A TUTORIAL ON THE INHERITANCE PROCEDURE FOR MULTIPLE TESTING OF TREE-STRUCTURED HYPOTHESES

by

Dilinuer Kuerban
B.Sc. (Statistics), Southwestern University of Finance & Economics, 2011

a Project submitted in partial fulfillment
of the requirements for the degree of

Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2013

APPROVAL

Name:	Dilinuer Kuerban
Degree:	Master of Science
Title of Project:	A Tutorial on the Inheritance Procedure for Multiple Testing of Tree-structured Hypotheses

Examining Committee: Dr. Tim Swartz
Chair

Dr. Brad McNeney
Senior Supervisor
Associate Professor

Dr. Jinko Graham
Co-Supervisor
Associate Professor

Dr. Joan Hu
External Examiner
Professor

Date Approved: July 25, 2013

Abstract

In a candidate gene association study the goal is to find associations between a trait of interest and genetic variation at markers, such as single-nucleotide polymorphisms, or SNPs. SNPs are grouped within candidate genes thought to influence the trait. Such grouping imposes a tree structure on the hypotheses, with hypotheses about single-SNP associations nested within gene-based associations. In this project we give a tutorial on the inheritance procedure, a powerful new method for testing tree-structured hypotheses. We define sequentially rejective procedures and show that the inheritance procedure is a sequentially rejective procedure that strongly controls the family-wise error rate under so-called monotonicity and single step conditions. We also show how to further improve power by taking advantage of the logical implications among the nested hypotheses. The resulting testing strategy enables more powerful detection of gene- and SNP-based associations, while controlling the chance of incorrectly claiming that such associations exist.

Keywords: hierarchical testing; sequential testing; tree-structured hypotheses; family-wise error rate; power

To my family.

Acknowledgments

First of all, I would like to express my sincere gratitude to my co-supervisors Dr. Brad McNeney and Dr. Jinko Graham for their continuous dedication and guidance throughout my graduate studies at Simon Fraser University. I am really thankful for their patience and kind encouragement during my study in many ways.

I also would like to thank all the faculty members in the department of Statistics and Actuarial Science for teaching and and being supportive to us; and thank all the staffs for the help they offered.

I would like to give special thanks to Jean Shin for her numerous help and kindness since I arrived to SFU. I feel so lucky to be assigned to her by the department buddy program. Thanks to my fellow graduate students for their friendship and help, especially to Maria, Rachel, Zhenhua, Lu, Tianyu, Yuanyu, Qian, Kasra, Megan and many others. I really enjoyed two years of pursuing graduate study with them. Moreover, I would like to send my deep thank to my friend Akbar and some other friends in residence for their accompany and support when I need it most.

Finally, I want to thank to my beloved family members for their love and support all the way.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This project is a tutorial on the inheritance procedure for multiple testing of tree-structured hypotheses (Goeman and Finos, 2012). In this introduction, we will establish notation for the statistical model and hypotheses to be tested, and review the basics of multiple testing. We will also describe the type of multiple testing problem that motivated this project. We then define tree-structured hypotheses and give an example of a tree of hypotheses to illustrate ideas. The trees of hypotheses of interest to us are comprised of nested hypotheses which induce so-called logical implications. We define logical implications and provide an example. We conclude with an outline of the remainder of the project.

We begin by establishing notation for the statistical model that we are considering, the collection of null hypotheses to be tested, and the sets of true and false null hypotheses. Throughout we will refer to the following example linear regression model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{1.1}$$

where Y is the response; β_k for $k = 0, \ldots, 3$ are regression parameters; X_k for $k = 1, 2, 3$ are covariates; and ϵ is a random error term. To simplify the exposition, we will assume that the error is from a known distribution; e.g., a Normal distribution with mean zero and known variance. Let $\{P_M : M \in \mathbb{M}\}$ be the statistical model indexed by a set $\mathbb{M}$. We will refer to the M as parameters and $\mathbb{M}$ as the parameter space. For the example, in equation (1.1) the parameter space is $\mathbb{M} = \{(\beta_0, \beta_1, \beta_2, \beta_3) \in \mathbb{R}^4\}$. Let H denote a null hypothesis and $\mathcal{H}$ denote the collection of all null hypotheses of interest. An example null hypothesis is $H_1 : \beta_1 = 0$. The collection of all null hypotheses of interest might be

$$H_1 : \beta_1 = 0; \quad H_2 : \beta_2 = 0; \quad H_3 : \beta_3 = 0$$

along with their intersections

$$H_{12} : \beta_1 = \beta_2 = 0; \quad H_{13} : \beta_1 = \beta_3 = 0; \quad H_{23} : \beta_2 = \beta_3 = 0; \quad H_{123} : \beta_1 = \beta_2 = \beta_3 = 0$$

Each null hypothesis corresponds to a subset of the parameter space $\mathbb{M}$. For example, the hypothesis $H_1 : \beta_1 = 0$ is the subset $\{(\beta_0, \beta_1, \beta_2, \beta_3) \in \mathbb{R}^4 : \beta_1 = 0\}$. Within $\mathcal{H}$, some null hypotheses are true and some are false, depending on the true probability measure underlying the data generating process. Let $\mathcal{T}(M)$ denote the collection of true null hypotheses under the parameter M and $\mathcal{F}(M) = \mathcal{T}(M)^C$ denote the collection of false null hypotheses under M. To see more precisely the dependence of $\mathcal{T}(M)$ on the parameter M, we can proceed as follows. If M is the parameter, a true null hypothesis H under M is one that contains M. For example, if the parameter in the example is $M = (1, 1, 0, 0)$, then the true null hypotheses under M are $H_2 : \beta_2 = 0$, $H_3 : \beta_3 = 0$ and $H_{23} : \beta_2 = \beta_3 = 0$. These are the hypotheses that include the point $\beta_2 = \beta_3 = 0$. We can therefore express $\mathcal{T}(M)$ as the set of all hypotheses that include M, or $\mathcal{T}(M) = \{H \in \mathcal{H} : M \in H\}$.

The project relies on standard notation and terminology for describing testing problems. For testing a single hypothesis H, a type I error is the event that we reject H given that it is a true null, and the type I error rate is the probability of this event. A type II error is the event that we do not reject H given that it is a false null and the type II error rate is the probability of this event. Power is the probability of rejecting H when it is false, which is one minus the type II error rate.

A multiple testing procedure is one that tests the collection or *family* $\mathcal{H}$ of null hypotheses of interest. For a multiple testing procedure, a type I error is the event that we reject any of the true null hypotheses in $\mathcal{T}(M)$, and the probability of this event is the family-wise error rate (FWER). Throughout this project we will study multiple testing procedures that control FWER in the strong sense (Westfall and Young, 1993). Strong control of the FWER at level α means that the probability of a type I error is less than or equal to α for all possible $\mathcal{T}(M)$. By contrast, weak control means that the probability of type I error is controlled for the complete null, in which all $H \in \mathcal{H}$ are true. We will not discuss weak control any further. There are several possible definitions of the power of a multiple testing procedure (e.g., Dudoit and van der Laan, 2007, p. 23). For this project, we will use the term "powerful" to mean that a multiple testing procedure is better than others at detecting false null hypotheses.

The type of multiple testing problem that motivated this project is the candidate gene association study. A prototype study has the following elements. The research focus is a trait Y. This could be a binary indicator of disease status, or a quantitative measure such as blood pressure or LDL cholesterol. Researchers have identified a list of genes thought to influence the trait. These are the candidate genes. Within each gene, genetic variation may be measured at markers known as single nucleotide polymorphisms, or SNPs. For this project, information on the k^{th} SNP will be summarized as a predictor variable X_k that takes values 0, 1 or 2 for the number of copies of an index variant, or allele, at the SNP. The goal of a candidate gene association study is to find SNPs that are associated with the trait. There can be hundreds of SNPs tested.

In multiple testing problems such as a candidate gene study, the hypotheses have a tree structure. The genes provide a natural top-level grouping of the SNP predictor variables. Thus, we start looking for associations at the gene level, and then proceed to the SNP level for genes that show an association with the trait. The genes themselves may be grouped according to the network of genes, or pathway, that they belong to, but we will not discuss such a higher-level grouping in this project.

Not only does a tree structure provide a natural way to organize the hypotheses but, as argued by Meinshausen (2008), one should group hypotheses in order to improve the power to detect associations. Meinshausen's claim is illustrated with a multiple testing problem that involves a single predictor (e.g., SNP) that is associated with a trait, and is highly correlated with several other predictors (e.g., other SNPs) that comprise a group (e.g., a gene). In a genetic association study, alleles or variants at SNPs within the same gene are often correlated in the population, or are in linkage disequilibrium (LD). LD can arise from recent population admixture, by chance or by natural selection (e.g., Falconer and Mackay, 1996, p. 16). The dependence among predictors yields unstable estimates in the joint analysis, to the point where no single predictor can be detected to be associated. However, the group as a whole *is* associated with the trait. Hence we choose to test the group first, so that we at least detect the group association.

