8-2009

A Hidden Markov Model Approach to Testing Multiple Hypotheses on a Gene Ontology Graph

Kun Liang
Iowa State University

Dan Nettleton
Iowa State University, dnett@iastate.edu

A Hidden Markov Model Approach to Testing Multiple Hypotheses on a Gene Ontology Graph

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.

Keywords
Bayesian data analysis, differential expression, directed acyclic graph, false discovery rate, gene set enrichment analysis, microarray, multiple testing, simultaneous inference

Disciplines
Statistics and Probability

Comments
This preprint was published as Kun Liang & Dan Nettleton, "A Hidden Markov Model Approach to Testing Multiple Hypotheses on a Tree-Transformed Gene Ontology Graph", *Journal of the American Statistical Association* (2010): 1444-1454, doi: 10.1198/jasa.2010.tm10195.

A Hidden Markov Model Approach to Testing Multiple Hypotheses on a Gene Ontology Graph

Kun Liang and Dan Nettleton

Department of Statistics

Iowa State University, Ames, IA 50011

email: liangkun@iastate.edu

August 1, 2009

Author's Footnote:

Kun Liang is Doctoral Candidate, Department of Statistics, Iowa State University, Ames, IA 50011 (email: liangkun@iastate.edu); and Dan Nettleton is Professor, Department of Statistics, Iowa State University, Ames, IA 50011 (email: dnett@iastate.edu). This work was supported by the National Science Foundation Grant 0714978.

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 enrichment analysis; Microarray; Multiple testing; Simultaneous inference.

1. INTRODUCTION

The initial analysis of many microarray experiments includes testing a null hypothesis of equivalent expression (EE) 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 G_i is a indicator matrix whose rows are a subset of the $P \times P$ identity matrix such that $G_i \boldsymbol{X}_{tu}$ 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 $G_i \boldsymbol{X}_{tu} \sim F_t^{(i)}$ for all $i = 1, \ldots, N$; $t = 1, \ldots, T$; and $u = 1, \ldots, n_t$. We consider the problem of testing

$$H_0^{(i)} : F_1^{(i)} = \cdots = F_T^{(i)} \tag{1}$$

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 genes 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.

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 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 (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 (PLAGE, Tomfohr, Lu, and Kepler 2005), and Domain Enhance Analysis (DEA, 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 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 rejecting 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 EE. 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. THE PROPOSED APPROACH

We begin by computing a single p-value for testing the null hypothesis in (1) separately for each node in the GO DAG. We then model the joint distribution of these p-values using a hidden Markov model (HMM). We treat the state of each null hypothesis (true or false) 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 DAG.

We do not claim that our proposed model for the states is the true data-generating model. 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. We use a fully Bayesian approach with Markov chain Monte Carlo (MCMC) to estimate the posterior dis-

tribution of the null hypotheses' states. The null hypotheses with high posterior probabilities of differential expression (PPDE) will be rejected. The rejected null hypotheses are guaranteed to be consistent with the logical constraints of the GO DAG.

3.1 A Hidden Markov Model for p-values on the GO DAG

For a gene set 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 indices of the parent nodes 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 gene set i that is computed by testing (1) using any test that produces a valid p-value. Let S_i be the state of gene set i where $S_i = 0$ if the ith gene set is EE and $S_i = 1$ if the ith gene set is DE. By the logical structure of the GO DAG, a node must be in state 0 if any of its parental nodes are in state 0. On the other hand, we assume that a node whose parents are all 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

$$\Pr(S_i = 0 | S_j = 0 \text{ for some } j \in \mathcal{P}_i) = 1 \quad \text{and} \quad \Pr(S_i = 1 | S_j = 1 \text{ for all } j \in \mathcal{P}_i) = \omega$$

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

$$p_i \sim \text{uniform}[0,1] \text{ if } S_i = 0 \text{ and } p_i \sim \text{beta}(\alpha, \beta) \text{ if } S_i = 1 \tag{2}$$

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 gene sets. 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 graph structure. It is hidden because the state of each node is unknown, and the Markov property follows as given its parents' 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.

3.2 Estimation

We are primarily interested in estimating the PPDE for each node. We utilize Metropolis-Hastings-in-Gibbs, a common MCMC strategy, to draw the posterior samples.

