m} \xrightarrow{a.s.} \frac{\pi_0}{B} + (1-\pi_0)G_{1i} \\
\Rightarrow \; & \lim_{m\to\infty} \frac{n_i}{m} \overset{a.s.}{\geq} \frac{\pi_0}{B} \\
\Rightarrow \; & \lim_{m\to\infty} \min_{1\leq i\leq B} \frac{n_i}{m} \overset{a.s.}{\geq} \frac{\pi_0}{B} \\
\Rightarrow \; & \lim_{m\to\infty} \hat{\pi}_0^* \overset{a.s.}{\geq} \pi_0.
\end{aligned}$$

From equation (2.2), $G_0(t) = t$. Then for any $\delta > 0, \lim_{m\to\infty} \inf_{t\geq\delta}[\hat{\pi}_0^* t - \pi_0 G_0(t)] \overset{a.s.}{\geq} 0$. Thus,

$$\lim_{m\to\infty} \inf_{t\geq\delta} \left[\frac{m\hat{\pi}_0^* t}{R(t) \vee 1} - \frac{m\pi_0 G_0(t)}{R(t) \vee 1}\right] \overset{a.s.}{\geq} 0.$$

Storey et al. (2004) Theorem 6 proved that

$$\lim_{m\to\infty} \sup_{t\geq\delta} \left[\frac{V(t)}{R(t) \vee 1} - \frac{m\pi_0 G_0(t)}{R(t) \vee 1}\right] \overset{a.s.}{=} 0$$

and

$$\lim_{m\to\infty} \sup_{t\geq\delta} \left[\frac{V(t)}{R(t) \vee 1} - \mathrm{FDR}(t) \right] \overset{a.s.}{=} 0$$

under weaker set of assumptions. By combining the last three results, the proof is complete.

Bibliography

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B*, 57(1):289–300.

Benjamini, Y. and Hochberg, Y. (2000). On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25(1):60.

Benjamini, Y., Krieger, A., and Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, 93(3):491.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, 29(4):1165–1188.

Black, M. (2004). A note on the adaptive control of false discovery rates. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, pages 297–304.

Blanchard, G. and Roquain, E. (2009). Adaptive false discovery rate control under independence and dependence. *Journal of Machine Learning Research*, 10:2837–2871.

Gavrilov, Y., Benjamini, Y., and Sarkar, S. (2009). An adaptive step-down procedure with proven FDR control under independence. *Ann. Statist*, 37(2):619–629.

Hochberg, Y. and Benjamini, Y. (1990). More powerful procedures for multiple significance testing. *Statistics in medicine*, 9(7):811–818.

Karlin, S. and Taylor, H. (1975). *A first course in stochastic processes*. Academic Press New York.

Mosig, M., Lipkin, E., Khutoreskaya, G., Tchourzyna, E., Soller, M., and Friedmann, A. (2001). A whole genome scan for quantitative trait loci affecting milk protein percentage in Israeli-Holstein cattle, by means of selective milk DNA pooling in a daughter design, using an adjusted false discovery rate criterion. *Genetics*, 157(4):1683.

Nettleton, D., Hwang, J., Caldo, R., and Wise, R. (2006). Estimating the number of true null hypotheses from a histogram of p values. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(3):337–356.

Schweder, T. and Spjøtvoll, E. (1982). Plots of p-values to evaluate many tests simultaneously. *Biometrika*, 69(3):493.

Storey, J. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 64(3):479–498.

Storey, J. (2003). The positive false discovery rate: A Bayesian interpretation and the q-value. *Annals of Statistics*, 31(6):2013–2035.

Storey, J., Taylor, J., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187–205.

Storey, J. and Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9440.

CHAPTER 3. A Hidden Markov Model Approach to Testing Multiple Hypotheses on a Tree-Transformed Gene Ontology Graph

A paper accepted by the *Journal of the American Statistical Association*

Kun Liang and Dan Nettleton

Department of Statistics

Iowa State University, Ames, IA 50011

email: liangkun@iastate.edu

Abstract

Gene category testing problems involve testing hundreds of null hypotheses that correspond to nodes in a directed acyclic graph. The logical relationships among the nodes in the graph imply that only some configurations of true and false null hypotheses are possible and that a test for a given node should depend on data from neighboring nodes. We developed a method based on a hidden Markov model that takes the whole graph into account and provides coherent decisions in this structured multiple hypothesis testing problem. The method is illustrated by testing Gene Ontology terms for evidence of differential expression.

KEY WORDS: Bayesian data analysis; Differential expression; Directed acyclic graph;

False discovery rate; Gene set enrichment analysis; Microarray; Multiple testing; Simultaneous inference.

3.1 Introduction

The initial analysis of many microarray experiments includes testing a null hypothesis of equivalent expression across conditions for each of thousands of genes. A single statistic, often a p-value, is calculated for each gene. These statistics are then compared to a threshold for significance to identify a list of genes that are declared to be differentially expressed (DE). To interpret the results of such an analysis, researchers study the characteristics of the genes on the DE list as known from past research. Known characteristics of genes may include the molecular function of a gene, the biological process in which the gene operates, or the component of the cell in which the gene product is known to be found. Such information is formalized in the ontologies developed as part of the Gene Ontology (GO) project (Ashburner et al., 2000).

GO provides a controlled vocabulary of terms that describe characteristics of genes. Each gene on a microarray may be associated with zero or more GO terms depending on how well each gene has been characterized in past research. The subset of genes on a microarray associated with any one GO term is known as a gene set or a gene category. Because some GO terms have very specific meanings while others are quite general, many gene sets are proper subsets of other gene sets. For example, the set of genes associated with the GO term *primary metabolic process* is a subset of the genes associated with the GO term *metabolic process* because a primary metabolic process is a special case of a metabolic process. We can visualize GO as a directed acyclic graph (DAG). Each node in the graph represents a GO term. Each directed edge connects a parent node to a child node, where the genes associated with the child node are a

subset of the genes corresponding to its parent node.

Rather than conducting a test for each gene, this paper focuses on conducting a test for each gene set defined by a GO term. Suppose for treatment conditions $t = 1, \ldots, T$ and experimental units $u = 1, \ldots, n_t$; $\boldsymbol{X}_{tu}$ is a vector of expression measurements with one element for each of P genes on a microarray. For $i = 1, \ldots, N$; suppose $\boldsymbol{G}_i$ is a indicator matrix whose rows are a subset of the $P \times P$ identity matrix such that $\boldsymbol{G}_i\boldsymbol{X}_{tu}$ is the subvector of expression values for the genes in the ith gene set and the uth experimental unit of the tth treatment group. Furthermore, suppose that $\boldsymbol{G}_i\boldsymbol{X}_{tu} \sim F_t^{(i)}$ for all $i = 1, \ldots, N$; $t = 1, \ldots, T$; and $u = 1, \ldots, n_t$. We consider the problem of testing

$$H_0^{(i)} : F_1^{(i)} = \cdots = F_T^{(i)} \tag{3.1}$$

for $i = 1, \ldots, N$. The goal is to identify gene sets (or, equivalently, nodes in the GO DAG) for which $H_0^{(i)}$ is false (DE nodes). Such sets are of scientific interest because these are gene sets whose multivariate expression distribution changes with treatment.

This is a challenging multiple hypothesis testing problem for several reasons. First, note that the number of genes in a gene set ranges from a few genes to thousands of genes. Thus, the dimension of the multivariate distribution of interest varies from test to test. Second, the number of experimental units $(n_1 + \cdots + n_T)$ in a microarray experiment is often quite small relative to the dimension of many gene sets. Third, the correlation structure among genes is unknown and expected to be nontrivial. Fourth, many genes are in multiple gene sets so that the tests would be dependent even if genes were independent. Finally, because many gene sets are subsets of others, there are logical relationships among the N null hypotheses that should be accounted for in inference. In particular, if node i is a parent of node j, then the truth of $H_0^{(i)}$ implies the truth of $H_0^{(j)}$ because the expression vector for gene set j is a subvector of the

expression vector for gene set i. Furthermore, the truth of $H_0^{(i)}$ implies the truth of the null hypotheses for all descendants of node i in the GO graph. On the other hand, if $H_0^{(j)}$ is false, $H_0^{(i)}$ must also be false along with the null hypotheses for all ancestors of node i in the GO graph. Accounting for this structure implied by the GO graph is the chief focus of this paper.

In Section 2, we describe past research related to gene set testing. Our proposed approach is presented in Section 3 and evaluated through data-driven simulation in Section 4. The paper concludes with an example application and discussion in Section 5.

3.2 Past Research on Gene Set Testing

Initial methods for identifying gene sets of interest have focused on testing whether gene sets are "over-represented" or "enriched" among a list of individual genes declared to be differentially expressed (DE). Reference to many of these methods can be found in review articles by Khatri and Draghici (2005) and Allison, Cui, Page, and Sabripour (2006). Though popular among many scientists, these methods have been criticized on statistical grounds because they rely on the assumption of independence among genes (see, for example, Subramanian et al. 2005, Barry, Nobel, and Wright 2005, Allison et al. 2006, Goeman and Buhlmann 2007, Nettleton, Recknor, and Reecy 2008 among others). Variations on tests of enrichment that do not require identifying a list of DE genes have been proposed by Subramanian et al. (2005), Barry et al. (2005), Newton, Quintana, den Boon, Sengupta, and Ahlquist (2007), and Efron and Tibshirani (2007). While some of these methods recognize and attempt to account for correlation among genes in inference, they are all based on values of statistics computed separately for each gene.

A very different yet natural way to assess the relevance of a gene set would be to test for differences in the multivariate expression distribution across treatment conditions as in (3.1). The multivariate test is potentially more powerful than combining single gene tests as discussed and demonstrated by Nettleton et al. (2008). The multivariate gene set test methods currently available include Goeman's Global Test (Goeman, van de Geer, de Kort, and van Houwelingen 2004), Mansmann's Global Ancova (Mansmann and Meister, 2005), the Multiple Response Permutation Procedure (MRPP) developed by Mielke and Berry (2001) and utilized in gene set testing by Nettleton et al. (2008), Pathway Level Analysis of Gene Expression (Tomfohr, Lu, and Kepler 2005), and Domain-Enhanced Analysis (Liu, Hughes-Oliver, and Menius 2007) among others. As discussed in Section 3, the method that we propose can be used with any multivariate testing method that produces valid p-values.

There has been relatively little work on testing gene sets while accounting for the structure of the Gene Ontology (GO) graph. We are interested in methods that recognize that the truth of a parental null hypothesis implies the truth of the null hypotheses of its children. There are two general testing approaches that can produce inferences consistent with the logical constraints imposed by the GO graph. The first is the bottom-up approach which conducts tests at the bottom of the graph at the leaf nodes (the nodes without any children). First, all leaf nodes are tested using a procedure that controls familywise error rate (FWER) for the family of tests corresponding to only the leaf nodes. The FWER can be controlled by the Bonferroni method or Holm's (1979) method, for example. Next, the null hypothesis for any non-leaf node in the graph is rejected if and only if the node is an ancestor of one or more rejected leaf nodes. It is easy to verify that FWER for the entire graph is bounded above by α by noting that a type I error cannot be made anywhere in the

graph unless a type I error is made during leaf node testing.

A second strategy is known as the top-down approach. Testing starts at the root of the graph (a node with no parents). If the root node null is rejected, each child of the root is tested. Any subsequent node is tested as long as all of its parental null hypotheses have been rejected. If a null for a node is accepted, the nulls for all of its descendents (children, children of children, etc.) are automatically accepted. The significance thresholds for each test must be selected carefully in order to control FWER. Marcus, Eric, and Gabriel (1976) proposed a top-down closed testing procedure that can control FWER on a GO DAG $\mathcal{G}$. First, $\mathcal{G}$ must be expanded to a bigger graph $\tilde{\mathcal{G}}$ such that the nodes of $\tilde{\mathcal{G}}$ are closed under union and directed edges are included to connect any node corresponding of a union of nodes to the individual nodes in the union. For example, if A and B are two gene sets in $\tilde{\mathcal{G}}$, then $A \cup B$ is also in $\tilde{\mathcal{G}}$ by the closure of union, and there is a directed edge from $A \cup B$ to each of A and B. If each null hypothesis is tested at level α in $\tilde{\mathcal{G}}$ in the top-down fashion, then a FWER of α on the original graph $\mathcal{G}$ can be guaranteed. FWER control follows because the node that is the union of all true null nodes has to be tested and rejected (which happens with probability no larger than α) before any true null node in $\mathcal{G}$ can be rejected. The problem with the approach is that the requirement of closure under union generates an exponential number of new nodes from the original GO nodes and makes this method computationally infeasible.

There have been rapid developments in the top-down camp recently. Goeman and Mansmann (2008) proposed a focus level method based on Marcus' method to control FWER on a DAG. The method has the flavor of the bottom-up approach but is more of a variant of the top-down approach. To circumvent the computational burden of closure under union, the test starts from the so-called focus level nodes that are in the

middle of the graph instead of at the top. If any focus level node is rejected, then all its ancestor nodes are rejected. Then Marcus' method is applied to each sub-graph that starts with each focus node as root, equally dividing a target FWER level among sub-graphs. The author suggested that the focus level should be near to the GO terms that are of most interest to the researcher to enhance detection power for gene sets of interest. Nevertheless the choice of focus level nodes is somewhat arbitrary. Furthermore, the burden of closure under union is alleviated but not avoided. The level of any focus node is still subject to computational constraints that dictate that each union-completed sub-graph with a focus node as root be smaller than a certain size. This effectively forces the focus level nodes to be on low levels of the DAG.

Two other top-down methods apply specifically to trees rather than more generally to DAGs. Meinshausen (2008) proposed a FWER controlling method by penalizing each node by the inverse of its cardinality. More specifically, for FWER level α and a node A, the p-value is compared with $\frac{|A|}{m}\alpha$, where $|A|$ is the number of genes in A and m is the total number of genes in the tree. Though GO was mentioned as a candidate application, the method further requires that nodes sharing a parent be disjoint, which is not the case in the GO graph. Yekutieli (2008) attempted to determine the overall false discovery rate (FDR) that results when FDR is controlled at a specified level for the tests conducted at each level of the tree. He was able to derive an upper bound for overall FDR under the condition of independent statistics.

In the top-down approach, a node is tested only after all the null hypotheses of its ancestors have been rejected. If the null for any one ancestor fails to be rejected, neither a child node nor its descendents will be tested. This is true whether one attempts to control FWER or FDR using a top-down strategy. Decisions made for the nodes at upper levels of the DAG are more important in the sense that further

tests depend on them. On the other hand, in the bottom-up approach, the penalty for a Bonferroni-type correction could be severe if a graph fans out steadily and has a large number of leaf nodes. Furthermore, the results of a bottom-up analysis depend heavily on whether leaf nodes are DE. All the leaf-node descendants of a DE node could be equivalent expressed. Such a DE node cannot be detected with a bottom-up approach unless type I errors are made in the leaf analysis.

Generally speaking, the bigger the graph, the more bottom-up, top-down, or focus-level analyses depend on their starting nodes. These approaches are forced to reject or accept null hypotheses at a local area of the graph, and decisions made using local information may have bad consequences for other areas of the graph. In the next section, we try to avoid this "near-sightedness" by proposing a method that takes the whole graph into account while making logically coherent decisions on the DAG.

3.3 The Proposed Approach

Our proposed approach can be outlined as follows. First, we transform the GO DAG into a tree. Then, a single p-value for testing the null hypothesis in (3.1) is computed separately for each node in the GO tree. We then model the joint distribution of these p-values using a hidden Markov model (HMM). We treat the state of each null hypothesis as a random variable and propose a Markov model for the joint distribution of states. This Markov model places probability zero on any configuration of states that is not consistent with the logical constraints imposed by the structure of the GO tree. We then use a Markov chain Monte Carlo (MCMC) strategy to sample the joint distribution of state configurations for the tree conditional on the observed p-values. Each of the state configurations of nodes in the tree is translated back into a set of state configurations for the nodes in the original GO DAG. These sets of

state configurations for the nodes in the original GO DAG each necessarily satisfy the logical constraints imposed by the structure of the GO DAG and are used to make inferences about the state of each GO DAG node's null hypothesis.