To illustrate a tree of hypotheses, we present the "toy example" from Goeman and Finos (2012) in Figure 1.1. We refer to this as Example 1. Testing on this tree structure starts with a test of the hypothesis at the root node N. If this hypothesis is rejected, we proceed to test the hypotheses at the child nodes N_1 and N_2 of N. There is no order associated with the testing of child nodes because they are tested simultaneously. Finally, if the hypothesis at node N_2 is rejected, we test the hypotheses at its child nodes N_{21} and N_{22}.

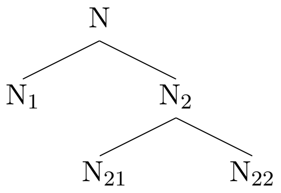


Figure 1.1: Example tree-structured hypotheses

To make the Example 1 more concrete, suppose the hypotheses relate to testing for an association between a quantitative trait and three SNPs with the linear model given in equation (1.1). Suppose the first SNP is from one candidate gene and the second and third SNPs are from another. The hypotheses to be tested are represented as nodes on the tree. The top, or root node is the null hypothesis of no association between the SNPs and the trait; i.e., N is $H_{123} : \beta_1 = \beta_2 = \beta_3 = 0$. The nodes at the next level are the null hypotheses of no association between the trait and SNPs in the first and second genes, respectively; i.e., N_1 is $H_1 : \beta_1 = 0$ and N_2 is $H_{23} : \beta_2 = \beta_3 = 0$. The nodes in the bottom level are the null hypotheses of no trait association with SNPs in the second gene; i.e., N_{21} is $H_2 : \beta_2 = 0$ and N_{22} is $H_3 : \beta_3 = 0$. The leaves of the tree are the single-SNP hypotheses N_1, N_{21} and N_{22}.

In general, let $\mathcal{L}$ denote the leaf nodes of the tree and $\mathcal{L}_H$ denote the set of descendant leaves of node H. We use common tree-structure terminology to describe the relationships between nodes of the tree, such as siblings, offspring, descendants, and ancestors. Examples of these relationships in the diagram of Figure 1.1 are as follows. Nodes N_{21} and N_{22} are siblings, and are the offspring of node N_2. Nodes N_1, N_2, N_{21} and N_{22} are the descendants of the root node N. The ancestors of node N_{21} are N_2 and N.

We are interested in tree-structured testing problems in which the hypotheses on the tree are nested, creating logical implications. Candidate gene studies provide an example of nested hypotheses, with hypotheses about single-SNP association nested within those about gene-based association. The logical implication between hypotheses induced by such nesting is that falseness of the hypothesis of no association between the trait and any of the SNPs within a gene implies falseness of at least one of the single-SNP hypotheses. For example, in Example 1 a logical implication exists between the gene-based hypothesis $H_{23} : \beta_2 = \beta_3 = 0$ (node N_2 in Figure 1) and the single-SNP hypotheses $H_2 : \beta_2 = 0$ and $H_3 : \beta_3 = 0$ (nodes

N_{21} and N_{22}, respectively, in Figure 1). Since H_{23} is the intersection of H_2 and H_3, falseness of H_{23} implies falseness of at least one of H_2 or H_3. As noted by Goeman and Finos (2012), in some situations (such as gene-set testing) hypotheses may follow a tree structure, but logical implications need not hold. We therefore distinguish between tree-structured hypotheses with and without logical implications.

The goal of the project is to explain the inheritance procedure (Goeman and Finos, 2012) for testing tree structured hypotheses with logical implications. The inheritance procedure is first developed to exploit the tree structure, but not logical implications. This basic inheritance procedure is then extended to exploit logical implications. Both formulations of the inheritance procedure, basic and extended, are designed to be sequentially rejective procedures in the sense of Goeman and Solari (2010). To prove that the two forms of the inheritance procedure control the FWER, Goeman and Finos (2012) show that they satisfy the conditions under which a sequentially rejective procedure controls the FWER.

An overview of the remainder of this project is as follows. We describe sequentially rejective procedures first (Chapter 2), and then the basic and extended inheritance procedures (Chapter 3) as special cases.

Chapter 2

Sequentially Rejective Procedures

In this chapter we define sequentially rejective multiple testing procedures (Goeman and Solari, 2010) and sufficient conditions under which they strongly control the FWER. Sequentially rejective procedures that satisfy two conditions, the monotonicity and single-step conditions, guarantee strong control of the FWER. The inheritance procedure is a sequentially rejective procedure. In particular, it is a so-called Bonferroni-Shaffer procedure. To facilitate understanding of the inheritance procedure, we next discuss the monotonicity and single-step conditions, both in the general case and in the special case of a Bonferroni-Shaffer procedure. The exposition is focused on an intuitive understanding of how the monotonicity and single-step conditions guarantee strong control of the FWER.

2.1 Definitions and Notation

The purpose of this section is to define sequentially rejective procedures and the notation used to describe them. We start with general definitions and notation, and then specialize to Bonferroni-Shaffer methods, which compare raw p-values to multiplicity-adjusted levels of tests of single hypotheses.

A sequentially rejective procedure is defined to be a sequence of single-step procedures. A single-step procedure is a multiple testing procedure that tests all currently-unrejected hypotheses simultaneously. The sequential procedure is initialized by declaring all hypotheses to be unrejected. At each step, a single-step procedure is used to test the currently-unrejected hypotheses, with a rejection rule that can depend on the set of previously-rejected hypotheses. The sequential procedure is terminated when the single-step procedure does not reject any further hypotheses.

Goeman and Solari define the following general notation to describe sequentially rejective procedures more precisely. Recall the notation $\mathcal{H}$ for the collection of all null hypotheses of interest. For a set $\mathcal{R} \subseteq \mathcal{H}$, write $\mathcal{R}^C$ for its complement within $\mathcal{H}$. Let $\mathcal{R}_i$ denote the rejected set of hypotheses after step i of the sequentially rejective procedure. Let $\mathcal{N}(\mathcal{R}_i)$ denote the set of hypotheses that are rejected by the single-step procedure given that the hypotheses in $\mathcal{R}_i$ have already been rejected. The function $\mathcal{N}(\cdot)$ is called the successor function. With this notation, a sequentially rejective procedure starts with

$$\mathcal{R}_0 = \varnothing$$

and iterates on

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{N}(\mathcal{R}_i)$$

until there are no more rejections. Note that the successor function is only used to add rejected hypotheses to the set already rejected in $\mathcal{R}_i$. There is no need for the successor function to return hypotheses already in $\mathcal{R}_i$. We may therefore assume that $\mathcal{N}(\mathcal{R}_i) \subseteq \mathcal{R}_i^C$. Goeman and Solari do not explicitly make this assumption part of their general definition of the successor function, but it seems to underlie much of their thinking, and it simplifies the exposition.

An important class of multiple testing procedures are the Bonferroni-Shaffer methods (Goeman and Solari, 2010) that compare raw p-values to multiplicity-adjusted levels of tests of single hypotheses. A Bonferroni-Shaffer sequentially rejective procedure is defined by its successor function, as follows. Let $\{p_H\}_{\mathcal{H}}$ denote the raw p-values for each hypothesis H in $\mathcal{H}$. The raw p-values have the property that for every parameter value $M \in \mathbb{M}$ and $H \in \mathcal{T}(M)$, $P_M(p_H \leq \alpha) \leq \alpha$. In a Bonferroni-Shaffer approach, the single-step procedure rejects hypotheses H with raw p-value, p_H, less than a multiplicity-adjusted level, $\alpha_H(\mathcal{R})$, of the test of the single hypothesis H, given that the hypotheses in $\mathcal{R}$ have already been rejected. That is, the successor function $\mathcal{N}(\mathcal{R})$ for a Bonferroni-Shaffer procedure is

$$\mathcal{N}(\mathcal{R}) = \{H \in \mathcal{R}^C : p_H \leq \alpha_H(\mathcal{R})\}.$$

2.2 Sufficient Conditions for Strong Control of the FWER

We now present the single-step and monotonicity conditions under which a sequentially rejective procedure strongly controls FWER. Monotonicity is used by Goeman and Solari

(2010) to reduce the problem of familywise error control of the sequentially rejective procedure to one of familywise error control of the single-step procedure in a particular scenario they call the "critical case". The single-step condition is familywise error control of the single-step procedure in this critical case. The two conditions are first stated in full generality, and then specialized to the case of Bonferroni-Shaffer methods. In this section we highlight the intuition for why the monotonicity and single-step conditions are sufficient to control FWER.