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 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 DAG. Chib (1996) showed that it is possible to sample the hidden states as a whole on a hidden Markov chain. Sampling states on a complex DAG like GO is generally hard. However, 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 approach is derived below.

For the moment we assume a tree structure, i.e., no node has more than one parent. We will show later how to transform a GO DAG to a tree.

Let P_{i1} denote the event that all the parent nodes of node i are in state 1, i.e.,

$$P_{i1} = \{S_j = 1 \ \forall \ j \in \mathcal{P}_i\}.$$

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, c_i's are the key quantities for sampling the states, and we now show how to compute it 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

$$A_{k0} = \frac{\pi(\boldsymbol{p}|S_i = k, C_{i0}, P_{i1}, \boldsymbol{\theta}) \Pr(S_i = k, C_{i0}|P_{i1}, \boldsymbol{\theta})}{\pi(\boldsymbol{p}|P_{i1}, \boldsymbol{\theta})} \tag{3}$$

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

$$\begin{aligned}
\left\{\prod_{j \in \mathcal{C}_i}(1 - c_j)\right\} c_i &= \Pr(C_{i0}|S_i = 1, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&= \Pr(C_{i0}|S_i = 1, P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
&= \Pr(C_{i0}, S_i = 1|P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) = A_{10} \qquad (4)
\end{aligned}$$

and

$$1 - c_i = \Pr(S_i = 0 | P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) = \Pr(S_i = 0, C_{i0} | \boldsymbol{p}, P_{i1}, \boldsymbol{\theta}) = A_{00}. \tag{5}$$

By equating A_{10}/A_{00} as given by (3) with A_{10}/A_{00} as given by (4) and (5) and then solving for c_i, we obtain the following expression for a node with at least one child.

$$\begin{aligned}
c_i &= \left\{ 1 + \frac{\pi(\boldsymbol{p} | S_i = 0, C_{i0}, P_{i1}, \boldsymbol{\theta}) \Pr(S_i = 0, C_{i0} | P_{i1}, \boldsymbol{\theta})}{\pi(\boldsymbol{p} | S_i = 1, C_{i0}, P_{i1}, \boldsymbol{\theta}) \Pr(S_i = 1, C_{i0} | P_{i1}, \boldsymbol{\theta})} \prod_{j \in \mathcal{C}_i} (1 - c_j) \right\}^{-1} \\
&= \left\{ 1 + \frac{\pi(p_i | S_i = 0, \boldsymbol{\theta}) \Pr(S_i = 0, C_{i0} | P_{i1}, \boldsymbol{\theta})}{\pi(p_i | S_i = 1, \boldsymbol{\theta}) \Pr(S_i = 1, C_{i0} | P_{i1}, \boldsymbol{\theta})} \prod_{j \in \mathcal{C}_i} (1 - c_j) \right\}^{-1} \\
&= \left\{ 1 + \frac{\pi(p_i | S_i = 0, \boldsymbol{\theta})(1 - \omega)}{\pi(p_i | S_i = 1, \boldsymbol{\theta}) \omega (1 - \omega)^{n_i}} \prod_{j \in \mathcal{C}_i} (1 - c_j) \right\}^{-1} \text{(where } n_i = \text{cardinality of } \mathcal{C}_i) \\
&= \left\{ 1 + \frac{1}{b(p_i | \alpha, \beta) \omega (1 - \omega)^{n_i - 1}} \prod_{j \in \mathcal{C}_i} (1 - c_j) \right\}^{-1} \\
&= \frac{b(p_i | \alpha, \beta) \omega (1 - \omega)^{n_i - 1}}{b(p_i | \alpha, \beta) \omega (1 - \omega)^{n_i - 1} + \prod_{j \in \mathcal{C}_i} (1 - c_j)}.
\end{aligned} \tag{6}$$