The benefit of our tree transformation procedure, which is described in Section 3.1, is twofold. First, working with the tree rather than a DAG enables fast computation. Chapter 9 of Darwiche (2009) shows that the computational cost of exact inference on a graph increases exponentially with its treewidth, which is a graph-theoretic parameter that measures the resemblance of the graph to tree structure. In the GO DAG, the treewidth is easily greater than 20. This makes inference for a GO DAG millions of times more costly than inference for a tree of the same size. Second, the process we use to create a tree results in a set of nodes that have considerably less overlap in terms of shared genes than the nodes in the original GO DAG. The hidden Markov model that we describe in Section 3.2 assumes that the node p-values are conditionally independent given node states. Although this assumption is clearly false, reduced gene sharing among nodes makes the model a better approximation to reality.

The downside of transforming the GO DAG to a tree is a potential loss of power. It is straightforward to construct examples where the joint expression distribution for genes in a set A is identical across treatment conditions, the joint expression distribution for genes in a set B is identical across treatment conditions, but the joint expression distribution for genes in $A \cup B$ varies across treatment conditions. (Nettleton et al. (2008) presents such an example where A and B are each single-gene sets.) Thus, splitting a larger gene set into two smaller gene sets can make some changes in the joint expression distribution difficult or impossible to detect. However, our procedure still takes far better advantage of the multivariate nature of gene set expression than single-gene approaches, and as we will demonstrate in Section 4, our

procedure can be far more powerful than other existing multivariate approaches.

3.3.1 Converting a DAG to a Tree

We want to transform the GO DAG to a tree structure while preserving as much of the original DAG structure as possible. The process is illustrated in a small example shown in Figure 3.1. Fortunately the graph structure in GO indicates subset relationships. If we can remove all but one incoming edges for each node that has multiple parents, the graph becomes a tree. This is equivalent to removing the genes in the child node from all but one of its parent nodes. The action will detach the child from extra parents, but strictly the child node will remain a subset of the grandparent or grandparents. The subset relationships can be updated by drawing directed edges from the original grandparents to the child (see the edge from node 2 to 6 in Figure 3.1b). By repeating this process, some of the new directed edges will eventually connect an ancestor of an existing parent to the child node (see the edge from node 1 to 6 in Figure 3.1c). Such edges are redundant and can be eliminated. We continue the process until all but one parent are eliminated for each node in the GO DAG (see Figure 3.1d).

Any one of a node's multiple parents could be arbitrarily selected for retention. However, to remain close to the original DAG structure, we choose to retain the parent that minimizes the number of parental relationships that need to be broken. We refer to this number as the structural change cost. When two parents have the same structural change cost, the parent with the fewest genes is kept.

After the procedure, every node except the root node will have one and only one parent, and thus, the DAG will be transformed into a tree. Each of the original DAG nodes will be a union of one or more tree nodes. For example, DAG node 2 in

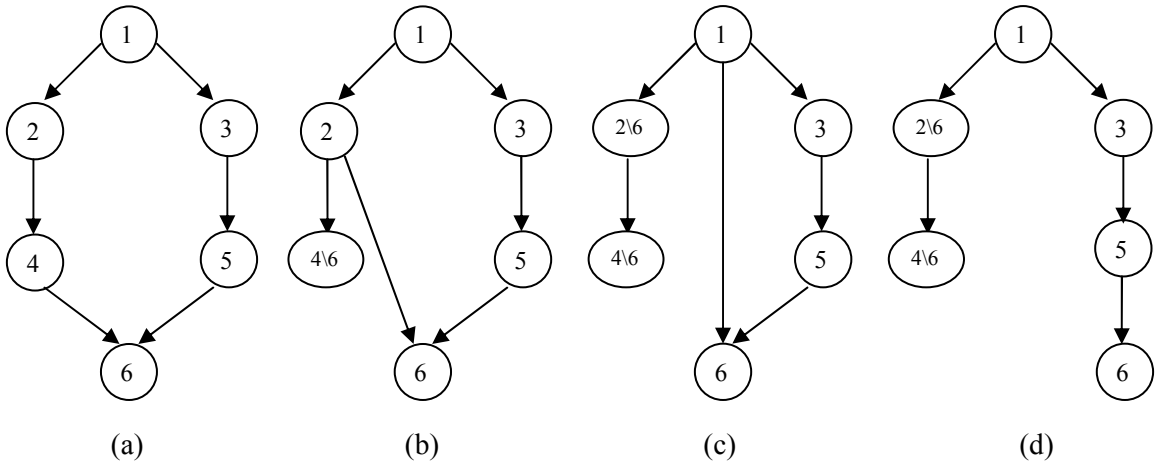


Figure 3.1: DAG to Tree: (a) Original DAG; (b) After remove genes in node 6 from node 4; (c) After remove genes in node 6 from node 2; (d) Tree after remove redundant edge from node 1 to node 6.

Figure 3.1a is a union of tree nodes 2\6 and 6 in Figure 3.1d. Although our MCMC algorithm (described in Section 3.3.3) samples tree nodes, we convert each draw of the complete tree into a draw of the original DAG. Specifically, any DAG node whose corresponding tree nodes are all in state 0 is set to state 0. All other DAG nodes are set to state 1 because if the null hypothesis associated with the genes in a tree node is false, the null must also be false for any DAG node which contains that set of genes. It is straightforward to show that this conversion process will always yield a logically consistent configuration of states for the original GO DAG.

3.3.2 A Hidden Markov Model for p-values on the GO Tree

For a tree node indexed by i, let $\mathcal{G}_i$ denote the set of indices of the genes in node i. Let $\mathcal{P}_i$ denote the index of the parent node of node i; i.e.,

$$\mathcal{P}_i = \{j : \mathcal{G}_i \subset \mathcal{G}_j \text{ and } \nexists\ k \text{ such that } \mathcal{G}_i \subset \mathcal{G}_k \subset \mathcal{G}_j\}.$$

Let p_i be the p-value associated with genes in the node i that is computed by testing (3.1) using any test that produces a valid p-value. Let S_i be the state of node i where $S_i = 0$ if the ith node is equivalent expressed and $S_i = 1$ if the ith node is DE. By the logical structure of the GO tree, a node must be in state 0 if its parental node is in state 0. On the other hand, we assume that a node whose parent is in state 1 can be in state 1 with some unknown transition probability ω. Hence, the transition portion of our hidden Markov model is given by

$$\Pr(S_i = 0 | S_{\mathcal{P}_i} = 0) = 1 \quad \text{and} \quad \Pr(S_i = 1 | S_{\mathcal{P}_i} = 1) = \omega.$$

Furthermore, we assume the root node of the tree (node with no parents) is in state 1 with probability ω. This establishes a simple model for the hidden node states. To model the observed p-values given the hidden states, we consider the model

$$p_i \sim \text{uniform}[0,1] \text{ if } S_i = 0 \text{ and } p_i \sim \text{beta}(\alpha, \beta) \text{ if } S_i = 1 \tag{3.2}$$

with p-values assumed to be conditionally independent of one another given the states. The parameters α and β are restricted to be in $(0, 1]$ and $(1, \infty)$, respectively, so that a strictly decreasing p-value density is guaranteed for p-values from DE nodes. This model for the conditional distribution of the p-values is borrowed from Allison et al. (2002), who proposed a finite mixture of beta distributions as a model for p-values from gene-specific tests for differential expression.

In essence, this is a hidden Markov process on the GO tree structure. It is hidden because the state of each node is unknown, and the Markov property follows as given its parent's states, a node's state is independent of the states of other ancestors.

To complete our model and facilitate estimation, we propose priors on our model parameters. The transition probability ω is assumed to follow the Jeffreys' prior of

beta$(0.5, 0.5)$. The parameters α and β are given diffuse priors of uniform$(0, 1]$ and uniform$(1, 2000)$, respectively.

We do not claim that our proposed model for the states is the true data-generating model. In particular, the assumption that p-values are conditionally independent given the states is clearly false because some genes belong to multiple nodes. This implies that the data for such genes will be involved in determining multiple p-values. The true model is undoubtedly more complex than we can afford to consider with datasets of practical size. However, despite the relative simplicity of our proposed working model, it leads to results that are quite useful in practice as we will demonstrate in subsequent sections of this paper.

3.3.3 Estimation

We are primarily interested in estimating the posterior probability of differential expression (PPDE) for each node in the original GO DAG. For any particular DAG node, this is the probability, given the observed p-values for all tree nodes, that one or more of the tree nodes that comprise the DAG node is DE. We utilize Metropolis-Hastings-in-Gibbs, a common MCMC strategy, to draw samples from the joint posterior distribution of tree nodes to estimate a PPDE for each DAG node.

We begin by examining the full conditional distributions. Given the data (p-values), all other parameters and states, ω depends only on the states and is the success probability of Bernoulli distributions for tree nodes whose parent nodes all are in state 1. With the conjugate beta prior, we can count the number of successes (n_s) and failures (n_f) to obtain beta$(n_s + 0.5,\ n_f + 0.5)$ as the full conditional distribution of ω.

Given the states, we know which p-values come from DE nodes. With uniform

priors, the full conditional distribution of α and β is proportional to the conditional likelihood of the p-values, $\prod_1^n [b(p_i|\alpha,\beta)]^{S_i}$, where $b(p_i|\alpha,\beta)$ is the value of beta density with parameter α and β at p_i. We sample α and β numerically using a Metropolis random-walk algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953).

Sampling state configurations from the full conditional distribution of states, given the data and the parameters (α, β and ω), is the most challenging aspect of our MCMC procedure. One possibility is to sample one state at a time conditional on all other states and parameters. This method is, in general, slow in mixing (Scott, 2002) and is especially so in our case due to the logical constraints forced on the GO tree. Chib (1996) showed that it is possible to sample the hidden states as a whole on a hidden Markov chain. We have devised a simple, direct, and computationally efficient method for sampling from the full conditional distribution of states in a binary state hidden Markov tree model. Our implementation, which is a special case of the so called dynamic programming algorithms that have been popular in computer science for many years, is derived below.

Assuming a tree structure, i.e., no node has more than one parent, let P_{i1} denote the event that the parent node of node i is in state 1, i.e.,

$$P_{i1} = \{S_{\mathcal{P}_i} = 1\}.$$

Let $\mathcal{C}_i$ denote the indices of the child nodes of node i, i.e.,

$$\mathcal{C}_i = \{j : \mathcal{G}_j \subset \mathcal{G}_i \text{ and } \nexists\ k \text{ such that } \mathcal{G}_j \subset \mathcal{G}_k \subset \mathcal{G}_i\}.$$

Let C_{i0} denote the event that all the child nodes of node i are in state 0, i.e.,

$$C_{i0} = \{S_j = 0\ \forall\ j \in \mathcal{C}_i\}.$$

Define the conditional probability of the ith node being DE as

$$c_i = \Pr(S_i = 1 | P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}),$$

where $\boldsymbol{p}$ is the vector of p-values and $\boldsymbol{\theta}$ is $\{\alpha, \beta, \omega\}$. As it turns out, the c_i probabilities are the key quantities for sampling the states, and we now show how to compute them recursively.

Let $\pi(\cdot|\cdot)$ denote a generic conditional density whose definition is to be inferred from its arguments. Let

$$A_{k0} = \Pr(S_i = k, C_{i0}|\boldsymbol{p}, P_{i1}, \boldsymbol{\theta})$$

for $k = 0, 1$ where the dependence on i is suppressed for notational simplicity.

By Bayes rule, we have

$$A_{k0} = \frac{\pi(\boldsymbol{p}|S_i = k, C_{i0}, P_{i1}, \boldsymbol{\theta}) \Pr(S_i = k, C_{i0}|P_{i1}, \boldsymbol{\theta})}{\pi(\boldsymbol{p}|P_{i1}, \boldsymbol{\theta})} \tag{3.3}$$

for $k = 0, 1$. But also, by the definition of c_i, we have

$$\begin{aligned} \left\{\prod_{j \in \mathcal{C}_i}(1 - c_j)\right\} c_i &= \Pr(C_{i0}|S_i = 1, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \\ &= \Pr(C_{i0}|S_i = 1, P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \\ &= \Pr(C_{i0}, S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) = A_{10} \end{aligned} \tag{3.4}$$

and

$$1 - c_i = \Pr(S_i = 0|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) = \Pr(S_i = 0, C_{i0}|\boldsymbol{p}, P_{i1}, \boldsymbol{\theta}) = A_{00}. \tag{3.5}$$

By equating A_{10}/A_{00} as given by (3.3) with A_{10}/A_{00} as given by (3.4) and (3.5) and then solving for c_i, we obtain the following expression for a node with at least one

child.

$$
\begin{aligned}
c_i &= \left\{1+\frac{\pi(\boldsymbol{p}|S_i=0,C_{i0},P_{i1},\boldsymbol{\theta})\Pr(S_i=0,C_{i0}|P_{i1},\boldsymbol{\theta})}{\pi(\boldsymbol{p}|S_i=1,C_{i0},P_{i1},\boldsymbol{\theta})\Pr(S_i=1,C_{i0}|P_{i1},\boldsymbol{\theta})}\prod_{j\in\mathcal{C}_i}(1-c_j)\right\}^{-1} \\
&= \left\{1+\frac{\pi(p_i|S_i=0,\boldsymbol{\theta})\Pr(S_i=0,C_{i0}|P_{i1},\boldsymbol{\theta})}{\pi(p_i|S_i=1,\boldsymbol{\theta})\Pr(S_i=1,C_{i0}|P_{i1},\boldsymbol{\theta})}\prod_{j\in\mathcal{C}_i}(1-c_j)\right\}^{-1} \\
&= \left\{1+\frac{\pi(p_i|S_i=0,\boldsymbol{\theta})(1-\omega)}{\pi(p_i|S_i=1,\boldsymbol{\theta})\omega(1-\omega)^{n_i}}\prod_{j\in\mathcal{C}_i}(1-c_j)\right\}^{-1} \text{(where } n_i=\text{cardinality of } \mathcal{C}_i) \\
&= \left\{1+\frac{1}{b(p_i|\alpha,\beta)\omega(1-\omega)^{n_i-1}}\prod_{j\in\mathcal{C}_i}(1-c_j)\right\}^{-1} \\
&= \frac{b(p_i|\alpha,\beta)\omega(1-\omega)^{n_i-1}}{b(p_i|\alpha,\beta)\omega(1-\omega)^{n_i-1}+\prod_{j\in\mathcal{C}_i}(1-c_j)}. \qquad (3.6)
\end{aligned}
$$

Now for a node i with no children,

$$
\begin{aligned}
c_i &= \Pr(S_i=1|P_{i1},\boldsymbol{p},\boldsymbol{\theta})=\Pr(S_i=1|P_{i1},p_i,\boldsymbol{\theta}) \\
&= \frac{\pi(p_i|S_i=1,\boldsymbol{\theta})\Pr(S_i=1|P_{i1},\boldsymbol{\theta})}{\pi(p_i|P_{i1},\boldsymbol{\theta})} \\
&= \frac{\pi(p_i|S_i=1,\boldsymbol{\theta})\Pr(S_i=1|P_{i1},\boldsymbol{\theta})}{\pi(p_i|S_i=1,\boldsymbol{\theta})\Pr(S_i=1|P_{i1},\boldsymbol{\theta})+\pi(p_i|S_i=0,\boldsymbol{\theta})\Pr(S_i=0|P_{i1},\boldsymbol{\theta})} \\
&= \frac{b(p_i|\alpha,\beta)\omega}{b(p_i|\alpha,\beta)\omega+1-\omega}. \qquad (3.7)
\end{aligned}
$$

Now using (3.6) and (3.7) together, we can compute c_i as a function of $\boldsymbol{p}$ and $\boldsymbol{\theta}$ for any node i in a bottom-up fashion. Given the values of c_i for all i, we can generate an observation from the conditional distribution of $\boldsymbol{S}$ given $\boldsymbol{p}$ and $\boldsymbol{\theta}$ by starting at the root of the tree and working down to the leaf nodes. Specifically, we begin by generating the state of the root node ($i=1$) from a Bernoulli distribution with success probability c_1. If the draw is 1, all its children become eligible for the drawing. This drawing process is then repeated for all eligible nodes, each with its own success probability c_i, until there is no eligible node left. All the nodes that do not participate in the drawing are set to state 0.