2.2.1 The Monotonicity Condition

The monotonicity condition specifies that, for already-rejected sets of hypotheses $\mathcal{R}$ and $\mathcal{S}$ in $\mathcal{H}$, such that $\mathcal{R} \subseteq \mathcal{S}$, the new rejected hypotheses of a procedure provided by the "successor function" $\mathcal{N}(\cdot)$ should follow:

$$\mathcal{N}(\mathcal{R}) \subseteq \mathcal{N}(\mathcal{S}) \cup \mathcal{S}. \tag{2.1}$$

We provide two alternate expressions for the monotonicity condition, one that justifies the name, and another that will be used to simplify the condition for Bonferroni-Shaffer procedures. Under the monotonicity condition, rejection sets grow monotonically as more hypotheses are rejected, which can be seen as follows. Since $\mathcal{R} \subseteq \mathcal{S}$ we may rewrite equation (2.1) as

$$\mathcal{N}(\mathcal{R}) \cup \mathcal{R} \subseteq \mathcal{N}(\mathcal{S}) \cup \mathcal{S}. \tag{2.2}$$

Thus, whenever $\mathcal{R} \subseteq \mathcal{S}$, $\mathcal{N}(\mathcal{R}) \cup \mathcal{R}$, the rejection set after applying the single-step procedure to $\mathcal{R}$, is contained in $\mathcal{N}(\mathcal{S}) \cup \mathcal{S}$, the rejection set after applying the single-step procedure to $\mathcal{S}$. An alternate form of the monotonicity condition (2.1) to be used later is

$$\mathcal{N}(\mathcal{R}) \cap \mathcal{S}^C \subseteq \mathcal{N}(\mathcal{S}), \tag{2.3}$$

which follows from simple set manipulations.

We now show how the monotonicity condition reduces the problem of familywise error control of the sequentially rejective procedure to one of familywise error control of the single-step procedure in a "critical case". For a given parameter M, let the critical case be that all of the false null hypotheses in $\mathcal{F}(M)$ and none of the true null hypotheses in $\mathcal{T}(M)$ have been rejected. In the critical case, the single-step procedure makes no false rejections if and only if the newly rejected set $\mathcal{N}(\mathcal{F}(M))$ is empty. Under monotonicity, $\mathcal{N}(\mathcal{F}(M)) = \varnothing$

implies that the sequentially rejective procedure makes no false rejections. To see why, suppose $\mathcal{N}(\mathcal{F}(M)) = \varnothing$ and apply the monotonicity condition (2.2) to the initial rejection set $\mathcal{R}_0$. By definition $\mathcal{R}_0 = \varnothing$ and so $\mathcal{R}_0 \subseteq \mathcal{F}(M)$. Hence,

$$\mathcal{R}_1 = \mathcal{N}(\mathcal{R}_0) \cup \mathcal{R}_0 \subseteq \mathcal{N}(\mathcal{F}(M)) \cup \mathcal{F}(M) = \mathcal{F}(M); \quad \text{i.e., } \mathcal{R}_1 \subseteq \mathcal{F}(M)$$

Next, since $\mathcal{R}_1 \subseteq \mathcal{F}(M)$, we may apply the monotonicity condition (2.2) again to get

$$\mathcal{R}_2 = \mathcal{N}(\mathcal{R}_1) \cup \mathcal{R}_1 \subseteq \mathcal{N}(\mathcal{F}(M)) \cup \mathcal{F}(M) = \mathcal{F}(M); \quad \text{i.e., } \mathcal{R}_2 \subseteq \mathcal{F}(M)$$

and so on for $\mathcal{R}_3, \ldots$. Thus, $\mathcal{N}(\mathcal{F}(M)) = \varnothing$ implies that the entire sequence of rejection sets for the sequentially rejective procedure are subsets of the set of false nulls. Thus, the sequentially rejective procedure makes no false rejections, as claimed. The key implication of monotonicity is that the event of no false rejections of the single-step procedure in the critical case implies the event of no false rejections of the sequentially rejective procedure. This implication allows us to control the family-wise error rate of the sequentially rejective procedure (i.e., the probability that it has no false rejections) by controlling the probability of no false rejections of the single-step procedure in the critical case. We discuss these ideas more fully in the Section 2.2.2.

For a Bonferroni-Shaffer procedure, Goeman and Solari claim that the monotonicity condition becomes

$$\alpha_H(\mathcal{R}) \leq \alpha_H(\mathcal{S}) \quad \forall H \in \mathcal{S}^C \tag{2.4}$$

where $\alpha(\mathcal{R})$ is the multiplicity-adjusted level of the test of a single hypothesis H, given that the hypotheses in $\mathcal{R}$ have already been rejected. To see why, we start with the definition of the successor function of a Bonferroni-Shaffer procedure

$$\mathcal{N}(\mathcal{R}) = \{H \in \mathcal{R}^C : p_H \leq \alpha_H(\mathcal{R})\}, \tag{2.5}$$

and the monotonicity condition from equation (2.3). Note that both sides of equation (2.3) are subsets of $\mathcal{S}^C$, and so the condition can be re-written as

$$\forall H \in \mathcal{S}^C, \;\; H \in \mathcal{N}(\mathcal{R}) \Rightarrow H \in \mathcal{N}(\mathcal{S}). \tag{2.6}$$

From the definition of the successor function $\mathcal{N}(\mathcal{R})$ in equation (2.5), H is in the newly rejected set $\mathcal{N}(\mathcal{R})$ if and only if $p_H \leq \alpha_H(\mathcal{R})$ and H is in the newly rejected set $\mathcal{N}(\mathcal{S})$ if and only if $p_H \leq \alpha_H(\mathcal{S})$. Thus the monotonicity condition in equation (2.6) becomes

$$\forall H \in \mathcal{S}^C, \;\; p_H \leq \alpha_H(\mathcal{R}) \Rightarrow p_H \leq \alpha_H(\mathcal{S}),$$

which is true if and only if $\alpha_H(\mathcal{R}) \leq \alpha_H(\mathcal{S})$ for all $H \in \mathcal{S}^C$; i.e., if and only if equation (2.4) holds. In Chapter 3 we show how the inheritance procedure satisfies the monotonicity condition (2.4).

To summarize: The monotonicity condition reduces the problem of familywise error control of the sequentially rejective procedure to one of familywise error control of the single-step procedure in the critical case when all of the false null hypotheses in $\mathcal{F}(M)$ have already been rejected. In particular, under monotonicity, no false rejections of the single-step procedure in the critical case implies no false rejections of the sequentially rejective procedure. For Bonferroni-Shaffer methods, the monotonicity condition simplifies to

$$\alpha_H(\mathcal{R}) \leq \alpha_H(\mathcal{S}) \quad \forall H \in \mathcal{S}^C \quad \text{for any sets of rejected hypotheses } \mathcal{R} \text{ and } \mathcal{S}, \text{ such that } \mathcal{R} \subseteq \mathcal{S}. \tag{2.4}$$

2.2.2 The Single-step Condition

The single-step condition is the control of familywise error in the critical case where all false null hypotheses in $\mathcal{F}(M)$ have been rejected and none of the true null hypotheses in $\mathcal{T}(M)$ have been rejected. Specifically, the condition is that for all parameter values M,

$$P_M(\mathcal{N}(\mathcal{F}(M)) = \varnothing) \geq 1 - \alpha \tag{2.7}$$

where α is the level at which to control the FWER. To show that a sequentially rejective procedure that obeys the single-step and monotonicity conditions also controls the FWER, we will use the form (2.7) of the single-step condition. To simplify the single-step condition for Bonferroni-Shaffer procedures, we will use the complementary form:

$$P_M(\mathcal{N}(\mathcal{F}(M)) \neq \varnothing) \leq \alpha \qquad \text{for every } M \in \mathbb{M}. \tag{2.8}$$

The formulations (2.7) and (2.8) differ from the single-step condition given by Goeman and Solari (2010), which is

$$P_M(\mathcal{N}(\mathcal{F}(M)) \subseteq \mathcal{F}(M)) \geq 1 - \alpha. \tag{2.9}$$

However, equivalence of our formulations (2.7)/(2.8), and Goeman and Solari's (2.9) follows from equivalence of the events

$$\{\mathcal{N}(\mathcal{F}(M)) \subseteq \mathcal{F}(M)\},$$

and

$$\{\mathcal{N}(\mathcal{F}(M)) = \varnothing\}.$$

This equivalence arises because we have assumed that a set of newly-rejected hypotheses $\mathcal{N}(\mathcal{R})$ is in the complement of the already-rejected hypotheses $\mathcal{R}$ and, hence, $\mathcal{N}(\mathcal{R})$ can only be a subset of $\mathcal{R}$ if it is empty.