Now for a node i with no children,

$$\begin{aligned}
c_i &= \Pr(S_i = 1 | P_{i1}, \boldsymbol{p}, \boldsymbol{\theta}) = \Pr(S_i = 1 | P_{i1}, p_i, \boldsymbol{\theta}) \\
&= \frac{\pi(p_i | S_i = 1, \boldsymbol{\theta}) \Pr(S_i = 1 | P_{i1}, \boldsymbol{\theta})}{\pi(p_i | P_{i1}, \boldsymbol{\theta})} \\
&= \frac{\pi(p_i | S_i = 1, \boldsymbol{\theta}) \Pr(S_i = 1 | P_{i1}, \boldsymbol{\theta})}{\pi(p_i | S_i = 1, \boldsymbol{\theta}) \Pr(S_i = 1 | P_{i1}, \boldsymbol{\theta}) + \pi(p_i | S_i = 0, \boldsymbol{\theta}) \Pr(S_i = 0 | P_{i1}, \boldsymbol{\theta})} \\
&= \frac{b(p_i | \alpha, \beta) \omega}{b(p_i | \alpha, \beta) \omega + 1 - \omega}.
\end{aligned} \tag{7}$$

Now using (6) and (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 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 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 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 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 1d).

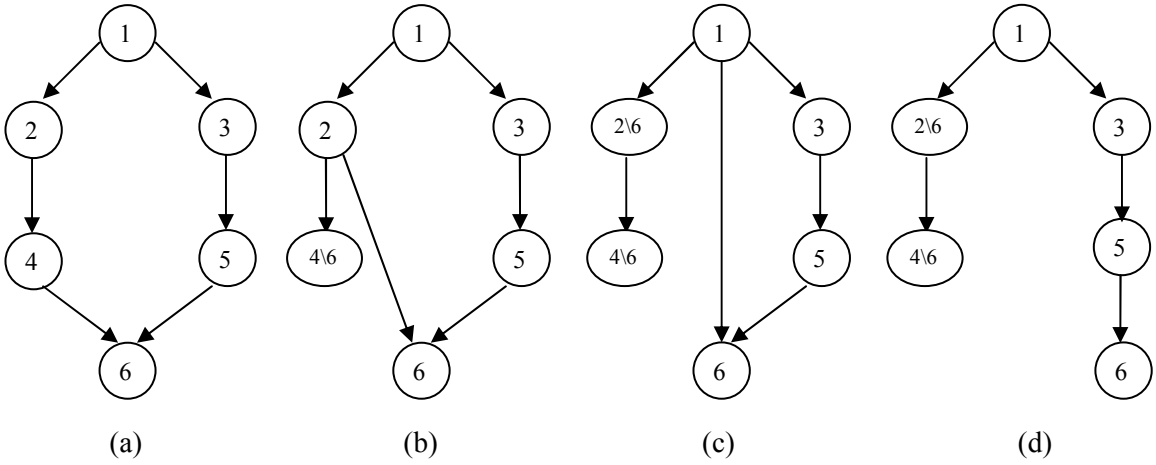


Figure 1: DAG to Tree: (a) Original DAG; (b) After remove genes in node 6 from node 4; (c) After remove genes in node 6 from node 2; (d) Tree after remove redundant edge from node 1 to node 6.

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 1a is a union of tree nodes 2\6 and 6 in Figure 1d. Although our MCMC algorithm 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.

3.4 Extensions

Though the above model fits most of cases well, 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

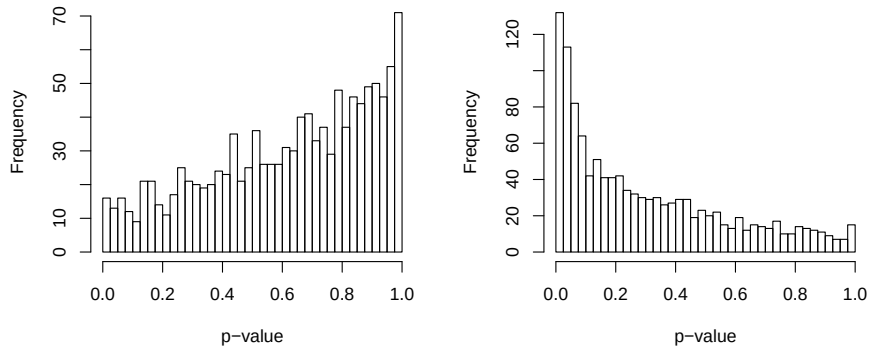
$$P(S_i = 1 | S_j = 1 \ \forall j \in \mathcal{P}_i) = \omega_i,$$

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 (6) and (7), ω will be replaced by ω_i's. 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.