A proof that the state configurations generated by this conditional probability scheme are draws from the full conditional distribution of $\boldsymbol{S}$ given $\boldsymbol{p}$ and $\boldsymbol{\theta}$ is provided in the Appendix.

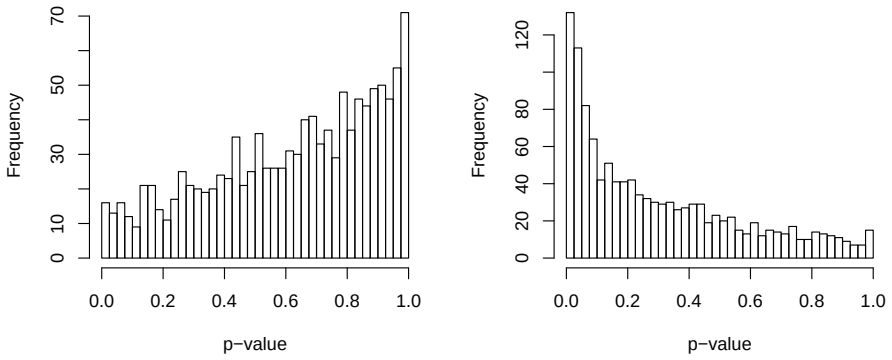
3.3.4 Extensions

Though the above model fits well in most of cases, we also considered a couple of extensions to make our model more realistic and robust. Let us first consider the transition portion of our Markov model for the states. In the initial model, we assumed the same transition probability for all transitions from a parent in state 1 to a child also in state 1. We realize that this is not a realistic assumption. For example, imagine that a DE parent node has 1000 genes while its child has 999 genes of these 1000. It is natural to expect the child to be DE with probability near 1. Indeed, the proportion of genes in a child node among those in its parent node contains information that hasn't been utilized. One simple mechanism for using this information would be to set the transition probability equal to the proportion $|\mathcal{G}_i|/|\mathcal{G}_{\mathcal{P}_i}|$. However, using only the proportion would automatically lead to small transition probabilities for child nodes that are small relative to their parents. Hence, we propose a transition probability that incorporates the proportion without punishing small child nodes. In particular, we assume

$$P(S_i = 1 | S_j = 1 \ \forall j \in \mathcal{P}_i) = \omega_i,$$

where $\omega_i = \max(\omega, |\mathcal{G}_i|/|\mathcal{G}_{\mathcal{P}_i}|)$. That is, for a child node whose genes make up a large proportion of its parent's genes, we use the proportion as the transition probability and ω otherwise. In the computation of conditional probabilities in (3.6) and (3.7), ω will be replaced by ω_i. With this modification to our model, the full conditional distribution of ω is no longer beta. Thus, we use the Metropolis-Hastings algorithm

when updating ω in our MCMC procedure. While the adjustment to our transition probability portion of the model is not necessary for achieving reasonable results in most cases, it does prevent overestimation of ω that can occur if many transitions from state 1 are nearly guaranteed by child nodes that are nearly identical to their parents.



(a) simulated dataset 11 (b) simulated dataset 16

Figure 3.2: Histograms of true null p-values from two datasets simulated in Section 3.4.

For a second variation on our modeling strategy, consider the distribution of p-values from true null tree nodes. Provided that we have a continuously distributed test statistic with a known null distribution, the distribution of a p-value from a test with a true null hypothesis should follow a uniform[0,1] distribution. Furthermore, our hidden Markov model implies that the p-values are independent given the states. Thus, if our model were correct, we would expect the collection of p-values with true null hypotheses to behave like an iid sample from a uniform distribution. However, in our case the nodes share genes so their p-values are not actually independent, even after conditioning on the states. In Section 3.4, we describe a data-based simulation

strategy that allows us to examine the joint distribution of null p-values under realistic correlation structures. Although marginally each null p-value is approximately distributed as uniform[0, 1], the joint distribution of null p-values will sometimes depart substantially from the product uniform distribution. Figure 3.2 includes the histograms of p-values of true null nodes for two simulated datasets. Notice that the null p-values of dataset 11 are skewed to the left while those from dateset 16 are skewed to the right.

If we insist on treating the null distribution as uniform, our method tends to overestimate the proportion of nodes that are null in simulated dataset 11 which leads to an overly conservative analysis. On the other hand, simulated dataset 16 will yield a liberal analysis because the excessive number of small null p-values will be mistaken as evidence for many DE nodes. Based on our observation of a large number of simulations, the distribution of p-values from equivalent expressed nodes usually has only one major peak due to positive correlations among nodes. Thus, we propose a mixture of a uniform and a unimodal beta distribution to approximate the distribution of p-values that come from the true null gene sets. The true null distribution of p-values in (3.2) changes to

$$p_i \sim \lambda + (1-\lambda)\text{beta}(\alpha_0, \beta_0) \quad \text{if } S_i = 0, \tag{3.8}$$

where α_0 and β_0 are each restricted to be bigger than 1 so that a unimodal p-value density is guaranteed. It is easy to see that a uniform model or a unimodal beta model are degenerated cases of (3.8). Bayes factor could be used to choose between the mixture model and the simpler uniform model or unimodal beta model. In practice, one can run the mixture model and simply look at the posterior diagnostics to tell which model provides a better fit. For the majority of the simulated cases that we examined, a simple uniform distribution was sufficient. However, for cases like

simulated dataset 16, the mixture model is needed to avoid a large number of false positive results. Note that our alteration of the uniform null p-value model is similar in spirit to the approach of Efron (2004) who recommends using data to estimate an "empirical" null distribution.

3.3.5 Rejection Region

After the MCMC chains converge, a posterior sample of size B can be obtained. Then the PPDE for DAG node i is estimated as $\text{PPDE}_i = \frac{1}{B}\sum_{k=1}^{B} S_i^{(k)}$, where $S_i^{(k)}$ is the kth posterior sample of the state of the ith DAG node (which is the maximum of the states of the tree nodes that comprise the DAG node). By definition, $1-\text{PPDE}_i = \Pr(S_i = 0|\boldsymbol{p})$, which is similar to the local index of significance defined by Sun and Cai (2009) in their work on testing HMM-dependent hypotheses. For any rejection index set R, a natural estimate for the FDR is

$$1 - \frac{1}{|R|}\sum_{i\in R}\text{PPDE}_i, \tag{3.9}$$

i.e., 1 − the average of the PPDE estimates for nodes in the rejection set. While the rejection set could in principle be any subset of nodes, we recommend selecting a subset of nodes with the highest estimated PPDE values. This guarantees the logical consistency of the rejection set and is in accordance with the strategy recommended by Sun and Cai (2009).

The number of nodes in the rejection set could be chosen to be the maximum number of nodes such that (3.9) is no larger than some user-specified FDR level. However, as noted by Goeman and Mansmann (2008), FDR may not be an appropriate quantity to control in a structured hypothesis testing problem. Furthermore, FDR by definition is the expected proportion of type I errors among rejected null hypotheses. Although this can be a useful error rate to examine when PPDE is unavailable, FDR

carries little information about how error prone each individual rejection is. It could happen that a list of rejections achieves a small estimated FDR by combining many nodes with PPDE near 1 together with a few low-PPDE nodes whose null hypotheses should not be rejected. Such a situation can arise in our case. Due to the logical constraints imposed by the GO graph structure, the higher-level ancestor nodes nearest the root have larger PPDE than the lower descendant nodes. Often the DE nodes at the highest levels have PPDE very close to 1, and this can give room for an FDR control method to admit some non-sensible low PPDE nodes from the lower levels of the DAG. Thus, rather than considering FDR, we suggest using a threshold on PPDE to choose the rejection set.

3.4 A Data-Based Simulation Study

We used a data-based simulation procedure proposed by Nettleton et al. (2008) to simulate a dataset that is as close to real data as possible. The B- and T-cell Acute Lymphocytic Leukemia dataset (Chiaretti et al., 2004) was used as a base to simulate data. The dataset is publicly available in the Bioconductor ALL package at www.bioconductor.org. The data consists of 12625-dimensional expression profiles from the Affymetrix HGU95aV2 GeneChip for each of 128 patients. Of the 128 patients, 95 have B-cell while 33 have T-cell acute lymphocytic leukemia. Using version 2.0.1 of the hug95av2 Bioconductor package, we were able to map 8192 of the Affymetrix probe sets (henceforth referred to as genes) to at least one Gene Ontology (GO) term from the biological process ontology. Note that we filtered out annotations that are inferred by electronic annotation instead of human curators because such annotations may be unreliable. This left 2353 unique GO terms for testing.

Liu et al. (2007) analyzed the same dataset to identify the most significant differentially expressed (DE) categories in the biological processes ontology for their Domain-Enhanced Analysis with Partial Least Squares method and the Fisher's exact test approach. We combined their result of the top ten categories for each method and got 14 unique categories. These 14 categories involve 845 of the 12625 genes in the Acute Lymphocytic Leukemia dataset. We will refer to this set of 845 genes as the *swap set*.

The following procedure was used to generate each of 20 simulated datasets. First n subjects were drawn randomly without replacement from T-cell patients and only the genes in the swap set were kept. $2n$ subjects were drawn randomly without replacement from B-cell patients. The first n of these subjects were left intact, and the swap sets of the second n subjects were replaced with the swap sets from the n T-cell subjects sampled in the first step. The n was chosen to be 9 in our simulations.

This simulation scheme allows us to simulate a dataset that mimics all the aspects of a real dataset. Not only does it preserve the marginal distributions of genes, but also it maintains the correlation structure among most genes. The only correlations the simulation scheme cannot maintain are the correlations between the swapped genes and others genes in the second half of the B-cell patients.

There are 1103 categories that don't share any gene with the swap set, and by construction their corresponding null hypotheses are true nulls. The other 1250 GO categories sharing some genes with the swap set are differentially expressed by construction. Using a similar simulation strategy, Nettleton et al. (2008) claimed that not all categories sharing some genes with the swap set were necessarily differentially expressed. However, note that genes in the swap set are sampled from two different finite populations of B-cell and T-cell patients. These finite populations have different

mean vectors, different gene-specific variances, different between gene correlations, etc. Thus, categories sharing genes with the swap set are indeed differentially expressed. Although technically DE by construction, many of these nodes contain only a few genes from the swap set or only genes with small effects. Thus, we expect low power to detect differential expression for many nodes.

The p-values were calculated for the tree nodes using the nonparametric method discussed in Mielke and Berry (2001) and Nettleton et al. (2008). This is essentially a subject-sampling permutation test which is free of distributional assumptions. More specifically, for any gene set, the treatment labels of subjects are permuted, and the sum of the within-group inter-subject Euclidean distances between gene set expression vectors is computed and compared with the sums computed for all other permutations. Then the p-value is the standardized rank of the original sum of within-group distances (scaled to be between 0 and 1). The total number of permutations which maintain two groups of size 9 is $\binom{18}{9}$, among which are pairs of symmetric permutations formed by flipping all treatment labels. Each permutation in a given symmetric pair will yield the same test statistic. Thus, it is necessary to consider only $\binom{18}{9}/2 = 24310$ of the permutations. Other multivariate testing methods mentioned in Section 3.1 could be used to compute p-values as well.

We compared our method with the bottom-up method described in Section 3.2 and the one-step min-p method proposed by Westfall and Young (1993) applied on the original GO DAG at FWER 0.05. For the min-p method, we compute p-values for each gene set for all 24310 permutations of the treatment labels as described in the previous paragraph. All the p-values are arranged in a 2353 by 24310 matrix. Then the minimum p-value across all gene sets for each permutation is obtained. These 24310 minimum p-values provide a reference distribution for the smallest p-value under the

null hypothesis of no treatment difference. We can declare significant any p-value that is no larger than the 0.05 quantile of the reference distribution. We then enforce logical consistency in accordance with the DAG by adjusting results using the bottom-up strategy so that all ancestors of significant nodes will also be declared significant. We also considered the potentially more powerful step-down min-p method but obtained results identical to the one-step approach in all 20 simulated datasets; for detail of the step-down version of the min-p method, see Westfall and Young (1993).

We considered a variety of other methods in our simulation study, but all other approaches were ultimately excluded. For example, a variant of the bottom-up method is to apply Holm's method to all the nodes and reject the ancestors of rejected nodes. This variant does not tend to work well when the number of nodes is large, as in the case of a GO DAG. Because the threshold for significance controlling FWER at 0.05 level is smaller than the smallest p-value, this would lead to no rejections for all the simulated samples. This variant can have better performance than the bottom-up method when the graph size is small, but it is useless in our situation. It is not computationally feasible to use the top-down approach because Marcus' method requires an exponential expansion of the already-large GO DAG (as discussed in Section 3.2). While it would be conceivable to try Goeman's focus level method, the performance would depend heavily on the choice of the focus level nodes that we have no basis for choosing. Because we transformed the GO DAG to a tree for computational reasons, the tree-based methods discussed in Section 3.2 seem viable. However, Meinshausen's method requires disjoint sets, and the nodes of our tree are not disjoint. While transforming the GO DAG into a disjoint tree seems feasible, it would result in a tree with the number of nodes close to the number of genes, and the graph structure of the GO DAG and the potential power gain from multivariate

testing would be largely lost. Yekutieli's estimate of FDR is not justified because the p-values of our tree nodes are not independent, and the dependence will be quite strong in many cases due to substantial sharing of genes among nodes. Furthermore, it is not clear how to calculate the FDR for the original GO DAG after controlling for a certain FDR level on the corresponding tree structure.

Table 3.1: Number of rejections and false positives across 20 simulated datasets for the proposed HMM method, the bottom-up method, and the min-p method.

	HMM		bottom-up		min-p	
Simulated Dataset	Number of Rejections	Number of False Positives	Number of Rejections	Number of False Positives	Number of Rejections	Number of False Positives
1	495	0	135	0	189	0
2	428	1	161	0	195	0
3	343	0	180	0	212	0
4	436	3	167	0	188	0
5	397	0	161	0	166	0
6	361	10	148	0	204	0
7	340	4	148	0	176	0
8	360	9	148	0	159	0
9	466	11	182	0	218	0
10	585	24	161	0	185	0
11	336	2	127	0	130	0
12	498	32	182	0	192	0
13	260	0	148	0	170	0
14	403	0	179	0	200	0
15	384	6	182	0	197	0
16	562	31	171	0	190	0
17	364	6	187	0	207	0
18	478	16	133	0	169	0
19	274	0	182	0	196	0
20	346	3	158	0	191	0

We chose the PPDE cutoff for our method to be 0.95. For the two FWER-controlling methods, we chose to control FWER at 0.05. We recognize that these two error control strategies are not directly comparable. However, methods for controlling error rates other than FWER are not available for the bottom-up and min-p approach. The results are shown in Table 3.1.