We now discuss why a sequentially rejective procedure that obeys the single-step and monotonicity conditions controls the FWER. Let $\mathcal{R}_\infty$ be the totality of rejected hypotheses from the sequentially rejective procedure (i.e. the final rejection set). The event that the procedure does not make a familywise error is equivalent to the event $\{\mathcal{R}_\infty \subseteq \mathcal{F}(M)\}$ that the final rejection set is a subset of the set of false null hypotheses $\mathcal{F}(M)$. For a procedure that obeys the monotonicity condition, the event $\{\mathcal{R}_\infty \subseteq \mathcal{F}(M)\}$ is implied by the event $\{\mathcal{N}(\mathcal{F}(M)) = \varnothing\}$ of no false rejections in the critical case of all false nulls having already been rejected; i.e.,

$$\{\mathcal{N}(\mathcal{F}(M)) = \varnothing\} \Rightarrow \{\mathcal{R}_\infty \subseteq \mathcal{F}(M)\}. \tag{2.10}$$

Therefore, combining the monotonicity condition (2.10) and the single-step condition (2.7) gives control of the probability of no familywise error:

$$P_M(\mathcal{R}_\infty \subseteq \mathcal{F}(M)) \geq P_M(\mathcal{N}(\mathcal{F}(M)) = \varnothing) \geq 1 - \alpha.$$

Hence a sequentially rejective procedure that obeys the single-step and monotonicity conditions controls the FWER.

For Bonferroni-Shaffer procedures, the single-step condition (2.8) can be specialized to the following:

$$P_M\left(\bigcup_{H \in \mathcal{T}(M)} \{p_H \leq \alpha_H(\mathcal{F}(M))\}\right) \leq \alpha \qquad \text{for every } M \in \mathbb{M}. \tag{2.11}$$

Equivalence of conditions (2.11) and (2.8) amounts to equivalence of the events

$$\left\{\bigcup_{H \in \mathcal{T}(M)} \{p_H \leq \alpha_H(\mathcal{F}(M))\}\right\}$$

and

$$\{\mathcal{N}(\mathcal{F}(M)) \neq \varnothing\}.$$

Recall that $\mathcal{N}(\mathcal{F}(M))$ is the set of newly-rejected hypotheses given that all false null hypotheses, $\mathcal{F}(M)$, have previously been rejected. Therefore, the event $\{\mathcal{N}(\mathcal{F}(M)) \neq \varnothing\}$ is that at least one of the true null hypotheses in $\mathcal{T}(M)$ is rejected in the new step. For a

Bonferroni-Shaffer method, this means that one or more of the p-values for testing hypotheses in $\mathcal{T}(M)$ was less than its corresponding α-value. Thus, the event $\{\mathcal{N}(\mathcal{F}(M)) \neq \varnothing\}$ is the union over $H \in \mathcal{T}(M)$ of the events $\{p_H \leq \alpha_H(\mathcal{F}(M))\}$; i.e.,

$$\{\mathcal{N}(\mathcal{F}(M)) \neq \varnothing\} = \left\{ \bigcup_{H \in \mathcal{T}(M)} \{p_H \leq \alpha_H(\mathcal{F}(M))\} \right\},$$

as claimed. It follows that the general single-step condition (2.8) is equivalent to the condition (2.11) for Bonferroni-Shaffer procedures.

We now discuss the “Bonferroni-Shaffer inequality” (Goeman and Solari, 2010), which can be used to create Bonferroni-Shaffer procedures that satisfy the single-step condition. Our particular interest is in a special case of the Bonferroni-Shaffer inequality that is relevant to our discussion of the inheritance procedure in Chapter 3:

$$P_M\left(\bigcup_{H \in \mathcal{T}(M)} \{p_H \leq \alpha_H(\mathcal{F}(M))\} \right) \leq \sum_{H \in \mathcal{T}(M)} P_M\left(p_H \leq \alpha_H(\mathcal{F}(M))\right) = \sum_{H \in \mathcal{T}(M)} \alpha_H(\mathcal{F}(M)). \tag{2.12}$$

The inequality in equation (2.12) is Boole’s inequality. The final equality follows from the assumption that the p-values, p_H, satisfy $P_M(p_H \leq \alpha) = \alpha$ for true null hypotheses $H \in \mathcal{T}(M)$ and $0 < \alpha < 1$. To guarantee the single-step condition (2.11), we use the special case of the Bonferroni-Shaffer inequality in equation (2.12). Specifically, we choose the level functions $\alpha_H(\cdot)$ of the tests of single hypotheses given the set of previously rejected hypotheses such that

$$\sum_{H \in \mathcal{T}(M)} \alpha_H(\mathcal{F}(M)) \leq \alpha \quad \text{for all } M \in \mathbb{M}. \tag{2.13}$$

In Chapter 3 we show how the inheritance procedure satisfies the single-step condition (2.13).

To summarize: The single-step condition is that the single-step procedure controls the probability of no false rejections in the critical case when all false null hypotheses and none of the true null hypotheses have been rejected. Bonferroni-Shaffer procedures, such as the inheritance procedure, can guarantee the single-step condition by choosing the levels of the tests of single hypotheses such that $\sum_{H \in \mathcal{T}(M)} \alpha_H(\mathcal{F}(M)) \leq \alpha$ for all $M \in \mathbb{M}$.

Chapter 3

The Inheritance Procedure

The goal of this chapter is to describe the inheritance procedure (Goeman and Finos, 2012) for testing tree-structured hypotheses with logical implications. Recall that logical implications exist when falseness of a hypothesis implies falseness of at least one of its offspring. We begin by describing a basic form of the inheritance procedure that does not make use of logical implications amongst hypotheses. This basic form is then extended to exploit the logical implications. We show that both forms of the inheritance procedure are Bonferroni-Shaffer procedures that obey the monotonicity and single-step conditions defined in Chapter 2. Satisfying these conditions guarantees strong control of the FWER. Using Example 1, we illustrate how the extended inheritance procedure sets the level of individual tests of logically-related hypotheses. We also point out an R package that implements the extended inheritance procedure for tree-structured hypotheses of the effects of covariates.

3.1 Basic Procedure

In this section we state the basic inheritance procedure for tree-structured hypotheses that does not make use of logical implications among hypotheses. To help fix ideas, the discussion in the main part of this section is brief on some key points, with expanded discussions appearing in subsections at the end. The basic inheritance procedure is extended to exploit logical implications in Section 3.2.

The basic inheritance procedure is a Bonferroni-Shaffer procedure that starts by testing the hypothesis at the root of the tree structure and then moves towards the hypotheses in the leaves, as illustrated in Example 1 of the **Introduction**. The levels of the tests of individual hypothesis are treated like “wealth” that is “inherited” through the tree. In particular, if a hypothesis is rejected at some step of the procedure (analogous to death), its “alpha wealth” is inherited to unrejected hypotheses that are related. For example, the alpha wealth of a hypothesis may go to its children on the tree, if it has any. If the hypothesis doesn’t have children, its alpha wealth may go to its unrejected siblings. If the hypothesis doesn’t have children or unrejected siblings, its wealth may go to its cousins, etc. Inheritance is specified by an “heir function” $h(\cdot)$ that takes a hypothesis H as its argument, and returns a list of hypotheses that are the heirs of H. The precise definition of $h(\cdot)$ and an illustration of how it leads to the kind of inheritance described above is deferred to Section 3.1.2.

When a rejected hypothesis passes on alpha wealth to its heirs, the amount of wealth that each heir receives is proportional to its hypothesis-specific weight. The weight for a hypothesis is defined to be the sum of the weights of all leaf hypotheses that are its descendants; i.e., $w_H = \sum_{L \in \mathcal{L}_H} w_L$, where w_H is the weight for hypothesis H and $\mathcal{L}_H$ is the set of leaf hypotheses that are descendants of H. By default, the weight of each leaf hypothesis is one, but Goeman and Finos (2012) allow for alternate schemes in which the leaf hypotheses have unequal weights.

The basic inheritance procedure may be stated in terms of the heir function and hypothesis weights, as follows (quoted directly from Goeman and Finos, 2012, page 4):

Inheritance procedure without logical implications

1. Set $\alpha_T = \alpha$ for the root node T and $\alpha_H = 0$ for all other hypotheses.
2. Reject all H for which $p_H \leq \alpha_H$.
3. Inherit the α_H of every rejected hypotheses to its heirs, proportional to the weight of each heir, setting,

$$\alpha_K = \alpha_K + \alpha_H \frac{w_K}{\sum\limits_{J \in h(H)} w_J} \quad \text{for all } K \in h(H) \tag{3.1}$$

   and setting $\alpha_H = 0$.
4. Repeat 2) and 3) until convergence, i.e., when α_H does not change for any H between subsequent steps.