For a second variation on our modeling strategy, consider the distribution of p-values from true null gene sets. 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 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 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



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

Figure 2: Histograms of true null p-values from two datasets simulated in Section 4.

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 EE 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 (2) changes to

$$p_i \sim \lambda + (1-\lambda)\text{beta}(\alpha_0, \beta_0) \quad \text{if } S_i = 0, \tag{8}$$

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 (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.

3.5 Estimation of False Discovery Rate

After the MCMC chains converge, a posterior sample of size B can be obtained. Then the PPDE for 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 node. For any rejection index set R, a natural estimate for the FDR is

$$1 - \frac{1}{|R|}\sum_{i \in R} \text{PPDE}_i, \tag{9}$$

i.e., 1 - average PPDE. However, rather than considering FDR, we suggest using a threshold on PPDE to choose the rejection set. 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 prone to error 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 easily arise in our case. Due to the logical constraints imposed by the GO graph structure, the higher-level nodes nearest the root have larger PPDE than the lower 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.

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 (ALL) 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 suffer from B-cell ALL while 33 have T-cell ALL. 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 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 ALL data to identify the most significant differentially expressed categories in the biological processes ontology for their DEA-PLS 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 ALL data. 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. 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). Other multivariate testing methods mentioned in Section 1 could be used to compute p-values as well.

We compared our method with the bottom-up method described in Section 2. 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 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 the Section 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 test will 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 on the original GO DAG after one controls for certain FDR on the corresponding tree structure.

Table 1: Number of rejections and false positives across 20 simulated datasets for the proposed HMM approach and the bottom-up approach.

	HMM		Bottom-Up	
Simulated Dataset	Number of Rejections	Number of False Positives	Number of Rejections	Number of False Positives
1	495	0	0	0
2	428	1	67	0
3	343	0	142	0
4	436	3	25	0
5	397	0	142	0
6	361	10	25	0
7	340	4	108	0
8	360	9	25	0
9	466	11	25	0
10	585	24	108	0
11	336	2	25	0
12	498	32	101	0
13	260	0	62	0
14	403	0	25	0
15	384	6	25	0
16	562	31	25	0
17	364	6	25	0
18	478	16	108	0
19	274	0	25	0
20	346	3	110	0

We chose the PPDE cutoff for our method to be 0.95. For the bottom-up method,

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 approach. The result is shown in Table 1. In all cases, our HMM method included all the discoveries that were made by the bottom-up method.

The HMM method exhibited far more power than the bottom-up method. In simulated dataset 1, the HMM method declared 495 GO terms to be DE and made no false positive discovery. In contrast, the bottom-up method made no discoveries. Even in dataset 16, where the HMM method made the second most false positive discoveries in all the simulated datasets, the HMM method was able to find 531 true DE terms while the bottom-up method only found 25 DE terms. These 25 DE terms were on a single chain in the GO graph so that the information derived from these discoveries would be very limited. Overall the HMM method exhibited far more power and produced results that would be more useful in practice.

To further illustrate the advantage of our HMM method, we drew the receiver operating characteristic (ROC) curve in Figure 3 to compare the HMM method with 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 part of the GO DAG structural information. The HMM method is superior to the bottom-up method because it further utilizes the GO DAG structural information by modeling the whole graph. Thus, the power advantage exhibited in our Table 1 simulation