Both FWER-controlling methods exhibited excellent performance with regard to type I error control. No type I errors were made by either of the FWER-controlling methods across all 20 simulated datasets. The min-p approach was superior to the bottom-up approach with respect to power for all 20 simulated datasets. The HMM method exhibited far more power than either of the FWER-controlling methods, often identifying more than twice as many true positive results at the cost of very few additional false positives. Even for simulated dataset 12 where the HMM procedure arguably performed worst, we believe that many scientists would prefer the HMM results to those obtained using the procedures that control FWER. For this case, the HMM approach found more than 2.4 times the number of truly differentially expressed categories that were identified by the better of the competing approaches ($498 - 32 = 466$ versus 192). The cost for the additional 274 discoveries in this worst case was 32 type I errors. In all cases, our HMM method included all the discoveries that were made by the two FWER-controlling methods except one made by the min-p method in 6th dataset.

To further illustrate the advantage of our HMM method, we drew the receiver operating characteristic (ROC) curves in Figure 3.3 to compare the HMM method with the min-p method, the bottom-up method and a method based only on p-values. This latter method rejects the nodes in the order of their p-values, from the smallest to the largest, without using any structural information in the GO DAG. The bottom-up method is superior to the method based on p-values alone because it uses some part of the GO DAG structural information (leaves and their parents). The min-p method is superior to the bottom-up method because it is not confined to leaf nodes. The HMM method is superior to the min-p method because it further utilizes the GO DAG structural information by modeling the whole graph. Thus, the power advantage

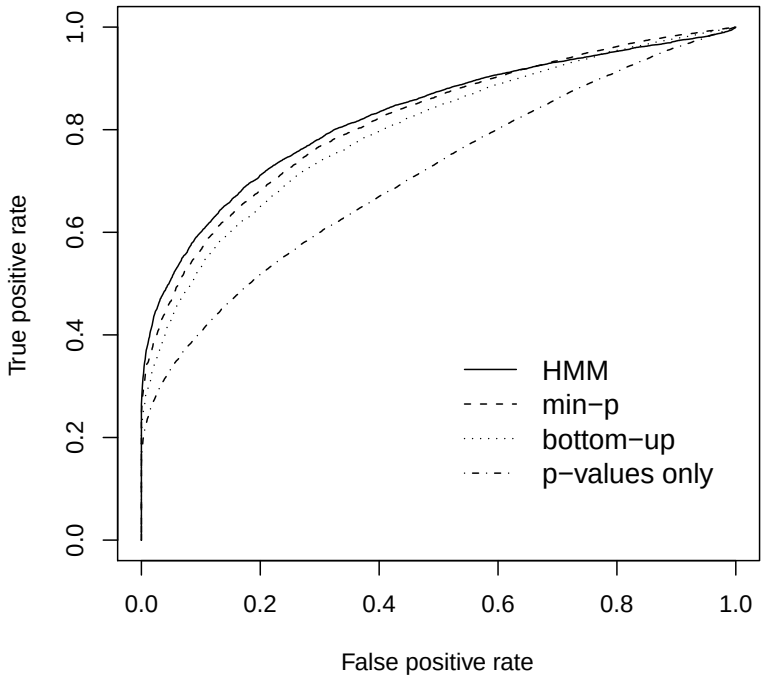


Figure 3.3: ROC curve for the HMM, min-p, bottom-up and p-values only methods.

exhibited in our Table 3.1 simulation result was not simply a consequence of differing error control criteria. Our HMM approach was better able to distinguish DE gene sets from equivalent expressed gene sets for all relevant significance thresholds.

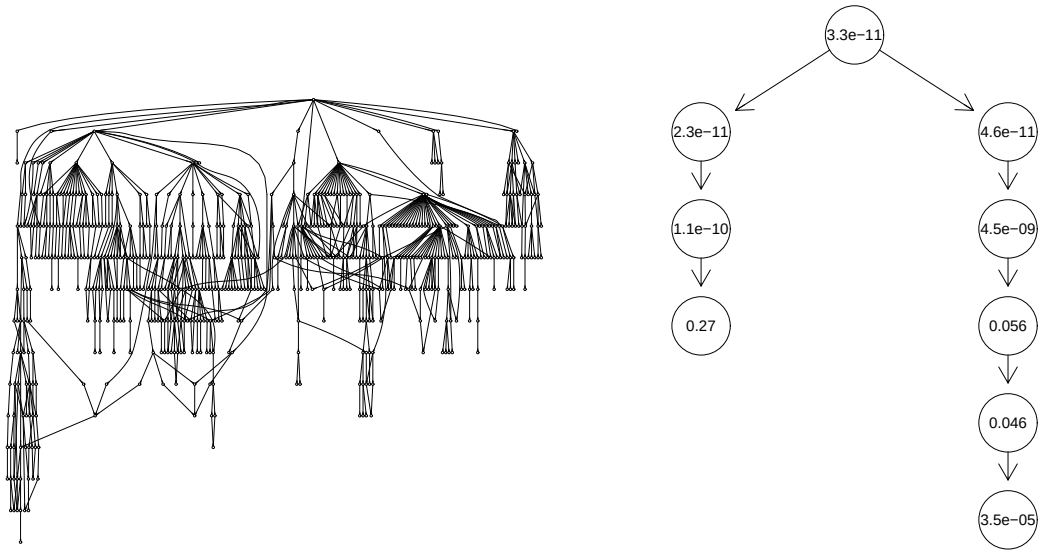
3.5 Application and Discussion

We applied our method to a well-known dataset collected by Golub et al. (1999). The dataset contains 7129 probe sets from the Affymetrix HuGeneFL Genome Array on 47 acute lymphocytic leukemia patients and 25 acute myeloid leukemia patients. Using version 2.0.1 of the hu6800 Bioconductor package, we were able to identify 1577 unique non-empty Gene Ontology (GO) terms from the molecular function on-

tology. The p-values were computed using Goeman's Global Test method (Goeman et al., 2004). Five MCMC chains were simulated from different starting parameter values. After a burn-in of 50k iterations, the chains converged. The convergence was partially judged by their Brooks-Gelman-Rubin statistic proposed by Brooks and Gelman (1998). The BGR was 1.00046, and BGR close to 1 indicates good convergence. We also looked at a trace plot for each parameter across chains, and they also showed good convergence behavior. After convergence, the medians of the posterior samples of parameters $\alpha, \beta, \omega, \alpha_0, \beta_0$ and λ were 0.18, 124.21, 0.47, 1.00, 10.86 and 0.62.

PPDE 0.95 was chosen as the cut off value and 547 GO terms were declared DE. The estimated FDR was 0.005. In comparison, the bottom-up method rejected 72 leaf nodes and 293 nodes overall when controlling FWER at 0.05. Applying Holm's method to all the nodes and rejecting the ancestors of rejected nodes yielded 353 total rejections at FWER level 0.05. Out of these 353 rejections, only one is not in the 547 rejections made by our HMM method.

Figure 3.4a shows the DAG for all the rejections. Figure 3.4b illustrates why our HMM method is more powerful than sequential FWER controlling methods. On the left branch, the leaf node has a p-value of 0.27, and it is the only node in this subgraph whose PPDE is below 0.95. This leaf node is the only leaf descendant for the node with a p-value 1.1e-10, and the bottom-up method will fail to reject any node in this branch. On the right branch, notice that one node in the middle has a p-value of 0.056. No top-down method controlling FWER at 0.05 level will go through this node. Thus the leaf node with a small p-value will be missed. In contrast, the HMM approach can overcome high p-values at leaf nodes as well as high p-values at nodes higher in the graph by making decisions at each node that account for p-values at all nodes in the graph.



(a) (b)

Figure 3.4: (a) DAG of all rejection in Section 3.5; (b) A subgraph of GO DAG with p-values annotated.

We are able to use both the information from data (p-values) and the structural information in the GO DAG to borrow the information across the nodes. For a node high in the GO graph hierarchy that contains a small portion of DE genes, the difference in high-dimensional multivariate distributions may be hard to detect because the difference exists for only a small subvector of the entire data vector. However, the HMM approach allows us to borrow information from descendants so that if a descendant consisting mostly of genes in the DE subvector is recognized as DE, we can correctly assign a high PPDE to the ancestor despite its unimpressive p-value.

Our method is primarily designed to distinguish between DE nodes and equivalent expressed nodes. For weak control of the FWER, i.e., control of FWER when all the null hypotheses are true, we suggest practitioners first use one of the FWER controlling methods as a gate keeper before applying our method. If the selected

FWER controlling procedure declares at least one gene set to be non-null, then our method can be used with no additional adjustment. If, however, the FWER controlling procedure declares no set to be non-null, then no further testing is done. Such a strategy will guarantee weak control of the FWER. None of the results reported in our paper would change with this modification because at least one rejection was obtained using FWER control in all simulation runs and in the data analysis.

3.6 Appendix

Let $\mathcal{T}$ be the original tree and $\mathcal{D}$ be the set of non-leaf nodes with state 1 within a tree, i.e.,

$$\mathcal{D}(\mathcal{T}) = \{i : i \in \mathcal{T}, S_i = 1 \text{ and } \mathcal{C}_i \neq \phi\}$$

Theorem 1. *The full conditional probability of any state configuration is*

$$\Pr(\boldsymbol{S}|\boldsymbol{p}, \boldsymbol{\theta}) = c_1^{S_1}(1-c_1)^{1-S_1} \prod_{i \in \mathcal{D}(\mathcal{T})} \left(\prod_{j \in \mathcal{C}_i} c_j^{S_j}(1-c_j)^{1-S_j} \right) \tag{3.10}$$

Proof. Note $\mathcal{P}_1 = \phi$ and $\Pr(P_{11}) = 1$. Thus (3.10) is equivalent to

$$\Pr(\boldsymbol{S}|P_{11}, \boldsymbol{p}, \boldsymbol{\theta}) = c_1^{S_1}(1-c_1)^{1-S_1} \prod_{i \in \mathcal{D}(\mathcal{T})} \left(\prod_{j \in \mathcal{C}_i} c_j^{S_j}(1-c_j)^{1-S_j} \right), \tag{3.11}$$

which we will prove by induction. For a tree with a single node,

$$\Pr(S_1 = 1|P_{11}, \boldsymbol{p}, \boldsymbol{\theta}) = c_1 \text{ and } \Pr(S_1 = 0|P_{11}, \boldsymbol{p}, \boldsymbol{\theta}) = 1 - c_1$$

directly by the definition of conditional probability.

For a tree $\widetilde{\mathcal{T}}$ whose root node is indexed by r and has n child nodes, let $\widetilde{\mathcal{T}}_i^c (i = 1, \ldots, n)$ represent the ith child tree of $\widetilde{\mathcal{T}}$, i.e., the sub-tree whose root is the ith child of r. Suppose $\widetilde{\mathcal{T}}_1^c, \ldots, \widetilde{\mathcal{T}}_n^c$ satisfy (3.11). Let r_i be the root of the $\widetilde{\mathcal{T}}_i^c$. Let $\boldsymbol{S}_i$ be the

state configuration of $\widetilde{\mathcal{T}}_i^c$. Let $\boldsymbol{S}0$ be a generic state configuration in which every node has state 0; its exact content depends on the tree in context.

$$\begin{aligned}
&\Pr(\boldsymbol{S} = \boldsymbol{S}0 | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&\quad = \Pr(\boldsymbol{S}_i = \boldsymbol{S}0, i = 1, \ldots, n | S_r = 0, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(S_r = 0 | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&\quad = \Pr(S_r = 0 | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&\quad = 1 - c_r
\end{aligned}$$

$$\begin{aligned}
&\Pr(S_r = 1, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_n | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&\quad = \Pr(S_r = 1 | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n | S_r = 1, P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&\quad = c_r \prod_{i=1}^{n} \Pr(\boldsymbol{S}_i | S_r = 1, \boldsymbol{p}, \boldsymbol{\theta}) \qquad (S_r = 1 \text{ implies } P_{r1}) \\
&\quad = c_r \prod_{i=1}^{n} \Pr(\boldsymbol{S}_i | P_{r_i 1}, \boldsymbol{p}, \boldsymbol{\theta}) \qquad (\text{node } r \text{ is the only parent of nodes } r_1, \ldots, r_n) \\
&\quad = c_r \prod_{i=1}^{n} \left[c_{r_i}^{S_{r_i}} (1 - c_{r_i})^{1 - S_{r_i}} \prod_{j \in \mathcal{D}(\widetilde{\mathcal{T}}_i^c)} \left(\prod_{k \in \mathcal{C}_j} c_k^{S_k} (1 - c_k)^{1 - S_k} \right) \right] \\
&\quad = c_r \prod_{i \in \mathcal{D}(\widetilde{\mathcal{T}})} \left(\prod_{j \in \mathcal{C}_i} c_j^{S_j} (1 - c_j)^{1 - S_j} \right)
\end{aligned}$$

This establishes that (3.11) holds for a tree $\widetilde{\mathcal{T}}$ as long as (3.11) holds for all the child trees of $\widetilde{\mathcal{T}}$. Because (3.11) holds for a single node, it then holds for a two-level tree. By induction, the result follows. □

Bibliography

Allison, D., Gadbury, G., Heo, M., Fernández, J., Lee, C., Prolla, T., and Weindruch, R. (2002), "A Mixture Model Approach for the Analysis Of Microarray Gene Expression Data," *Computational Statistics and Data Analysis*, 39, 1–20.

Allison, D. B., Cui, X., Page, G. P., and Sabripour, M. (2006), "Microarray Data Analysis: from Disarray to Consolidation and Consensus," *Nature Reviews Genetics*, 7, 55–65.

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al. (2000), "Gene Ontology: Tool for the Unification of Biology," *Nature Genetics*, 25, 25–29.

Barry, W. T., Nobel, A. B., and Wright, F. A. (2005), "Significance Analysis of Functional Categories in Gene Expression Studies: a Structured Permutation Approach," *Bioinformatics*, 21, 1943–1949.

Brooks, S. and Gelman, A. (1998), "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J., and Foa, R. (2004), "Gene Expression Profile of Adult T-cell Acute Lymphocytic Leukemia Identifies Distinct Subsets of Patients with Different Response to Therapy and Survival," *Blood*, 103, 2771–2778.

Chib, S. (1996), "Calculating Posterior Distributions and Modal Estimates in Markov Mixture Models," *Journal of Econometrics*, 75, 79–97.

Darwiche, A. (2009), *Modeling and Reasoning with Bayesian Networks*, Cambridge University Press.

Efron, B. (2004), "Large-scale Simultaneous Hypothesis Testing: the Choice of a Null Hypothesis," *Journal of the American Statistical Association*, 99, 96–105.

Efron, B. and Tibshirani, R. (2007), "On Testing the Significance of Sets of Genes," *Annals of Applied Statistics*, 1, 107–129.

Goeman, J. and Buhlmann, P. (2007), "Analyzing Gene Expression Data in Terms of Gene Sets: Methodological Issues," *Bioinformatics*, 23, 980.

Goeman, J. J. and Mansmann, U. (2008), "Multiple Testing on the Directed Acyclic Graph of Gene Ontology," *Bioinformatics*, 24, 537–544.

Goeman, J. J., van de Geer, S. A., de Kort, F., and van Houwelingen, H. C. (2004), "A Global Test for Groups of Genes: Testing Association with a Clinical Outcome," *Bioinformatics*, 20, 93–99.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., et al. (1999), "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, 286, 531.

Holm, S. (1979), "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, 6, 1979.

Khatri, P. and Draghici, S. (2005), "Ontological Analysis of Gene Expression Data: Current Tools, Limitations, and Open Problems," *Bioinformatics*, 21, 3587–3595.

Liu, J., Hughes-Oliver, J. M., and Menius, A. J. (2007), "Domain-enhanced Analysis of Microarray Data using GO Annotations," *Bioinformatics*, 23, 1225–1234.

Mansmann, U. and Meister, R. (2005), "Testing Differential Gene Expression in Functional Groups. Goeman's Global Test versus an ANCOVA Approach." *Methods of Information in Medicine*, 44, 449–53.

Marcus, R., Eric, P., and Gabriel, K. (1976), "On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance," *Biometrika*, 63, 655–660.