We show that the basic inheritance procedure is a Bonferroni-Shaffer procedure in Section 3.1.1, and that it obeys the monotonicity and single-step conditions in Section 3.1.3. Goeman and Finos (2012) note the connection between their inheritance procedure and other multiple testing procedures that can be described similarly, but with different choices for the heir function and weights. As our focus is on the inheritance procedure, we will not elaborate on comparisons to other multiple testing procedures.

Goeman and Finos claim that their method of inheriting alpha wealth leads to more interpretable results. Inheritance of alpha wealth has the effect of adaptively concentrating the level of the tests for hypotheses, in parts of the tree where hypotheses have previously been rejected. A concentration of the level of tests for hypotheses in certain parts of the tree will tend to result in clusters of rejected hypotheses. Goeman and Finos (2012) claim that such clusters of rejected hypotheses are more interpretable than rejected hypotheses scattered throughout the tree. Rejected hypotheses scattered throughout the tree are more likely with a procedure that distributes alpha wealth evenly among all unrejected hypotheses.

In the following subsections we expand on some of the points that were discussed only briefly in the above description of the basic inheritance procedure. In Section 3.1.1 we show that the basic inheritance procedure is a Bonferroni-Shaffer procedure. In Section 3.1.2 we give a detailed description of the heir function. Finally, in Section 3.1.3 we show that the basic inheritance procedure obeys the monotonicity and single-step conditions, and hence has strong control of the FWER.

3.1.1 Basic Inheritance is a Bonferroni-Shaffer Procedure

In this subsection, we show that the basic inheritance procedure is a Bonferroni-Shaffer procedure. Recall that a Bonferroni-Shaffer procedure is a sequence of single-step procedures. Each step tests all currently-unrejected hypotheses simultaneously, using a rejection rule that can depend on the set of previously-rejected hypotheses. Specifically, when hypotheses in $\mathcal{R}$ have already been rejected, the single-step procedure rejects every hypothesis $H \notin \mathcal{R}$ with p-value p_H less than a well-defined, multiplicity-adjusted level $\alpha_H(\mathcal{R})$.

The basic inheritance procedure can be seen to be a Bonferroni-Shaffer procedure as follows. The procedure starts with an initialization of the rejection rule (i.e., initialization of alpha wealth) for all hypotheses, and then iterates between the following two steps until no further hypotheses are rejected:

Single-step procedure. Consider all currently-unrejected hypotheses in $\mathcal{R}^C$. For every $H \in \mathcal{R}^C$, reject H if p_H is less than the alpha wealth of H, where the alpha wealth of H depends on the position of H in the tree of hypotheses, the set of previously rejected hypotheses in $\mathcal{R}$ and the desired FWER α.

Update of alpha wealth. Inherit the alpha wealth of every rejected hypothesis to its heirs.

Thus, in the basic inheritance procedure the rejection rule for the single-step procedure is in the spirit of a Bonferroni-Shaffer procedure because it compares p-values to the "alpha wealth" or multiplicity-adjusted levels of individual tests. What remains is to show that the alpha wealth for a hypothesis $H \in \mathcal{R}^C$ can be expressed as a well-defined function $\alpha_H(\mathcal{R})$ of $\mathcal{R}$. In Appendix A it is shown that

$$\alpha_H(\mathcal{R}) = \begin{cases} \frac{w_H}{w_T}\alpha \prod\limits_{K \in an(H)} m_K(\mathcal{R}), & an(H) \subseteq \mathcal{R}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

where $an(H)$ denotes the set of all ancestors of hypothesis H and

$$m_K(\mathcal{R}) = \frac{w_K}{\sum\limits_{J \in ch(K) - \mathcal{E}(\mathcal{R})} w_J}. \tag{3.3}$$

In establishing equation (3.2), Goeman and Finos (2012) show that the alpha allocation for hypotheses outside a given rejection set $\mathcal{R}$ does not depend on how one arrives at $\mathcal{R}$. Therefore, for $H \in \mathcal{R}^C$, $\alpha_H(\mathcal{R})$ is a well-defined function of $\mathcal{R}$ alone, and does not depend on the order of rejection sets or the number of steps leading to $\mathcal{R}$.

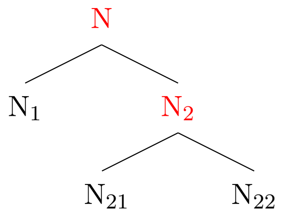
3.1.2 The Heir Function

The heir function $h(\cdot)$ takes a hypothesis as its argument and returns a list of its heirs. As shown below, the list of heirs depends on the current set of rejected hypotheses $\mathcal{R}$. To highlight the dependence on the rejection set, we write the heir function as $h_{\mathcal{R}}(\cdot)$. The purpose of the heir function is to transfer the inheritance of alpha wealth from a rejected hypothesis to its unrejected relatives. To motivate the definition of Goeman and Finos (2012) [see equation (3.5) below] we discuss two examples. The first is Example 1 and the second is a new example, Example 2, that illustrates the key concept of *extinct clades.*

With the tree of hypotheses from Example 1, we will illustrate the inheritance of alpha wealth, highlighting how heirs are defined, how their definition depends on the rejection set, and how they are used to inherit wealth from rejected hypotheses to their unrejected relatives. To help fix ideas, we will not focus on the exact amount of wealth being passed to heirs. We will, however, keep track of which unrejected hypotheses on the tree have positive alpha wealth, and will focus the discussion on the tests of these hypotheses. Since the single-step procedure rejects hypotheses whose p-values are less than their alpha wealth, we know that hypotheses with zero alpha wealth can not be rejected.

1. The basic inheritance procedure starts by testing all currently-unrejected hypotheses, which is the set of all hypotheses. Initially the hypothesis at the root node N is the only hypothesis on the tree with positive alpha wealth. All other hypotheses have zero alpha wealth, but this will change due to inheritance as the procedure moves through the tree. Suppose the hypothesis at N is rejected, so that the first set of rejected hypotheses is $\mathcal{R}_1 = \{N\}$. The heirs of N are its children N_1 and N_2. Each heir inherits part of N's alpha wealth. Thus, at the end of the first step, N_1 and N_2 are the only hypotheses with positive alpha wealth. All others have zero wealth.

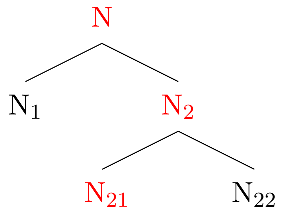
2. The second step is to test all currently-unrejected hypotheses. As described above, this amounts to testing single hypotheses having positive alpha wealth. In the example, such hypotheses are the heirs N_1 and N_2. Suppose the hypothesis at N_1 is not rejected but the hypothesis at N_2 is, so that the second set of rejected hypotheses is $\mathcal{R}_2 = \{N, N_2\}$, represented by the red nodes on the following diagram.

   The heirs of N_2 are its children N_{21} and N_{22}. Each heir inherits part of N_2's alpha wealth. Thus, at the end of the second step, hypotheses N_1, N_{21} and N_{22} have positive alpha wealth. All others have zero wealth.

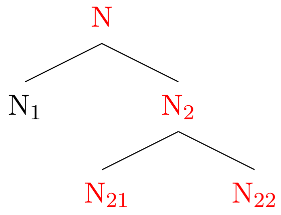
3. The third step is to test all currently-unrejected hypotheses, which amounts to testing the single hypotheses at nodes N_1, N_{21} and N_{22} having positive alpha wealth. Since

N_1's alpha wealth has not changed since the second step, we know it will not be rejected at this step of the procedure, and we focus on testing the hypotheses at nodes N_{21} and N_{22}. Suppose that N_{21} is rejected, but N_{22} is not, so that the third set of rejected hypotheses is $\mathcal{R}_3 = \{N, N_2, N_{21}\}$, represented by the red nodes in the following updated diagram.



Node N_{21} has no children, so its alpha wealth should be inherited by its nearest unrejected relative, which is its sibling N_{22}. Though N_{22} inherits the wealth of its sibling N_{21}, it is *not* an heir of N_{21}. Heirs must always be children or parents. In this case, the parent N_2 is the heir of N_{21}'s wealth. However, since N_2 has already been rejected, the inherited wealth from N_{21} is immediately passed to its child N_{22}, an unrejected hypothesis. At the end of the third step, hypotheses N_1 and N_{22} have positive alpha wealth. All others have zero wealth.

4. The fourth step of the basic inheritance procedure is to test all currently-unrejected hypotheses, which are at nodes N_1 and N_{22}. Both of these hypotheses have positive alpha wealth. The alpha wealth of N_1 has not changed, but the wealth of N_{22} has. Suppose with the addition of alpha wealth inherited from N_{21}, the hypothesis in node N_{22} is now rejected, so that the fourth set of rejected hypotheses is $\mathcal{R}_4 = \{N, N_2, N_{22}, N_{21}\}$, represented by the red nodes in the following updated diagram.