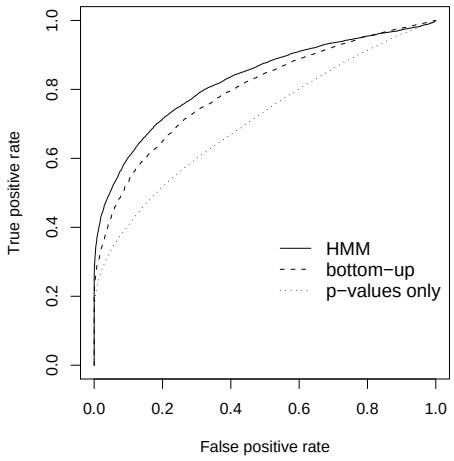


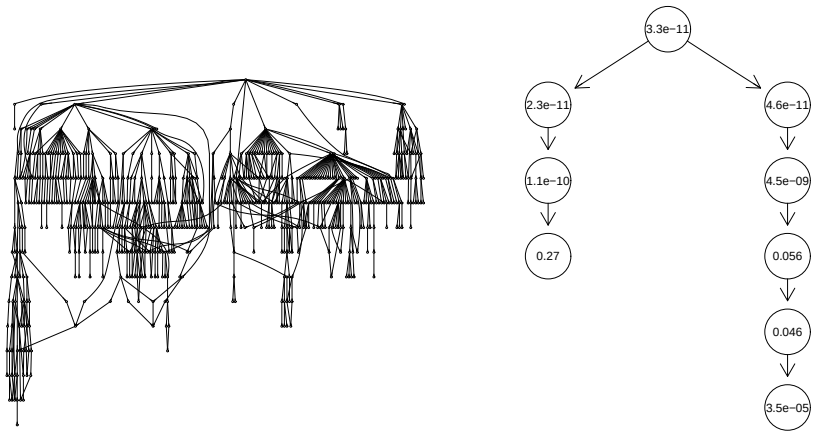
Figure 3: ROC curve for the HMM, bottom-up and p-values only methods.

result was not simply a consequence of differing error control criteria. Our HMM approach was better able to distinguish DE gene sets from EE gene sets for all relevant significance thresholds.

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 ALL and 25 acute myeloid leukemia (AML) patients. Using version 2.0.1 of the hu6800 Bioconductor package, we were able to identify 1577 unique non-empty GO terms from the molecular function ontology. The p-values were computed using Goeman's Global Test method (Goeman et al. 2004).

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.



(a) (b)

Figure 4: (a) DAG of all rejection in Section 5; (b) A subgraph of GO DAG with p-values annotated.

Figure 4a shows the DAG for all the rejections. Figure 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.

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.

APPENDIX: PROOF OF FULL CONDITIONAL DISTRIBUTION OF STATES

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*

$$\Pr(\boldsymbol{S}|\boldsymbol{p}, \boldsymbol{\theta}) = c_1^{S_1}(1-c_1)^{1-S_1} \prod_{i \in \mathcal{D}(\mathcal{T})} \left(\prod_{j \in \mathcal{C}_i} c_j^{S_j}(1-c_j)^{1-S_j} \right) \tag{10}$$

Proof. Note $\mathcal{P}_1 = \phi$ and $\Pr(P_{11}) = 1$. Thus (10) is equivalent to

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

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 (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.

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

$$
\begin{aligned}
& \Pr(S_r = 1, \boldsymbol{S}_1, \ldots, \boldsymbol{S}_n | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
= \; & \Pr(S_r = 1 | P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \Pr(\boldsymbol{S}_1, \ldots, \boldsymbol{S}_n | S_r = 1, P_{r1}, \boldsymbol{p}, \boldsymbol{\theta}) \\
= \; & c_r \prod_{i=1}^{n} \Pr(\boldsymbol{S}_i | S_r = 1, \boldsymbol{p}, \boldsymbol{\theta}) \qquad (S_r = 1 \text{ implies } P_{r1}) \\
= \; & c_r \prod_{i=1}^{n} \Pr(\boldsymbol{S}_i | P_{r_i 1}, \boldsymbol{p}, \boldsymbol{\theta}) \qquad (r \text{ is the only parent of } r_i's) \\
= \; & c_r \prod_{i=1}^{n} \left[c_{r_i}^{S_{r_i}} (1 - c_{r_i})^{1 - S_{r_i}} \prod_{j \in \mathcal{D}(\widetilde{\mathcal{T}}_i^c)} \left(\prod_{k \in \mathcal{C}_j} c_k^{S_k} (1 - c_k)^{1 - S_k} \right) \right] \\
= \; & c_r \prod_{i \in \mathcal{D}(\widetilde{\mathcal{T}})} \left(\prod_{j \in \mathcal{C}_i} c_j^{S_j} (1 - c_j)^{1 - S_j} \right)
\end{aligned}
$$

This establishes that (11) holds for a tree $\widetilde{\mathcal{T}}$ as long as (11) holds for all the child trees of $\widetilde{\mathcal{T}}$. Because (11) holds for a single node, it then holds for a two-level tree. By induction, the result follows. □

REFERENCES

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.

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.

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.

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

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