Meinshausen, N. (2008), "Hierarchical Testing of Variable Importance," *Biometrika*, 95, 265.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, 21, 1087.

Mielke, P. and Berry, K. (2001), *Permutation Methods: A Distance Function Approach*, Springer.

Nettleton, D., Recknor, J., and Reecy, J. M. (2008), "Identification of Differentially Expressed Gene Categories in Microarray Studies using Nonparametric Multivariate Analysis," *Bioinformatics*, 24, 192–201.

Newton, M., Quintana, F., den Boon, J., Sengupta, S., and Ahlquist, P. (2007), "Random-set Methods Identify Distinct Aspects of the Enrichment Signal in Gene-set Analysis," *Annals of Applied Statistics*, 1, 85–106.

Scott, S. (2002), "Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century," *Journal of the American Statistical Association*, 97, 337–352.

Subramanian, A., Tamayo, P., Mootha, V., Mukherjee, S., Ebert, B., Gillette, M., Paulovich, A., Pomeroy, S., Golub, T., Lander, E., et al. (2005), "Gene Set Enrichment Analysis: A Knowledge-based Approach for Interpreting Genome-wide Expression Profiles," *Proceedings of the National Academy of Sciences*, 102, 15545–15550.

Sun, W. and Cai, T. (2009), "Large-scale Multiple Testing Under Dependence," *Journal of the Royal Statistical Society, Series B*, 71, 393–424.

Tomfohr, J., Lu, J., and Kepler, T. B. (2005), "Pathway Level Analysis of Gene Expression using Singular Value Decomposition," *BMC Bioinformatics*, 6, 225.

Westfall, P. and Young, S. (1993), *Resampling-based Multiple Testing: Examples and Methods for p-value Adjustment*, Wiley-Interscience.

Yekutieli, D. (2008), "Hierarchical False Discovery Rate-Controlling Methodology," *Journal of the American Statistical Association*, 103, 309–316.

CHAPTER 4. A Hidden Markov Tree Model for Multiple Hypotheses Testing of Gene Ontology Gene Sets

A paper to be submitted to *Biometrics*

Kun Liang and Dan Nettleton

Department of Statistics

Iowa State University, Ames, IA 50011

email: liangkun@iastate.edu

Abstract

Testing predefined gene categories has become a common practice for scientists analyzing high throughput transcriptome data. A systematic way of testing gene categories leads to testing hundreds of null hypotheses that correspond to nodes in a directed acyclic graph. The relationships among gene categories induce logical restrictions among the corresponding null hypotheses. Liang and Nettleton (2010) proposed a fully Bayesian method that incorporates the dependence information among the null hypotheses by using a hidden Markov model. The method was shown to have better performance than other existing methods, but it is computationally intensive. Under a hidden Markov tree model, we develop a more computationally efficient method. Our method also provides more powerful results than existing methods that honor

the logical restrictions. The method is illustrated by testing Gene Ontology terms for evidence of differential expression in an expression quantitative trait loci study.

KEY WORDS: Deterministic annealing; Differential expression; Directed acyclic graph; Expectation maximization; Expression quantitative trait loci; False discovery rate; Gene set enrichment analysis; Microarray; Multiple testing; RNA-seq; Simultaneous inference.

4.1 Introduction

An important challenge facing scientists is how to interpret and report the results from high throughput transcriptome experiments, for example, microarray and RNA-seq experiments. Thousands of genes are measured simultaneously from subjects under different treatment conditions. A routine analysis, e.g., a two sample t-test for each gene on a microarray, produces a list of genes that are declared to be differential expressed (DE) across conditions. The length of the DE gene list can run up to a few thousand, and this makes the interpretation and reporting of the results a challenging task. However, genes are known to work collaboratively to regulate or participate in biological processes, to perform molecular functions and to produce gene products that form cell components. Thus, it is intuitive and useful to interpret and report results in terms of meaningful gene sets instead of individual genes (Allison et al., 2006). It has become a common practice for scientists to test whether some predefined gene categories/sets are differential expressed. Gene Ontology (GO) (Ashburner et al., 2000) is one of the most popular sources of gene set definitions. GO provides a controlled vocabulary of terms that form a directed acyclic graph (DAG) with directed edges drawn from general terms to more specific terms. The genes that share a GO

term comprise a well defined gene set. Each GO term and its gene set correspond to a node in the GO DAG. The genes annotated to a specific term are automatically annotated to the more general terms linked by directed edges. Thus, the directed edges also indicate gene set subset relationships. Testing these predefined gene sets on the GO DAG yields meaningful results that are relatively easy to interpret.

Suppose for treatment conditions $t = 1, \ldots, T$ and experimental units $u = 1, \ldots, n_t$; $\boldsymbol{X}_{tu}$ is a vector of expression measurements with one element for each of P genes on a microarray. For $i = 1, \ldots, N_G$; suppose $\boldsymbol{G}_i$ is a indicator matrix whose rows are a subset of the $P \times P$ identity matrix such that $\boldsymbol{G}_i\boldsymbol{X}_{tu}$ is the subvector of expression values for the genes in the ith GO gene set and the uth experimental unit of the tth treatment group. Furthermore, suppose that $\boldsymbol{G}_i\boldsymbol{X}_{tu} \sim F_t^{(i)}$ for all $i = 1, \ldots, N_G$; $t = 1, \ldots, T$; and $u = 1, \ldots, n_t$. We consider the problem of testing

$$H_0^{(i)} : F_1^{(i)} = \cdots = F_T^{(i)} \tag{4.1}$$

for $i = 1, \ldots, N_G$. An important goal of biological research is to identify gene sets (or, equivalently, nodes in the GO DAG) for which $H_0^{(i)}$ is false (DE nodes) because these are the gene sets whose multivariate expression distribution changes with treatment.

Testing the gene sets on the GO DAG is a very challenging task. First, the size of a gene set ranges from a few to thousands, often larger than the sample size ($n_1 + \cdots + n_T$). Second, the correlation structure among genes is unknown and expected to be non-trivial. Third, many of the gene sets share genes with others so they can't be assumed to be independent even if genes were independent. Finally, there are logical relationships among the N_G null hypotheses due to the subset relationships among the gene sets that should be accounted for in inference. In particular, if node i is a parent of node j, then the truth of $H_0^{(i)}$ implies the truth of $H_0^{(j)}$ because the expression vector for gene set j is a subvector of the expression vector for gene set i. Furthermore, the

truth of $H_0^{(i)}$ implies the truth of the null hypotheses for all descendants of node i in the GO graph. These structural restrictions implied by the GO graph make the inference complicated. On the other hand, we should explore ways that exploit these structural dependences to make better inferences.

Sequential testing methods used to be the only procedures available to honor these logical restrictions, but such methods cannot fully utilize all available information and suffer from loss of power in large-scale applications. Liang and Nettleton (2010) proposed a method that circumvent the drawback of the sequential methods by taking the whole graph into account. Their method is fully Bayesian and was shown to have better receiver operating characteristic than other existing methods. In short, by incorporating the structural dependence information among the null hypotheses into the model, Liang and Nettleton (2010) turned the structural restrictions on the GO DAG into information and were able to gain a power advantage over existing methods. However, the implementation of Liang and Nettleton (2010) relies on Markov chain Monte Carlo (MCMC) sampling, which can be computationally intensive. There are many circumstances in which a faster approach is needed.

A prime example involves a generalization of expression quantitative trait loci (eQTL) studies. In eQTL studies, a goal is to determine whether variation in DNA at a particular genomic location is associated with variation in the expression of one or more genes. Tens, hundreds, or thousands of genomic locations may be scanned for association with thousands of genes. A natural generalization of eQTL mapping involves testing genomic locations for association with gene sets rather than individual genes. In principle, the approach of Liang and Nettleton (2010) could be used for each of many genetic markers to identify associations between markers and traits. However, as the number of markers grows, this strategy quickly becomes computation-

ally intractable. Thus, we develop an alternative and more computationally efficient implementation in this paper.

We review background for the problem at hand in Section 4.2. In Section 4.3, we present a hidden Markov tree model (HMT) approach to testing multiple gene sets on a tree-transformed GO DAG. Then we evaluate its performance through data-driven simulation in Section 4.4. The paper concludes with an example application and discussion in Section 4.5.

4.2 Background

To address the challenges of gene set testing, many statistical methods have been proposed. Among them, many of the early approaches are based on test statistics derived from individual genes. These methods have subsequently been reviewed and criticized on statistical grounds in Khatri and Draghici (2005), Allison et al. (2006), Goeman and Buhlmann (2007) and Nettleton et al. (2008). The criticism can be summarized as follows. First, the majority of these methods unrealistically assume gene independence, and their resulting p-values can be wildly anti-conservative. Second, most of these methods are based on the competition between genes within and outside a gene set, which may not be the main interest of biological research. Third, many methods are based on single gene statistics that cannot detect many types of multivariate expression change. We will use the Fisher's exact test, which is the most widely used method, as an example to illustrate these points.

The Fisher's exact test is conceptually simple and easy to understand yet suffers all three aforementioned drawbacks. Fisher's exact test is used to test whether a certain gene set is "enriched" or "over-represented" on a list of genes declared to be DE. Each gene on a microarray can be classified as belonging to a certain gene set or

not and cross-classified as being on the DE gene list or not. Thus, all the genes can be arranged in a 2×2 table according to their classifications to these two criteria as shown in Table 4.1.

Table 4.1: A 2×2 table of gene classification for a certain gene set.

	DE gene	Non-DE gene	Total
In gene set	m_{SD}	m_{SD^c}	m_S
Not in gene set	m_{S^cD}	$m_{S^cD^c}$	m_{S^c}
	m_D	m_{D^c}	m

Assuming genes are independent and the margins of the table are fixed, the probability of a particular table configuration can be computed. More specifically, if we randomly sample without-replacement m_S genes out of a total of m genes, the probability that m_{SD} of the m_S genes are on a DE gene list of length m_D follows a hypergeometric distribution. However, the validity of the Fisher's exact test relies on the assumption of gene independence, which is clearly false from the biological point of view. Furthermore, the test is not testing whether the genes in a certain gene set are differentially expressed across conditions. Rather, Fisher's exact test is actually testing whether the gene set contains a significantly different proportion of DE genes than the rest of the genes outside of the gene set. Finally, Fisher's exact test only uses gene-specific statistics (whether genes are on the DE gene list or not), and thus, it has no power to detect multivariate distributional differences that are beyond marginal differences. An example of such power loss is illustrated in Nettleton et al. (2008) where the joint expression distribution for genes in a set A is identical across treatment conditions, the joint expression distribution for genes in a set B is identical across treatment conditions, but the joint expression distribution for genes in $A \cup B$ varies across treatment

conditions.

In recent years, a viable alternative has been developed. Many authors have proposed methods to test multivariate gene set differences as in (4.1). This class of methods include Goeman's Global Test (Goeman et al., 2004), Mansmann's Global Ancova (Mansmann and Meister, 2005), the Multiple Response Permutation Procedure (MRPP, Mielke and Berry, 2001), Pathway Level Analysis of Gene Expression (Tomfohr et al., 2005), and Domain-Enhanced Analysis (Liu et al., 2007) among others. The multivariate tests avoid the unrealistic assumption of gene independence and are potentially more powerful than the individual gene tests combined. Thus, the multivariate gene set tests will be our methods of choice in this study.

As a consequence of testing for equality of multivariate distributions within each node of the hierarchical GO DAG, only some configurations of true and false null hypotheses are possible. More specifically, if the null hypothesis holds for a gene set A then it should hold for all subsets of A, which include all the descendants of A in a GO DAG. Most of the methods honoring this logical consistency that are applicable to a GO DAG are sequential methods, each of which can be generally classified as a *top-down* or a *bottom-up* procedure (Goeman and Mansmann, 2008). Both procedures are designed to control family-wise error rate (FWER). The top-down procedure is based on the closed testing procedure of Marcus et al. (1976), but it is computational prohibitive for large graphs like a GO DAG. The bottom-up procedure only tests the leaf nodes of a graph (the nodes without children) and declares significance of some leaf nodes according to a certain FWER control procedure. Then a higher level GO node can be declared significant whenever it has any significant leaf descendant. Goeman and Mansmann (2008) proposed a focus-level method which can be viewed as a combination or compromise between top-down and bottom-up procedures. All

sequential methods are subject to power loss due to the fact that a rejection decision has to be made at each step with no regard to the information beyond the current step. For example, if FWER is controlled at the 0.05 level, then a node with a p-value of 0.051 will be an impasse for the top-down procedure even if the p-value associated with one of its descendant nodes is very small (this could happen when the descendant node has a high concentration of DE genes while the ancestor is "diluted" by many equivalently expressed genes). On the other hand, a DE node's leaf descendants could all be null nodes, which would render the power for detecting such a DE node to be negligible for a bottom-up procedure.

Liang and Nettleton (2010) proposed the first method that incorporates nearly all the information in the GO DAG to provide a coherent hypotheses testing solution. The method first transforms a GO DAG into a GO tree and then models the p-values of the gene sets on the GO tree using a hidden Markov model (HMM). Using a fully Bayesian framework, the method estimates the posterior probabilities of differential expression (PPDE) for each GO gene set through MCMC sampling. It was shown that the method has better receiver operating characteristic than other existing methods. However, the MCMC sampling process can be computationally demanding, so we develop a more efficient alternative solution in the following sections.

4.3 The Proposed Approach

The logical constraints among the null hypotheses on a GO DAG induce a natural Markov model on the states of the null hypotheses, but exact computation on a complex graph like the GO DAG is computationally prohibitive (Liang and Nettleton, 2010). Thus, following Liang and Nettleton (2010), we transform a GO DAG into a GO tree to facilitate the computation. Then, a single p-value for testing the null hy-

pothesis in (4.1) is computed separately for each node in the GO tree. We then model the joint distribution of these tree node p-values using a hidden Markov tree model. We treat the state of each null hypothesis as a random variable and propose a Markov model for the joint distribution of states. This Markov model places zero probability on any configuration of states that is not consistent with the logical constraints imposed by the structure of the GO tree.

We summarize the tree transformation and hidden Markov model in Liang and Nettleton (2010) in Sections 4.3.1 and 4.3.2. Then we use a hidden Markov tree model to obtain the maximum likelihood estimates of the parameters. Furthermore, instead of sampling state configurations given the parameters, we deterministically compute the probabilities of the original DAG nodes being DE. Thus, the new implementation dramatically reduces the computational expense of the estimation process.

4.3.1 Tree Transformation of a GO DAG

Transforming a GO DAG into a tree structure can make computation feasible on one hand and greatly reduce the sharing of genes and dependences among gene sets on the other hand. The tree transformation process is illustrated using a tiny example in Figure 4.1. Interested readers can refer to Section 3.1 of Liang and Nettleton (2010) for a more detailed description of the process. The basic idea of the tree transformation is as follows. If we remove all but one incoming edges for each node that has multiple parents, the graph becomes a tree. This is equivalent to removing the genes in the child node from all but one of its parent nodes. For example, see the removal of the edge from node 2 to 4 in Figure 4.1a.