N_{22} has no children and no unrejected hypotheses in its immediate family, so its alpha wealth should be inherited by its nearest unrejected relative, which is its aunt N_1.

Though N_1 inherits the wealth of its niece N_{22}, it is *not* an heir of N_{22}. In this case the parent N_2 is the heir of N_{22}, and the grandparent N is the heir of the parent N_2. Since N has already been rejected, the inherited wealth from N_{22} (via N_2) is immediately passed to its child N_1, an unrejected hypothesis. At the end of the fourth step, the only node with positive alpha wealth is N_1.

5. The fifth and final step of the basic inheritance procedure is to test the currently-unrejected hypothesis N_1. Regardless of whether or not hypothesis N_1 is rejected, the procedure will terminate and no alpha wealth will be inherited.

For each of the inheritances in the preceding description, alpha wealth was passed from a rejected hypothesis to an unrejected relative through one or more heirs. The heirs of each node were either its unrejected children, if they existed, or its parent. This suggests the following working definition for the heir function.

$$h_{\mathcal{R}}(H) = \begin{cases} ch(H) - \mathcal{R} & \text{if } H \text{ has unrejected children} \\ pa(H) & \text{otherwise} \end{cases} \tag{3.4}$$

where $ch(H)$ denotes the children of H and $pa(H)$ denotes the parent of H. As shown in the next example, this working definition is not sufficient to inherit wealth from a node to its cousin.

Example 2 is the tree of hypotheses from Example 1 augmented by nodes N_{11} and N_{12} that are the children of N_1, as illustrated in Figure 3.1.

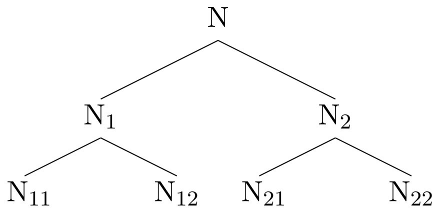
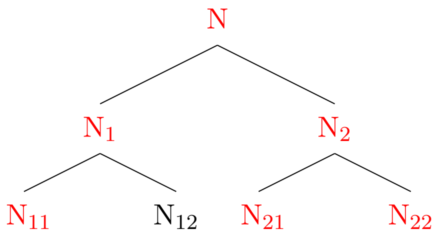


Figure 3.1: Second example of tree-structured hypotheses

1. Suppose that the first two steps of the basic inheritance procedure applied to this example have resulted in rejection of N, N_1 and N_2; i.e., the first two rejection sets are $\mathcal{R}_1 = \{N\}$ and $\mathcal{R}_2 = \{N, N_1, N_2\}$. The nodes with positive alpha wealth at the end

of these two steps will be the leaf nodes N_{11}, N_{12}, N_{21} and N_{22}. All other nodes have zero wealth.

2. The third step is to test all currently-unrejected hypotheses, which are at the leaf nodes. All of the leaf nodes have positive alpha wealth. Suppose the hypothesis at N_{12} is not rejected, but that N_{11}, N_{21} and N_{22} are rejected, so that the third set of rejected hypotheses is $\mathcal{R}_3 = \{N, N_1, N_2, N_{11}, N_{21}, N_{22}\}$, represented by the red nodes in the following diagram.



The rejected nodes N_{11}, N_{21} and N_{22} have no children and should inherit their alpha wealth to their nearest unrejected family member, node N_{12}. Though N_{12} inherits the wealth of N_{11}, N_{21} and N_{22} it is *not* an heir of any of them. To illustrate ideas, let us focus on the wealth N_{12} inherits from its cousin N_{21}. N_2 is the heir of N_{21}, and N is the heir of N_2. The next steps should be to specify N_1 as the heir of N and, finally, N_{12} as the heir of N_1. According to equation (3.4), while N_{12} is the heir of N_1, N_1 is not the heir of N because it has already been rejected. In fact, according to equation (3.4), N has *no* heirs because both of its children have already been rejected. Thus, equation (3.4) needs to be modified so that N_1 is the heir of N. Goeman and Finos (2012) circumvent these problems by defining what we refer to as "extinct clades". In the example, the rejected node N_2 has only rejected descendants and is therefore defined to be in an extinct clade. By contrast, the rejected nodes N and N_1 have an unrejected descendant N_{12} and are therefore defined to be outside of an extinct clade. For a given rejection set $\mathcal{R}$, the set of nodes in extinct clades is denoted $\mathcal{E}(\mathcal{R})$ and is formally defined as

$$\mathcal{E}(\mathcal{R}) = \{H \in \mathcal{R} : of(H) \subseteq \mathcal{R}\}$$

where $of(H)$ is the collection of descendant, or offspring nodes of H, with the convention that $of(H) = \varnothing$ when H is a leaf node. For example, $\mathcal{E}(\mathcal{R}_3) = \{N_2, N_{21}, N_{22}\}$.

The rejected node N_1 will be an heir of the rejected node N if we define heirs to be children outside of extinct clades. A revised definition of the heir function is therefore

$$h_{\mathcal{R}}(H) = \begin{cases} ch(H) - \mathcal{E}(\mathcal{R}) & \text{if any such children exist} \\ pa(H) & \text{otherwise} \end{cases} \tag{3.5}$$

With the revised definition of the heir function in equation (3.5), N_1 is the heir of N.

3.1.3 The Monotonicity and Single-Step Conditions

In this subsection, we show that the basic inheritance procedure obeys the monotonicity and single-step conditions. Hence, the basic inheritance procedure has strong control of the FWER.

1. Monotonicity condition

   Recall from Section 3.1.1 that the basic inheritance procedure is a Bonferroni-Shaffer procedure. In Chapter 2 we saw that for a Bonferroni-Shaffer procedure the monotonicity condition simplifies to the requirement that the alpha wealth or test levels of unrejected hypotheses do not decrease as more hypotheses are rejected. The basic inheritance procedure fulfills this requirement because the alpha wealth of an unrejected hypothesis either stays the same, or increases by inheritance as more hypotheses are rejected.

2. Single-step condition

   Recall that the single-step condition is familywise error control for any parameter value M when all the false null hypotheses in $\mathcal{F}(M)$ have been rejected and none of the true null hypotheses in $\mathcal{T}(M)$ have been rejected. In Chapter 2 we saw that for a Bonferroni-Shaffer procedure the single-step condition simplifies to

   $$\sum_{H \in \mathcal{T}(M)} \alpha_H(\mathcal{F}(M)) \leq \alpha \quad \text{for all } M \in \mathbb{M}. \tag{2.13}$$

   where $\alpha_H(\mathcal{R})$ denotes the alpha wealth for hypothesis H given the set $\mathcal{R}$ of previously-rejected hypotheses. Equation (2.13) says that, when $\mathcal{F}(M)$ is the rejection set, the total amount of alpha wealth among hypotheses outside of $\mathcal{F}(M)$ (i.e., in $\mathcal{T}(M)$) is no bigger than α. For sets $\mathcal{R}$ of previously rejected hypotheses,

   $$\sum_{H \in \mathcal{R}^C} \alpha_H(\mathcal{R}) \leq \alpha \tag{3.6}$$

because, at each step of the procedure, the total alpha wealth α stays the same, and is allocated to the currently-unrejected hypotheses in $\mathcal{R}^C$. That is, equation (3.6) holds with equality for any $\mathcal{R}$ including $\mathcal{F}(M)$.

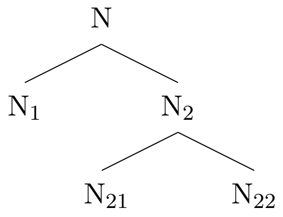
3.2 Extension to Account for Logical Implications Among Hypotheses

In this section, we will describe how the basic inheritance procedure can be extended to take advantage of logical implications among hypotheses. The basic inheritance procedure made use of the structure of the tree of hypotheses, to order the testing of hypotheses and determine which nodes should receive the alpha wealth of rejected hypotheses, but did not exploit logical implications. We begin this section with an illustration of logical implications between hypotheses, using Example 1. We then re-state the monotonicity and single-step conditions, to motivate an extension of the basic procedure that exploits logical implications, known as Shaffer improvement. Using Example 1, we illustrate how certain alpha levels for tests of single hypotheses can be increased by a so-called Shaffer factor, without violating the monotonicity and single-step conditions. We then present the formal definition of the Shaffer factor. The section concludes with a note on an R package that implements the extended inheritance procedure for tree-structured hypotheses of covariate effects.

In Example 1 from the **Introduction**, the linear model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

described the influence of three SNPs on a quantitative trait. In this example, the first SNP is from one candidate gene and the second and third SNPs are from another. Based on this structure, the hypotheses were arranged on the tree in Figure 1.1, reproduced below.