After the procedure, every node except the root node will have one and only one parent, and thus, the DAG will be transformed into a tree. Each of the original

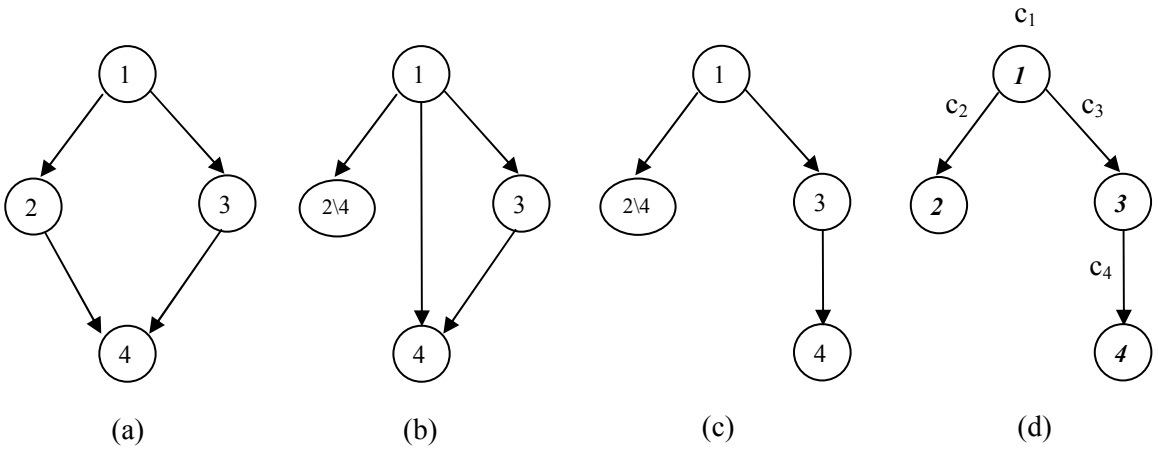


Figure 4.1: DAG to Tree: (a) Original DAG; (b) After remove genes in node 4 from node 2; (c) Tree after remove redundant edge from node 1 to node 4; (d) Tree nodes renumbered with bold and italic numbers.

DAG nodes will be a union of one or more tree nodes. For example, DAG node 2 in Figure 4.1a is a union of tree nodes 2 and 4 in Figure 4.1d. More formally, for $j = 1, \ldots, N_G$; let $\mathcal{G}_j$ be the gene set corresponding to GO DAG node j. For $i = 1, \ldots, N_T$; let $\mathcal{T}_i$ be the set of genes that are in GO tree node i. Let $\mathcal{GT}_j$ denote the set of tree nodes/indices whose corresponding gene sets are subsets of $\mathcal{G}_j$, i.e., $\mathcal{GT}_j = \{k = 1, \ldots, N_T : \mathcal{T}_k \subseteq \mathcal{G}_j\}$. The tree transformation process guarantees that the original DAG node can be reconstructed from its comprising tree nodes, i.e., $\mathcal{G}_j = \bigcup_{k \in \mathcal{GT}_j} \mathcal{T}_k$. Let the state of ith GO tree node be S_i. Let $S_i = 0$ if $H_0^{(i)}$ is true and let $S_i = 1$ if $H_0^{(i)}$ is false. For the jth GO DAG node, define

$$S_j^* = \max\{S_k : k \in \mathcal{GT}_j\}. \tag{4.2}$$

Note that $S_j^* = 1$ implies that the state of GO DAG node j is 1 because a vector of genes corresponding to a gene set must have different multivariate distributions across conditions if any subvector does. It is straightforward to show this conversion guarantees the logical consistency of states $\{S_j^* : j = 1, \ldots, N_G\}$ for the original GO

DAG. In Section 4.3.5, we will show how to estimate, for $j = 1, \ldots, N_G$, the probability that $S_j^* = 1$ using the results derived from a HMT on the corresponding GO tree.

4.3.2 A Hidden Markov Tree Model for p-values on the GO Tree

By the nature of the null hypothesis of multivariate distribution equivalence in (4.1) and the subset relationship among GO tree gene sets, a node must be in state 0 if its parent node is in state 0. On the other hand, a node whose parent is in state 1 can be in state 1 with some unknown probability. This conditional dependence scenario clearly demonstrates the Markov property.

Thus, the hidden Markov tree model (HMT) is proposed as follows. Let S_i be as defined in Section 4.3.1, and let p_i be the p-value associated with GO tree node i (gene set i) that is computed by testing (4.1) using any method that produces a valid p-value. Then the HMT is composed of an observed random tree $\boldsymbol{p} = \{p_1, \ldots, p_{N_T}\}$ and an unobserved random tree $\boldsymbol{S} = \{S_1, \ldots, S_{N_T}\}$. Both trees have the same index structure. Let $\rho(i)$ denote the index of the parent node of node i. The transition portion of our HMT is

$$\mathcal{P}(S_i = 0 | S_{\rho(i)} = 0) = 1 \quad \text{and} \quad \mathcal{P}(S_i = 1 | S_{\rho(i)} = 1) = \omega, \tag{4.3}$$

for some $\omega \in (0, 1)$. To streamline the expressions of recursion in Section 4.3.3, we express (4.3) in an equivalent way through the generic definition of transition probabilities. Let $q_{jk} = \mathcal{P}(S_i = k | S_{\rho(i)} = j)$ be the transition probability from a parent node in state j to a child node in state k, and thus, $q_{00} = 1, q_{01} = 0, q_{10} = 1 - \omega$ and $q_{11} = \omega$. Furthermore, we assume the root node of the tree (the node with no parent) is in state 1 with some probability $\pi \in (0, 1)$. To model the observed p-values

given the hidden states, we consider the model

$$\begin{cases} p_i \sim f_0(\lambda, \alpha_0, \beta_0) = \lambda + (1-\lambda)\text{beta}(\alpha_0, \beta_0) & \text{if } S_i = 0 \\ p_i \sim f_1(\alpha, \beta) = \text{beta}(\alpha, \beta) & \text{if } S_i = 1 \end{cases} \tag{4.4}$$

with p-values assumed to be conditionally independent of one another given the states. The parameters α and β for the p-value density of false nulls are restricted to be in $(0, 1]$ and $(1, \infty)$, respectively, so that a strictly decreasing p-value density is guaranteed for DE gene sets. The p-value density of true nulls is assumed to be a mixture of uniform and unimodal beta, where λ denotes the mixing proportion. The parameters α_0 and β_0 are restricted to be bigger than 1 so that a unimodal p-value density is guaranteed. Notice that a uniform model or a unimodal beta model is a degenerated case of this mixture model. In most cases, a simple uniform model will work well. However, the null mixture model is designed to adapt to the possible deviation from the uniform distribution caused by positive correlations among the null gene sets due to the sharing of genes and correlations among genes. This alteration of the commonly used uniform null p-value distribution is similar in spirit to the approach of Efron (2004) who recommends using data to estimate an "empirical" null distribution.

Let $\boldsymbol{\theta} = \{\pi, \omega, \alpha, \beta, \lambda, \alpha_0, \beta_0\}$, the collection of all HMT parameters. Liang and Nettleton (2010) used a Bayesian approach that assumes $\boldsymbol{\theta}$ to be random with diffuse priors. To speed up the estimation, we assume in this paper that $\boldsymbol{\theta}$ is a vector of fixed unknown parameters to be estimated. These two approaches are expected to give similar results.

4.3.3 Upward-downward Algorithm for HMT

The forward-backward algorithm is widely used in hidden Markov chain applications; its parallel in hidden Markov tree models is the upward-downward algorithm

developed by Ronen et al. (1995) and Crouse et al. (1998). Durand et al. (2004) reformulated the algorithm to make the algorithm numerically stable. Given the parameter vector $\boldsymbol{\theta}$, the upward-downward algorithm leads to efficient computation of the likelihood, $\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{p})$. Furthermore, the results from the upward-downward algorithm are useful in obtaining the maximum likelihood estimates of parameters (Section 4.3.4) and computing probabilities of differential expression of the nodes on the original GO DAG (Section 4.3.5). We formulate our HMT on the GO tree in the framework of Durand et al. (2004) as follows.

Without loss of generality, let the root node of the GO tree be indexed by 1. Let $i = 1, \ldots, N_T$ be any GO tree node index and $k = 0$ or 1 be a possible state of a node. Let $\mathcal{C}(i)$ denote the set of indices of node i's children nodes. Let $\mathfrak{T}(i)$ denote the subtree whose root is node i. Let $\boldsymbol{p}_i$ be a vector of p-values corresponding to the subtree rooted at node i, i.e., $\boldsymbol{p}_i$ is a vector whose elements are $\{p_l : l \in \mathfrak{T}(i)\}$. Denote $\boldsymbol{p}_{i\backslash j}$ as a vector of p-values corresponding to the nodes in subtree $\mathfrak{T}(i)$ but not in $\mathfrak{T}(j)$, i.e., $\boldsymbol{p}_{i\backslash j}$ is a vector whose elements are $\{p_l : l \in \mathfrak{T}(i); l \notin \mathfrak{T}(j)\}$. Let $f(\cdot)$ and $f(\cdot|\cdot)$ denote a generic density and conditional density, respectively, whose precise definition are easily inferred from function arguments. Assuming $\boldsymbol{\theta}$ is known, we define three quantities that can be computed efficiently by recursion:

$$\begin{aligned} \tau_i(k) &= \mathcal{P}(S_i = k|\boldsymbol{p}_i); \\ \tau_{\rho(i),i}(k) &= \frac{f(\boldsymbol{p}_i|S_{\rho(i)} = k)}{f(\boldsymbol{p}_i)}; \\ \kappa_i(k) &= \frac{f(\boldsymbol{p}_{1\backslash i}|S_i = k)}{f(\boldsymbol{p}_{1\backslash i}|\boldsymbol{p}_i)}. \end{aligned}$$

First we compute the marginal state probabilities $\mathcal{P}(S_i = k)$ for $i = 1, \ldots, N_T$ and $k = 0$ or 1 in a downward recursion, i.e., $\mathcal{P}(S_1 = k) = \pi^k(1-\pi)^{1-k}$ and $\mathcal{P}(S_i = k) = \sum_j q_{jk}\mathcal{P}(S_{\rho(i)} = j)$ for $i > 1$. Then the $\tau_i(k)$ quantities can be computed recursively

in an upward fashion. For any leaf node i, $\tau_i(k)$ is initialized as

$$\tau_i(k) = \frac{f(p_i|S_i = k)\mathcal{P}(S_i = k)}{N_i},$$

where $N_i = \sum_k f(p_i|S_i = k)\mathcal{P}(S_i = k)$ is a normalizing factor for the leaf node i such that $\sum_k \tau_i(k) = 1$. An upward computation for a non-leaf node is

$$\tau_i(k) = \frac{f(p_i|S_i = k)\mathcal{P}(S_i = k)\prod_{\nu\in\mathcal{C}_{(i)}} \tau_{i,\nu}(k)}{N_i},$$

where the normalizing factor N_i is again a summation of the numerators over all possible ks, i.e., $N_i = \sum_k \left[f(p_i|S_i = k)\mathcal{P}(S_i = k)\prod_{\nu\in\mathcal{C}_{(i)}} \tau_{i,\nu}(k)\right]$ for the non-leaf node. The $\tau_{\rho(i),i}(k)$ quantities can be derived from the $\tau_i(k)$s as follows:

$$\tau_{\rho(i),i}(k) = \sum_j \frac{\tau_i(j)q_{kj}}{\mathcal{P}(S_i = j)}.$$

Note that the upward recursion process requires us to compute $\tau_i(k)$s for the leaf nodes first, then $\tau_{\rho(i),i}(k)$s for the leaf nodes, then $\tau_i(k)$s for the parents of the leaf nodes, and so forth.

The $\kappa_i(k)$ quantities are computed in a downward fashion. After we initialize $\kappa_1(0) = \kappa_1(1) = 1$, the downward recursion is

$$\kappa_i(k) = \frac{1}{P(S_i = k)} \sum_j \frac{q_{jk}\tau_{\rho(i)}(j)\kappa_{\rho(i)}(j)}{\tau_{\rho(i),i}(j)}.$$

It can be shown that the log-likelihood $l(\boldsymbol{\theta}|\boldsymbol{p}) = \sum_i \log N_i$, which is useful for monitoring the convergence of the expectation maximization (EM) algorithm in the next subsection.

4.3.4 Deterministic Annealing EM Algorithm

The EM algorithm (Dempster et al., 1977) is commonly used for estimating the parameters of a hidden Markov model. For example, the widely used Baum-Welch

algorithm (Baum et al., 1970) is a special case of the EM algorithm. We will show how to find $\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, l(\boldsymbol{\theta}|\boldsymbol{p})$, the maximum likelihood estimate of $\boldsymbol{\theta}$, through EM. In this subsection we first use the results from the upward-downward algorithm to apply the EM algorithm. Then we will use a modified version of the EM algorithm, deterministic annealing EM (DAEM), to make the procedure robust to the initial parameter values.

For the E step of the EM algorithm,

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbf{E}_{\boldsymbol{S}|\boldsymbol{p},\boldsymbol{\theta}^{(t)}}[\log \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{p},\boldsymbol{S})] \\ &= \mathbf{E}_{\boldsymbol{S}|\boldsymbol{p},\boldsymbol{\theta}^{(t)}}\Big[S_1 \log\pi + (1-S_1)\log(1-\pi) + \\ &\quad \sum_{i=2}^{N_T} \mathrm{I}(S_{\rho(i)}=1, S_i=1)\log\omega + \sum_{i=2}^{N_T} \mathrm{I}(S_{\rho(i)}=1, S_i=0)\log(1-\omega) + \\ &\quad \sum_{i=1}^{N_T} S_i \log f_1(p_i|\alpha,\beta) + \sum_{i=1}^{N_T}(1-S_i)\log f_0(p_i|\lambda,\alpha_0,\beta_0)\Big]. \end{aligned}$$

In the $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ expression, the conditional expectations for the terms associated with S_is can be derived separately as follows:

$$\begin{aligned} \mathbf{E}\left(S_i|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right) = \mathcal{P}\left(S_i=1|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right) &= \tau_i^{(t)}(1)\kappa_i^{(t)}(1); \\ \mathbf{E}\left[\mathrm{I}(S_{\rho(i)}=1, S_i=1)|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right] &= \frac{\tau_i^{(t)}(1)\omega^{(t)}\mathbf{E}(S_{\rho(i)}|\boldsymbol{p},\boldsymbol{\theta}^{(t)})}{\mathcal{P}(S_i=1)\tau_{\rho(i),i}^{(t)}(1)}; \\ \mathbf{E}\left[\mathrm{I}(S_{\rho(i)}=1, S_i=0)|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right] &= \frac{\tau_i^{(t)}(0)(1-\omega^{(t)})\mathbf{E}(S_{\rho(i)}|\boldsymbol{p},\boldsymbol{\theta}^{(t)})}{\mathcal{P}(S_i=0)\tau_{\rho(i),i}^{(t)}(1)}. \end{aligned}$$

In the M step, we obtain $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\text{argmax}}\, Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. Let $\mathcal{P}_{1k} = \sum_{i=2}^{N_T} \mathbf{E}\left[\mathrm{I}(S_{\rho(i)}=1, S_i=k)|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right]$, $k=0$ or 1. By solving score functions, we have

$$\begin{aligned} \pi^{(t+1)} &= \mathbf{E}\left(S_1|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right), \\ \text{and}\quad \omega^{(t+1)} &= \frac{\mathcal{P}_{11}}{\mathcal{P}_{11}+\mathcal{P}_{10}}. \end{aligned}$$

The parameters α and β can be estimated by numerically maximizing a sum of weighted log-likelihoods given by $\sum_{i=1}^{N_T} w_i \log f_1(p_i|\alpha,\beta)$, where $w_i = \mathbf{E}\left(S_i|\boldsymbol{p},\boldsymbol{\theta}^{(t)}\right)$ for $i = 1,\ldots,N_T$. The parameters λ, α_0 and β_0 can be estimated similarly.

However, the EM result can highly depend on its initial parameter values especially in a multivariate context like ours. A commonly used ad hoc procedure is to perform EM from many different starting values, but the result is far from satisfactory. Ueda and Nakano (1998) proposed a deterministic annealing EM (DAEM) algorithm to alleviate EM's dependency on starting values. Through the principle of the maximum entropy, they derived a posterior that is parameterized by a "temperature" parameter, which is used to control an "annealing process."

Adapting the DAEM framework to our HMT problem, the posterior of the unobserved/missing data $\boldsymbol{S}$ is

$$f_\gamma(\boldsymbol{S}|\boldsymbol{p},\boldsymbol{\theta}) = \frac{f^\gamma(\boldsymbol{p},\boldsymbol{S}|\boldsymbol{\theta})}{\int f^\gamma(\boldsymbol{p},\boldsymbol{S}|\boldsymbol{\theta})\ \mathrm{d}\boldsymbol{S}},$$

where $1/\gamma$ corresponds to the "temperature." Note that if $\gamma = 1$ then the posterior is exactly the same as a regular posterior. On the other hand, when γ is close to 0 (temperature is high), $f^\gamma(\boldsymbol{p},\boldsymbol{S}|\boldsymbol{\theta})$ is close to 1 and is insensitive to the values of $\boldsymbol{p}$ and $\boldsymbol{\theta}$. So if we start with a high temperature, the impact of initial parameter values will be minimized. The DAEM operates according to a deterministic schedule of temperatures, in which the temperature drops from high to low (γ changes from near zero to 1). At each temperature/γ, an EM algorithm is used to estimate $\boldsymbol{\theta}$ assuming the conditional distribution of $\boldsymbol{S}$ given $\boldsymbol{p}$ and $\boldsymbol{\theta}$ follows $f_\gamma(\boldsymbol{S}|\boldsymbol{p},\boldsymbol{\theta})$. The DAEM starts from a vector of random initial parameter values and uses the parameter estimates from the previous step/temperature as the starting values at each subsequent step. Running the DAEM leads to running an EM at each temperature, and thus, the DAEM is slower than a single run of EM. So the DAEM can be thought of as trading

time for the procedure's robustness to starting parameter values.

To implement the DAEM in our HMT problem, notice that $f(\boldsymbol{p}, \boldsymbol{S}|\boldsymbol{\theta})$ can be expressed as a product of probabilities and densities raised to the power of indicators of the hidden states, and thus, $f^{\gamma}(\boldsymbol{p}, \boldsymbol{S}|\boldsymbol{\theta})$ has the effect of making all the probabilities and densities raised to the power γ, i.e.,

$$\begin{aligned} f^{\gamma}(\boldsymbol{p}, \boldsymbol{S}|\boldsymbol{\theta}) &= \pi^{\gamma S_1}(1-\pi)^{\gamma(1-S_1)} \prod_{i=2}^{N_T} \omega^{\gamma \mathrm{I}(S_{\rho(i)}=1, S_i=1)} \prod_{i=2}^{N_T} (1-\omega)^{\gamma \mathrm{I}(S_{\rho(i)}=1, S_i=0)} \\ & \quad \prod_{i=1}^{N_T} f_1^{\gamma S_i}(p_i|\alpha, \beta) f_0^{\gamma(1-S_i)}(p_i|\lambda, \alpha_0, \beta_0). \end{aligned}$$

That is, in the E step where we calculate the conditional expectation of S_i and $\mathrm{I}(S_{\rho(i)} = 1, S_i = k)$ under the conditional distribution $f(\boldsymbol{S}|\boldsymbol{p}, \boldsymbol{\theta})$, if we use $\pi^{\gamma}, (1-\pi)^{\gamma}, \omega^{\gamma}, (1-\omega)^{\gamma}, f_1^{\gamma}(p_i|\alpha, \beta)$ and $f_0^{\gamma}(p_i|\lambda, \alpha_0, \beta_0)$ in the places of $\pi, 1-\pi, \omega, (1-\omega), f_1(p_i|\alpha, \beta)$ and $f_0(p_i|\lambda, \alpha_0, \beta_0)$, we are effectively calculating the expectation under $f_{\gamma}(\boldsymbol{S}|\boldsymbol{p}, \boldsymbol{\theta}^{(t)})$. A similar adaptation of the DAEM has been used in Granat and Donnellan (2002), where the deterministic annealing method was applied to a hidden Markov chain.

4.3.5 Compute Probabilities for the Original GO DAG Nodes

At the end, the results on the GO tree need to be converted back to the state probabilities on the original GO DAG. We design an efficient algorithm to do so through the use of conditional transition probabilities on the GO tree. Define $c_{jk}(i)$ as the probability of GO tree node i being state k condition on all the observed data ($\boldsymbol{p}$) and its parent being in state j. Given $\boldsymbol{\theta}$ and for $i = 2, \ldots, N_T$, $c_{jk}(i)$s can be

computed from the upward probabilities as follows:

$$\begin{aligned} c_{jk}(i) &\equiv \mathcal{P}(S_i = k|\boldsymbol{p}, S_{\rho(i)} = j) \\ &= \mathcal{P}(S_i = k|\boldsymbol{p}_i, S_{\rho(i)} = j) \\ &= \frac{f(S_i = k, \boldsymbol{p}_i|S_{\rho(i)} = j)}{f(\boldsymbol{p}_i|S_{\rho(i)} = j)} \\ &= \frac{f(\boldsymbol{p}_i|S_i = k)\mathcal{P}(S_i = k|S_{\rho(i)} = j)}{f(\boldsymbol{p}_i|S_{\rho(i)} = j)} \\ &= \frac{q_{jk}\mathcal{P}(S_i = k|\boldsymbol{p}_i)f(\boldsymbol{p}_i)/\mathcal{P}(S_i = k)}{f(\boldsymbol{p}_i|S_{\rho(i)} = j)} \\ &= \frac{q_{jk}\tau_i(k)}{\tau_{\rho(i),i}(j)\mathcal{P}(S_i = k)}. \end{aligned} \tag{4.5}$$

To simplify the notation for our two-state GO tree, define $c_i \equiv c_{11}(i)$. This is because by logical restriction, $c_{00}(i) = 1$, and $c_{01}(i) = 0$. Furthermore, $c_{10}(i) = 1 - c_{11}(i)$, so c_i is sufficient for computation of all four conditional transition probabilities. Thus, from (4.5) and for $i = 2, \ldots, N_T$,

$$c_i = \frac{\omega\tau_i(1)}{\tau_{\rho(i),i}(1)\mathcal{P}(S_i = 1)}. \tag{4.6}$$

Finally, it is straightforward to show that $c_1 = \tau_1(1)$. This derivation has not been shown in literature before, but the result is very useful in applications.

Recall that the state of jth GO DAG node $S_j^* = \max\{S_k : S_k \in \mathcal{GT}_j\}$, i.e., the maximum of its comprising tree node states. Given $\boldsymbol{\theta}$, define $\text{PDE}_j = \mathcal{P}_{\boldsymbol{\theta}}(S_j^* = 1|\boldsymbol{p})$, the conditional probability that the jth GO DAG node is in state 1 (or, equivalently, that gene set $\mathcal{G}_j$ is DE) given all p-values corresponding to nodes of the HMT on the GO tree defined in Section 4.3.2. It is straightforward to use c_is to compute the PDE_js by using the GO tree structure and conditional independence of the states in the HMT. For example, in the toy example in Figure 4.1, original GO DAG node 2 is the union of tree nodes 2 and 4. Then the probability that DAG node 2 is in state 1

is the probability that either tree node 2 or 4 is in state 1. Note that S_2 and S_4 are independent given S_1 and $\boldsymbol{p}$. Furthermore, c_is are computed as in (4.6) and annotated in Figure 4.1d. Then the computation can be carried out as follows:

$$\begin{aligned}
\text{PDE}_2 &= \mathcal{P}(S_2^* = 1|\text{HMT}) \\
&= \mathcal{P}(S_2 = 1 \text{ or } S_4 = 1|\boldsymbol{p}) \\
&= \mathcal{P}(S_1 = 1|\boldsymbol{p})\mathcal{P}(S_2 = 1 \text{ or } S_4 = 1|S_1 = 1, \boldsymbol{p}) \\
&= \mathcal{P}(S_1 = 1|\boldsymbol{p})[1 - \mathcal{P}(S_2 = 0, S_4 = 0|S_1 = 1, \boldsymbol{p})] \\
&= \mathcal{P}(S_1 = 1|\boldsymbol{p})[1 - \mathcal{P}(S_2 = 0|S_1 = 1, \boldsymbol{p})\mathcal{P}(S_4 = 0|S_1 = 1, \boldsymbol{p})] \\
&= c_1[1 - (1 - c_2)(1 - c_3c_4)].
\end{aligned}$$

The second from the last step is due to the fact that S_2 and S_4 are independent given S_1 and $\boldsymbol{p}$. The PDEs of each GO DAG node can be carried out in similar way with tedious technical computations. We estimate $\boldsymbol{\theta}$ as $\hat{\boldsymbol{\theta}}$ as in Section 4.3.4, then compute the plug-in estimates of $\hat{c}_i$s and $\widehat{\text{PDE}}_j$s using $\hat{\boldsymbol{\theta}}$.

4.3.6 Rejection Region

By definition, $1-\text{PDE}_i = \mathcal{P}_{\boldsymbol{\theta}}(S_j^* = 0|\boldsymbol{p})$, which is closely related to the local index of significance defined by Sun and Cai (2009) in their work on testing HMM-dependent hypotheses. For any rejection index set R, a natural estimate for the FDR is

$$1 - \frac{1}{|R|}\sum_{i\in R}\widehat{\text{PDE}}_i, \tag{4.7}$$

i.e., $1 -$ the average of the PDE estimates for nodes in the rejection set. However, as noted by Goeman and Mansmann (2008) and Liang and Nettleton (2010), FDR may not be an appropriate quantity to control in a structured hypothesis testing problem like the GO DAG. Thus, we recommend selecting a subset of nodes with the highest

estimated PDE values with suggested threshold for significance of 0.95 or 0.99, for example.

4.4 A Data-Based Simulation Study

To simulate data that mimics nearly all aspects of real data , we used the simulation procedure proposed by Nettleton et al. (2008). This procedure not only preserves the marginal distribution of genes, but also keeps the correlations among genes largely intact. The dataset of B- and T-cell Acute Lymphocytic Leukemia (ALL) (Chiaretti et al., 2004, publicly available through Bioconductor ALL package at www.bioconductor.org) was used in the simulation as a population. The ALL dataset consists of gene expressions of 95 B-cell and 33 T-cell ALL patients measured by Affymetrix HGU95aV2 GeneChips. 8192 genes out of the total 12625 genes measured were mapped to one or more GO terms using the hug95av2 package version 2.0.1 from Bioconductor, and there were totally 2353 non-empty unique biological process GO terms to be investigated. Note that the electronic annotations (the annotations without the confirmations of human curators) were excluded to increase annotation reliability.

A list of DE genes was derived from the study of Liu et al. (2007), who compared their Domain-Enhanced Analysis method using Partial Least Squares with the Fisher's exact test method on the same ALL dataset and reported a list of the top ten DE gene sets between B- and T-cell patients for each method. We merged the two lists to form a list of 14 unique gene sets. The union of these 14 gene sets consisted of 845 genes out of the 8192 genes on the GeneChip that were mapped to GO terms. This set of 845 genes was used to simulate differential expression and will be referred to as the DE gene list.

Each of 20 simulated datasets was generated as follows: first, $2n$ and n patients were drawn randomly without-replacement from B- and T-cell populations, respectively; second, data from the 845 genes on the DE list of the latter half of the $2n$ B-cell patients were replaced with data from these 845 genes from the n T-cell patients. The first n of the B-cell patients were left intact. Then only the $2n$ B-cell patients were kept as our simulated data (n intact multivariate observations and n modified multivariate observations). The sample of intact observations was then compared to the sample of modified observations. A total of 1250 gene sets each contained at least some of the 845 genes on the DE list. These 1250 gene sets were DE by construction because the 845 DE-list genes of the first n B-cell patients came from the finite population of 95 B-cell patients, and the DE-list genes of the latter n B-cell patients came from the finite population of 33 T-cell patients. These two finite populations have different mean vectors, different gene-specific variances, different between gene correlations, etc. The rest of 1103 gene sets consist of genes from the same B-cell population, and thus their corresponding null hypotheses are true nulls. The sample size n was chosen to be 9 in our simulation study.

The p-values of the gene sets could be computed using any of the multivariate gene set testing methods mentioned in Section 4.1. We used the MRPP method of Mielke and Berry (2001) , which is a permutation test and free of distributional assumptions. The MRPP method permutes the treatment labels among the subjects and computes the sum of within-group inter-subject distances between gene set expression vectors for each permutation. Then the p-value is the standardized rank (scaled between 0 and 1) of the original sum of distances among all permutation sums. In our simulation, n modified B-cell subjects were compared to n intact B-cell subjects, and flipping the treatment label for each subject would result in a symmetric grouping that has the

same sum of within-group distances. Thus, it is necessary to consider only $\binom{18}{9}/2 = 24310$ of the permutations.

We compared our HMT method with the HMM method in Liang and Nettleton (2010), the one-step min-p procedure proposed by Westfall and Young (1993) and the bottom-up procedure, which is described in Section 4.1. The first two methods were applied to the tree-transformed GO DAG with a PDE or PPDE significance threshold of 0.95. The latter two methods were applied to the original GO DAG to control FWER at the 0.05 level. We recognize that these two error control strategies are not directly comparable. However, methods for controlling error rates other than FWER are not available for the min-p and bottom-up procedures.

For the min-p procedure, we computed p-values for each gene set for all 24310 permutations of the treatment labels as described in the previous paragraph. All the p-values were arranged in a 2353 by 24310 matrix. Then the minimum p-value across all gene sets for each permutation was obtained. These 24310 minimum p-values provide a reference distribution for the smallest p-value under the null hypothesis of no treatment difference. We can declare significant any p-value that is no larger than the 0.05 quantile of the reference distribution. We then enforce logical consistency in accordance with the DAG by adjusting results using the bottom-up strategy so that all ancestors of significant nodes will also be declared significant. We also considered the potentially more powerful step-down min-p method but obtained results identical to the one-step approach in all 20 simulated datasets; for detail of the step-down version of the min-p method, see Westfall and Young (1993).

We also considered other potentially useful methods in our simulation study, but all other methods were ultimately excluded. For example, a variant of the bottom-up procedure is to apply Holm's method to all the nodes and reject the ancestors of

rejected nodes. We call this procedure the *global-up* procedure. This variant does not tend to work well when the number of nodes is large, as in the case of a GO DAG. Because the threshold for significance controlling FWER at 0.05 level is smaller than the smallest p-value, this would lead to no rejections for all the simulated samples. The global-up procedure can have better performance than the bottom-up procedure when the graph size is small, but it is useless in our simulations. Another example is the top-down procedure, which is too computationally prohibitive to use for a large graph. Another option is Goeman's focus level method, but this approach depends heavily on the choice of a focus level that we have no basis for choosing.

As shown in Table 4.2, both FWER-controlling methods exhibited excellent performance with regard to type I error control. No type I error was made by either of the FWER-controlling methods across all 20 simulated datasets. The min-p procedure was superior to the bottom-up procedure with respect to power for all 20 simulated datasets. The HMT and HMM method exhibited far more power than either of the FWER-controlling methods, often identifying more than twice as many true positive results at the cost of very few additional false positives. Even for simulated dataset 12 where the HMT procedure arguably performed worst, we believe that many scientists would prefer the HMT results to those obtained using the procedures that control FWER. For this case, the HMT method found more than 2.8 times the number of truly differentially expressed categories that were identified by the better of the competing FWER procedures ($603 - 54 = 549$ versus 192). The cost for the additional 357 discoveries in this worst case was 54 type I errors. In all cases, our HMT method included all the discoveries that were made by the two FWER-controlling methods except one made by the min-p method in the 6th dataset and 25 by the min-p and bottom-up procedures in the 14th dataset.

Table 4.2: Number of rejections and false positives across 20 simulated datasets for the proposed HMT method, HMM method, bottom-up method and min-p method. R denotes # of rejections; V denotes # of false positives.