The root node is the null hypothesis of no association between the SNPs and the trait; i.e., $N : \beta_1 = \beta_2 = \beta_3 = 0$. The nodes at the next level are the null hypotheses of no association between the trait and SNPs in the first and second genes, respectively; i.e., $N_1 : \beta_1 = 0$ and $N_2 : \beta_2 = \beta_3 = 0$. The nodes in the bottom level are the null hypotheses of no trait association with SNPs in the second gene; i.e., $N_{21} : \beta_2 = 0$ and $N_{22} : \beta_3 = 0$. The nodes are arranged such that each parent hypothesis is the intersection of its children. Thus, falseness of a parent hypotheses implies falseness of at least one child hypothesis. For example, falseness of N_2 implies falseness of at least one of N_{21} or N_{22}. This implication means that a set of hypotheses that includes N_2 but not N_{21} and N_{22} is only a partial list of false null hypotheses – a so-called *incongruent set* of hypotheses, as defined by Goeman and Finos. By contrast, a *congruent set* of hypotheses can be a complete list of false null hypotheses.

To motivate Shaffer improvement, we now re-state the monotonicity and single-step conditions for a Bonferroni-Shaffer procedure, such as the basic inheritance procedure. The monotonicity condition is that the test levels of unrejected hypotheses do not decrease as more hypotheses are rejected. In the context of Shaffer improvement, the relevant implication of monotonicity is that for an unrejected hypothesis H and an incongruent rejection set $\mathcal{R}$, $\alpha_H(\mathcal{R}) \leq \alpha_H(\mathcal{S})$ for every congruent set $\mathcal{S}$ that contains $\mathcal{R}$ but not H. The single-step condition is that

$$\sum_{H \in \mathcal{F}(M)^C} \alpha_H(\mathcal{F}(M)) \leq \alpha \quad \text{for all } M \in \mathbb{M}.$$

where $\alpha_H(\mathcal{R})$ denotes the alpha wealth for hypothesis H given the set $\mathcal{R}$ of previously-rejected hypotheses. Note that the single-step condition only applies to congruent sets of false null hypotheses; i.e., to sets that are $\mathcal{F}(M)$ for some parameter value M. For incongruent rejection sets, the levels of tests of unrejected hypotheses are not constrained by the single-step condition, and can be made as large as possible, so long as they do not violate the monotonicity condition. The enlargement of test levels to exploit logical implications is known as Shaffer improvement, and the factor by which each test level is increased is known as the Shaffer factor.

We illustrate Shaffer factors using Example 1. For each of the rejection sets $\mathcal{R}$ that could be obtained by applying the basic inheritance procedure to Example 1, the levels $\alpha_H(\mathcal{R})$ of the tests of unrejected hypotheses are given in Table 3.1.

Rejection sets that include a parent node and none of its children are incongruent. In the example, these are the sets $\{N\}, \{N, N_2\}$ and $\{N, N_1, N_2\}$, indicated in red in Table 3.1. Shaffer improvement applies only to incongruent sets. For congruent sets, the test levels for unrejected hypotheses are constrained by the single-step condition and are therefore not

Table 3.1: Test levels of the basic inheritance procedure for rejection sets from Example 1. Incongruent sets are indicated in red.

	Rejection sets $\mathcal{R}$									
	$\varnothing$	N	N	N	N	N	N	N	N	N
			N_1	N_2	$N_1 N_2$	N_2	N_2	N_2	$N_1 N_2$	$N_1 N_2$
						N_{21}	N_{22}	$N_{21} N_{22}$	N_{21}	N_{22}
$\alpha_N(\mathcal{R})$	α	-	-	-	-	-	-	-	-	-
$\alpha_{N_1}(\mathcal{R})$	0	$\alpha/3$	-	$\alpha/3$	-	$\alpha/3$	$\alpha/3$	α	-	-
$\alpha_{N_2}(\mathcal{R})$	0	$2\alpha/3$	α	-	-	-	-	-	-	-
$\alpha_{N_{21}}(\mathcal{R})$	0	0	0	$\alpha/3$	$\alpha/2$	-	$2\alpha/3$	-	-	α
$\alpha_{N_{22}}(\mathcal{R})$	0	0	0	$\alpha/3$	$\alpha/2$	$2\alpha/3$	-	-	α	-

changed. We illustrate Shaffer improvement with the incongruent rejection set $\{N\}$ in the second column of Table 3.1. For this rejection set there are four unrejected hypotheses, N_1, N_2, N_{21} and N_{22}. Recall that when testing unrejected hypotheses, the basic inheritance procedure effectively only tests those with positive alpha wealth, because hypotheses with zero alpha wealth can't be rejected. The inheritance procedure with Shaffer improvement also only tests hypotheses with positive alpha wealth. Thus we only compute Shaffer factors for the hypotheses N_1 and N_2 because N_{21} and N_{22} have zero wealth.

We start with the hypothesis N_1. The constraint on $\alpha_{N_1}(\{N\})$ that is imposed by monotonicity is that $\alpha_{N_1}(\{N\}) \leq \alpha_{N_1}(\mathcal{S})$ for any congruent set $\mathcal{S}$ that contains $\{N\}$, but not N_1. We find three congruent sets that contain $\{N\}$ but not N_1: $\mathcal{S}_1 = \{N, N_2, N_{21}\}$, $\mathcal{S}_2 = \{N, N_2, N_{22}\}$, and $\mathcal{S}_3 = \{N, N_2, N_{21}, N_{22}\}$. The test levels for N_1, given each of these three congruent rejection sets, are $\alpha_{N_1}(\mathcal{S}_1) = \alpha/3$, $\alpha_{N_1}(\mathcal{S}_2) = \alpha/3$ and $\alpha_{N_1}(\mathcal{S}_3) = \alpha$, respectively. The test level $\alpha_{N_1}(\{N\})$ can be as large as the minimum of these three test levels, which is $\alpha/3$. Thus, Shaffer improvement yields a test level of $\alpha/3$, which is the same as the test level from the basic inheritance procedure. The Shaffer factor $s_{N_1}(\{N\})$ for hypothesis N_1 and incongruent rejection set $\{N\}$ is therefore 1.

We now compute the Shaffer factor for the hypothesis N_2. The constraint on $\alpha_{N_2}(\{N\})$ that is imposed by monotonicity is that $\alpha_{N_2}(\{N\}) \leq \alpha_{N_1}(\mathcal{S})$ for any congruent set $\mathcal{S}$ that contains $\{N\}$, but not N_2. The only congruent set that meets these criteria is $\mathcal{S}_1 = \{N, N_1\}$. The test level for N_2, given this congruent rejection set, is $\alpha_{N_2}(\mathcal{S}_1) = \alpha$, so that $\alpha_{N_2}(\{N\})$ can be as large as α. Thus, for the hypothesis N_2 and incongruent rejection set $\{N\}$, Shaffer improvement yields a test level of α, which is bigger than the test level of $2\alpha/3$ from the basic inheritance procedure by a Shaffer factor of $s_{N_2}(\{N\}) = 3/2$. The complete list of Shaffer factors for Example 1 is given in Table 3.2.

Table 3.2: Shaffer factors for Example 1.

	Incongruent sets $\mathcal{R}$		
Hypothesis	N	N N_2	N N_1N_2
$s_N(\mathcal{R})$	-	-	-
$s_{N_1}(\mathcal{R})$	1	1	-
$s_{N_2}(\mathcal{R})$	3/2	-	-
$s_{N_{21}}(\mathcal{R})$	-	2	2
$s_{N_{22}}(\mathcal{R})$	-	2	2

We did not use any specific features of the basic inheritance procedure when computing Shaffer factors for Example 1, only the test levels implied by the procedure and the notions of congruent and incongruent sets. Goeman and Finos (2012) provide a general form for the Shaffer factor that can be used to improve *any* sequentially rejective procedure:

$$s_H(\mathcal{R}) = \frac{\min\{\alpha_H(S) : H \notin S \supseteq \mathcal{R}, \text{S is congruent}\}}{\alpha_H(\mathcal{R})}. \tag{3.7}$$

However, as shown in Appendix B, the general formula can be rewritten for the basic inheritance procedure as:

$$s_H = \begin{cases} \frac{\mu_H + w_H}{\mu_H - \nu_H + w_H} & \text{if } H \notin \mathcal{R}, si(H) \subseteq L \backslash \mathcal{R}, \\ 1 & \text{otherwise.} \end{cases} \tag{3.8}$$

where $\mu_H = \sum\limits_{J \in si(H)} w_J$, $\nu_H = \min_{J \in si(H)} w_J$ and $si(H)$ denotes the siblings of H.

3.2.1 The `globaltest` R package

Goeman *et al.* (2010) developed an R package `globaltest` for testing whether groups of covariates are associated with a response variable. In this case, the groups of covariates being tested for association correspond to nested hypotheses. By virtue of their nesting, the hypotheses are logically related and so the inheritance procedure with logical implications can be applied. The R package is available on Bioconductor and implements the inheritance procedure for logically related hypotheses in the `covariates` function.