	HMT		HMM		bottom-up		min-p	
Dataset	R	V	R	V	R	V	R	V
1	543	4	495	0	135	0	189	0
2	509	5	428	1	161	0	195	0
3	467	5	343	0	180	0	212	0
4	573	7	436	3	167	0	188	0
5	516	0	397	0	161	0	166	0
6	459	18	361	10	148	0	204	0
7	358	3	340	4	148	0	176	0
8	358	8	360	9	148	0	159	0
9	567	25	466	11	182	0	218	0
10	661	32	585	24	161	0	185	0
11	318	1	336	2	127	0	130	0
12	603	54	498	32	182	0	192	0
13	260	0	260	0	148	0	170	0
14	338	0	403	0	179	0	200	0
15	462	26	384	6	182	0	197	0
16	684	55	562	31	171	0	190	0
17	381	6	364	6	187	0	207	0
18	611	30	478	16	133	0	169	0
19	377	0	274	0	182	0	196	0
20	395	18	346	3	158	0	191	0

Because different methods use different error rates, it is important to examine the trade-off between sensitivity and specificity in each case. To allow a fair comparison and further illustrate the advantage of the newly developed HMT method, we used receiver operating characteristic (ROC) curves in Figure 4.2 to compare the HMT method with the other three methods and a method based only on p-values. The latter method rejects the GO DAG nodes by their p-value in an ascending order without regard to graph structure.

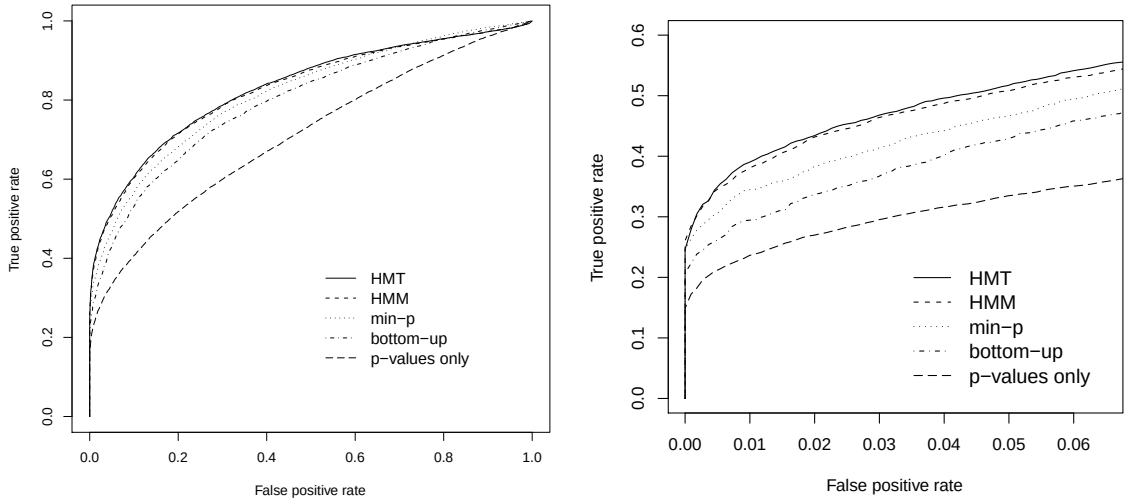
To draw the ROC curves, we arranged the GO DAG nodes in the order they would be rejected according to each procedure and then plotted the true positive rate versus false positive rate as the number of rejections increased. Assume that the GO DAG nodes are numbered in such way that an ancestor node always has a smaller index than any of its descendant nodes. For the min-p procedure, the rejections happen in the following order. We find the GO DAG node with the smallest p-value among the nodes that have not been rejected, and then this node and all its ancestor nodes are rejected in an increasing order of their indexes such that an ancestor node is always rejected before any of its descendant nodes is rejected. The above step is repeated until all the nodes are rejected. Rejecting an ancestor node before any of its descendant nodes is a sensible thing to do in the context of testing the null hypotheses in (4.1) on a GO DAG because a necessary condition for a descendant node to be DE is that all its ancestor nodes are DE. This ancestor-first rejection strategy is a good way to use the structural information of the GO DAG, and all methods except the p-values only procedure use this strategy. Note that the HMT and HMM methods use this strategy implicitly because an ancestor's PDE or PPDE is guaranteed to be larger than that of any of its descendants by the Markov model and the conversion processes in (4.2). It is easy to see that the global-up procedure will have exactly the same rejection order as the min-p procedure, and arguably this rejection order is the best one that can be achieved based solely on the p-values and structural relationship information of the original GO DAG.

It is clear from Figure 4.2 that the bottom-up procedure is better than the p-values only method because the p-values only method doesn't use GO DAG structural information at all. The min-p procedure is superior to the bottom-up procedure because it is not confined to testing only leaf nodes. The performance of HMT and HMM meth-

ods are close to each other with HMT slightly better. They both are superior to the min-p procedure because they more fully utilize the GO DAG structural information by modeling the whole GO DAG. Thus, the power advantage exhibited in our Table 4.2 simulation result was not simply a consequence of differing error control criteria. In related work under a hidden Markov model, Sun and Cai (2009) proposed a statistic, the local index of significance (LIS), that incorporates the underlying dependence relationship among null hypotheses, and showed that the procedure using the LIS is more powerful than the FDR control procedure that is based only on the p-values. The definition of PDE is closely related to the LIS, and thus, it is not surprising that the HMT method is superior to the p-values only method. It is somewhat remarkable that our method outperformed the min-p procedure, whose rejection order is optimal among all the procedures that use only p-values and structural information of the original DO DAG. Thus, by modeling the structural dependence among the null hypotheses, the HMT and HMM methods turn the restrictions on the GO DAG into information and are superior to methods that simply ignore the information or methods that passively obey the restrictions. In summary, our HMT method was better able to distinguish DE gene sets from equivalent expressed gene sets for all relevant significance thresholds.

4.5 Application and Discussion

Our HMT method was applied to a large-scale expression quantitative trait loci (eQTL) dataset collected by West et al. (2007). The dataset contains 211 recombinant inbred lines (RIL) of Arabidopsis thaliana, a model organism in plant genetics. Each RIL was measured on two biological replicates, and a total of 422 Affymetrix ATH1 GeneChips were used. Each GeneChip measures 22,810 genes of Arabidopsis



(a) ROC plot

(b) Left-lower part of the ROC plot

Figure 4.2: ROC curves for the HMT, HMM, min-p, bottom-up and p-values only methods.

thaliana. The microarray dataset can be obtained at http://elp.ucdavis.edu. Microarray measurements were normalized using the robust multichip average (RMA) method (Bolstad et al., 2003). The measurements of the two biological replicates were averaged to give a single transcript measurement per gene and RIL.

These 211 RILs are part of a population of 420 RILs that were genotyped by Loudet et al. (2002). The 420 RILs were the result of crossing between two genetically distant ecotypes, Bay-0 and Shahdara. A set of 38 physically anchored microsatellite markers was measured for each RIL; the genotype at each marker either comes from Bay-0 or Shahdara. The marker genotype data can be found at http://dbsgap.versailles.inra.fr/vnat/Documentation/33/DOC.html.

Using version 2.2.13 of the ath1121501.db Bioconductor package, 2031 unique nonempty GO terms from the biological process ontology were identified. The p-values

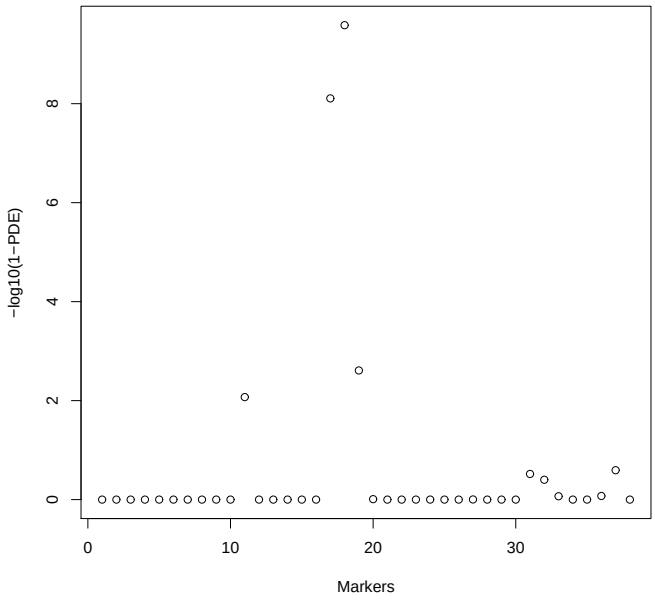


Figure 4.3: PDEs of GO term "GO:0006549" across markers.

for the gene sets corresponding to the GO tree nodes were computed using Goeman's Global Test method (Goeman et al., 2004). For each of the 38 markers, the HMT method was carried out to calculate the PDEs for the GO terms.

The results associated with Figure 4.3 illustrates why our HMT method is more powerful than the sequential FWER controlling top-down procedure. PDEs of GO term "GO:0006549", isoleucine metabolic process, were plotted against markers. It is evident that there is a eQTL for the gene set near marker 17 and 18. The larger p-value for the GO term at the two markers is 2.2e-11, which is remarkably small. On the other hand, one of its ancestor GO terms, "GO:0009082", has p-values of 0.30 and 0.12 at the two markers. If the top-down procedure were used, the highly significant GO term "GO:0006549" would never be tested even at FWER level 0.1.

Our HMT method has been shown to be superior in classifying DE and EE nodes

to other existing methods. We also suggest practitioners to use any FWER control method as a first step. If the FWER method declares that no gene set is DE, then we stop and reject nothing. Otherwise, our HMT method can be applied. This added step will provide weak control of FWER, i.e., control of FWER when all the null hypotheses are true. Note that none of the results in our paper would change with this modification.

In summary, our HMT method provides a more powerful and sensible solution to testing gene sets on the GO DAG over the existing sequential methods. The improved power comes from the method's ability to borrow information throughout the GO DAG structure. The HMT method is also more computationally efficient than the HMM method proposed by Liang and Nettleton (2010). Thus, the HMT method is both powerful in inference and efficient in computation and is well-suited for computationally intensive applications and practitioners with limited computation resources.

Bibliography

Allison, D. B., Cui, X., Page, G. P., and Sabripour, M. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, 7(1):55–65.

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al. (2000). Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Bolstad, B., Irizarry, R., Astrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185.

Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J., and Foa, R. (2004). Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103(7):2771–2778.

Crouse, M., Nowak, R., and Baraniuk, R. (1998). Wavelet-based statistical signal processing using hidden Markov models. *IEEE transactions on signal processing*, 46(4):886–902.

Dempster, A., Laird, N., Rubin, D., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38.

Durand, J., Goncalves, P., and Guedon, Y. (2004). Computational methods for hidden markov tree models-an application to wavelet trees. *IEEE Transactions on Signal Processing*, 52(9):2551–2560.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–105.

Goeman, J. and Buhlmann, P. (2007). Analyzing gene expression data in terms of gene sets: Methodological issues. *Bioinformatics*, 23(8):980.

Goeman, J. J. and Mansmann, U. (2008). Multiple testing on the directed acyclic graph of gene ontology. *Bioinformatics*, 24(4):537–544.

Goeman, J. J., van de Geer, S. A., de Kort, F., and van Houwelingen, H. C. (2004). A global test for groups of genes: Testing association with a clinical outcome. *Bioinformatics*, 20(1):93–99.

Granat, R. and Donnellan, A. (2002). A Hidden Markov Model Based Tool for Geophysical Data Exploration. *Pure and Applied Geophysics*, 159(10):2271–2283.

Khatri, P. and Draghici, S. (2005). Ontological analysis of gene expression data: Current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595.

Liang, K. and Nettleton, D. (2010). A hidden markov model approach to testing multiple hypotheses on a gene ontology graph. *Journal of the American Statistical Association*, To appear.

Liu, J., Hughes-Oliver, J. M., and Menius, A. J. (2007). Domain-enhanced analysis of microarray data using go annotations. *Bioinformatics*, 23(10):1225–1234.

Loudet, O., Chaillou, S., Camilleri, C., Bouchez, D., and Daniel-Vedele, F. (2002). Bay-0× Shahdara recombinant inbred line population: a powerful tool for the genetic dissection of complex traits in Arabidopsis. *Theoretical and Applied Genetics*, 104(6):1173–1184.

Mansmann, U. and Meister, R. (2005). Testing differential gene expression in functional groups. goeman's global test versus an ancova approach. *Methods of Information in Medicine*, 44(3):449–53.

Marcus, R., Eric, P., and Gabriel, K. (1976). On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3):655–660.

Mielke, P. and Berry, K. (2001). *Permutation Methods: A Distance Function Approach*. Springer.

Nettleton, D., Recknor, J., and Reecy, J. M. (2008). Identification of differentially expressed gene categories in microarray studies using nonparametric multivariate analysis. *Bioinformatics*, 24(2):192–201.

Ronen, O., Rohlicek, J., and Ostendorf, M. (1995). Parameter estimation of dependence tree models using the EM algorithm. *IEEE Signal Processing Letters*, 2(8):157–159.

Sun, W. and Cai, T. (2009). Large-scale multiple testing under dependence. *Journal of the Royal Statistical Society, Series B*, 71:393–424.

Tomfohr, J., Lu, J., and Kepler, T. B. (2005). Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, 6(1):225.

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282.

West, M., Kim, K., Kliebenstein, D., van Leeuwen, H., Michelmore, R., Doerge, R., and St Clair, D. (2007). Global eQTL mapping reveals the complex genetic architecture of transcript-level variation in Arabidopsis. *Genetics*, 175(3):1441.

Westfall, P. and Young, S. (1993). *Resampling-based Multiple Testing: Examples and Methods for p-value Adjustment*. Wiley-Interscience.

CHAPTER 5. Summary

5.1 Conclusion

This dissertation explored important issues of adaptive multiple testing problems. In Chapter 2, we showed that a class of dynamic adaptive procedures provides conservative point estimations for the proportion of true null hypotheses (π_0) and FDR. These procedures are truly adaptive procedures because of their ability to adapt to the data when estimating π_0. Thus, the dynamic adaptive procedures offer a solution to the problem of choosing the tuning parameters for adaptive procedures. In Chapter 3, we discussed important issues of gene set testing, which are commonly used in biological research, and the related multiple testing problems. We developed new methodology based on a hidden Markov model to test multiple gene sets of the Gene Ontology. Our method not only honors the logical relationships among the null hypotheses but also uses them to achieve more powerful results than other existing methods. In a sense, our method is able to adapt to dependences among null hypotheses to make better inference. In Chapter 4, we developed a more computationally efficient method to implement our hidden Markov methodology.

5.2 Future work

5.2.1 Dynamic Adaptive FDR Control Procedure

In Chapter 2 we showed that a class of dynamic adaptive procedures provides conservative point estimation for FDR. Although it has been shown in the literature that conservative FDR estimation is closely related to FDR control, our result does not directly translate to the control of FDR. Thus, we should try to prove that at least some subset of this class of procedures control FDR. Furthermore, the right-boundary procedure enjoys both finite sample and asymptotic properties, but the selection of optimal number of histogram bins is still unsettled.

5.2.2 Direct Inference on the GO DAG

The hidden Markov model was originally proposed on the GO DAG. We transformed the GO DAG into a GO tree in Chapters 3 and 4 mainly for computational reasons. As commented by one reviewer to our paper, it is somehow wasteful to not use the original GO DAG structure. The exact inference on the GO DAG is not computationally feasible, but some approximation may be attempted without the tree transformation. Furthermore, even the hidden Markov model on the original GO DAG has its drawback. In particular, we assumes the conditional independence of the p-values given their states. However, p-values of the gene sets that share genes are correlated even after conditioning on their corresponding states. That is, to more realistically model the p-values or some other statistics corresponding to the gene sets, we could drop the conditional independence assumption. It is always a challenge to strike a balance between the model complexity (accurate approximation to reality) and our ability to compute/estimate a model.