Appendix A

A.1 Derivations of the critical value function for the inheritance procedure

We will develop the derivation of the $\alpha_H(\mathcal{R})$ function in equation (3.2) based on step 3 of the basic inheritance procedure and heir function $h_{\mathcal{R}}(H)$ defined in equation (3.1) regardless of the order in which hypotheses have been rejected.

Based on the basic inheritance procedure, if $J = pa(H)$ is rejected, we can write the amount of α_J inherited by each child H as,

$$\alpha_J \frac{w_H}{\sum\limits_{I \in h_{\mathcal{R}}(H)} w_I} = \alpha_J \frac{w_H}{\sum\limits_{I \in ch(J) - \mathcal{E}(\mathcal{R})} w_I}$$

By definition we have,

$$m_J(\mathcal{R}) = \frac{w_J}{\sum\limits_{I \in ch(J) - \mathcal{E}(\mathcal{R})} w_I}$$

Therefore,

$$\alpha_J \frac{w_H}{\sum\limits_{I \in ch(J) - \mathcal{E}(\mathcal{R})} w_I} = \alpha_J \frac{w_H}{w_J} m_J(\mathcal{R}) = \alpha_J \frac{w_H}{w_{pa(H)}} m_{pa(H)}(\mathcal{R})$$

Let us use mathematical induction for the proof. Suppose for all $\mathcal{S} \subset \mathcal{R}$, we choose some $J \in \mathcal{R}$ and some $H \notin \mathcal{R}$ and let $\mathcal{S} \cup \{J\} = \mathcal{R}$.

1. When $\mathcal{R} = \varnothing$, we have $an(H) \nsubseteq \mathcal{R}$ and thus critical values both for the basic inheritance procedure and critical value function $\alpha_H(\mathcal{R})$ are identical. i.e.

$$\alpha_I = \alpha_H(\mathcal{R}) = 0$$

2. Suppose that when the current rejected set is $\mathcal{S}$, the alpha-wealth allocated to H before the rejection of J is,

$$\alpha_H(\mathcal{S}) = \begin{cases} \frac{w_H}{w_T}\alpha \prod_{K \in an(H)} m_K(\mathcal{S}), & an(H) \subseteq \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases}$$

3. Let the wealth inherited by H after rejection of J be $\widetilde{\alpha}_H$. Then, we need to show that,

$$\widetilde{\alpha}_H = \alpha_H(R) = \begin{cases} \frac{w_H}{w_T}\alpha \prod_{K \in an(H)} m_K(R), & an(H) \subseteq R, \\ 0 & \text{otherwise.} \end{cases}$$

   We assume that $an(H) \subseteq R$, and that J is not offspring of H as the offspring is tested and rejected only after the rejection of its ancestor. We will develop the proof for the case where J is an ancestor of H. Similar proof can be derived for other cases where J and H have different relationships.

In general, let us assume that there are $\{K_1, K_2, ..., K_n\}$ ancestors that are rejected, between J and H even though J might be the parent of H in a practical sense. Here, we have $an(H) \nsubseteq S$ as S does not include J. Therefore, $\alpha_H(S) = 0$. We can express the amount of wealth inherited by H after the rejection of J through rejected ancestors $\{K_1, K_2, ..., K_n\}$ as,

$$\begin{aligned}
\widetilde{\alpha}_H = \alpha_H(S) + \alpha_J(S)\frac{w_{K_1}}{\sum\limits_{I \in ch(K_1)-\mathcal{E}(\mathcal{R})} w_I} * \frac{w_{K_2}}{\sum\limits_{I \in ch(K_2)-\mathcal{E}(\mathcal{R})} w_I} * ... * \frac{w_{K_n}}{\sum\limits_{I \in ch(K_n)-\mathcal{E}(\mathcal{R})} w_I} \\
= \alpha_J(S)\frac{w_{K_1}}{w_{pa(K_1)}} m_J(\mathcal{R}) * \frac{w_{K_2}}{w_{pa(K_2)}} m_{K_1}(\mathcal{R}) * ... * \frac{w_{K_n}}{w_{pa(K_n)}} m_{K_{n-1}}(\mathcal{R}) * \frac{w_H}{w_{pa(H))}} m_{K_n}(\mathcal{R}) \\
= \alpha_J(S)\frac{w_{K_1}}{w_J} m_J(\mathcal{R}) * \frac{w_{K_2}}{w_{K_1}} m_{K_1}(\mathcal{R}) * ... * \frac{w_{K_n}}{w_{K_{n-1}}} m_{K_{n-1}}(\mathcal{R}) * \frac{w_H}{w_{K_n}} m_{K_n}(\mathcal{R}) \\
= \alpha_J(S)\frac{w_H}{w_J} \prod_{K \in an(H) \setminus an(J)} m_K(\mathcal{R}) \qquad \text{(A.1)}
\end{aligned}$$

On the other hand, by the induction assumption the $\alpha_J(S)$ or the α level that assigned to J when the current rejected set is S is equal to,

$$\alpha_J(S) = \frac{w_J}{w_T}\alpha \prod_{K \in an(J)} m_K(S)$$

after plugging in $\alpha_J(S)$, the equation (A.1) becomes,

$$= \alpha \frac{w_H}{w_T} \left(\prod_{K \in an(H) \setminus an(J)} m_K(\mathcal{R}) \right) \left(\prod_{K \in an(J)} m_K(S) \right)$$

$$\widetilde{\alpha}_H = \alpha \frac{w_H}{w_T} \prod_{K \in an(H)} m_K(\mathcal{R})$$

as $m_K(S) = m_K(\mathcal{R})$ for all $K \in an(J)$.

Appendix B

B.1 The equivalence of two definitions of a Shaffer factor

Let us show the connection between the general definition in (3.7) and specific definition in (3.8) of a Shaffer factor in the inheritance procedure through the toy example. Let us assume, the current rejected set is $\mathcal{R} = \{N, N_2\}$ and we want to calculate a Shaffer factor for $H = N_{21}$. Then, the possible congruent set which yields the minimum $\alpha_H(S)$ is $S = \{N, N_2, N_{22}\}$. In addition, let ancestor $k_1 = N$ and ancestor $k_2 = N_2$.

By the general definition, we know that

$$\begin{aligned} s_H &= \min\{\frac{\alpha_H(S)}{\alpha_H(\mathcal{R})} : H \notin S \supseteq \mathcal{R}, \text{S congruent}\} \\ &= \frac{\min\{\alpha_H(S) : H \notin S \supseteq \mathcal{R}, \text{S congruent}\}}{\alpha_H(\mathcal{R})} \end{aligned}$$

Based on the $\alpha_H(\mathcal{R})$ function in 3.2, we have,

$$\begin{aligned} s_H &= \frac{\min\{\frac{w_H}{w_T}\alpha \prod\limits_{K\in an(H)} m_K(S) : H \notin S \supseteq \mathcal{R}, \text{S congruent}\}}{\frac{w_H}{w_T}\alpha \prod\limits_{K\in an(H)} m_K(\mathcal{R})} \\ &= \frac{\prod\limits_{K\in an(H)} \frac{w_K}{\sum\limits_{J\in ch(K)-\mathcal{E}(S)} w_J}}{\prod\limits_{K\in an(H)} \frac{w_K}{\sum\limits_{J\in ch(K)-\mathcal{E}(R)} w_J}} \\ &= \frac{\frac{w_{K_1}}{\sum\limits_{J\in ch(K_1)-\mathcal{E}(S)} w_J}}{\frac{w_{K_1}}{\sum\limits_{J\in ch(K_1)-\mathcal{E}(R)} w_J}} * \frac{\frac{w_{K_2}}{\sum\limits_{J\in ch(K_2)-\mathcal{E}(S)} w_J}}{\frac{w_{K_2}}{\sum\limits_{J\in ch(K_2)-\mathcal{E}(\mathcal{R})} w_J}} \end{aligned}$$

For both the current rejected set and congruent set, the ancestors $\{N, N_2\}$ are the same for $H = N_{21}$ and thus have the same weight. Therefore, we have in further,

$$s_H = \frac{\dfrac{\sum\limits_{J \in ch(K_1) - \mathcal{E}(\mathcal{R})} w_J}{\sum\limits_{J \in ch(K_1) - \mathcal{E}(S)} w_J}}{\dfrac{\sum\limits_{J \in ch(K_2) - \mathcal{E}(\mathcal{S})} w_J}{\sum\limits_{J \in ch(K_2) - \mathcal{E}(R)} w_J}}$$

Moreover, when $k_1 = N$, we have $m_{K_1}(\mathcal{R}) = m_{K_1}(S)$ as there is no extinct branch. Hence,

$$\begin{aligned} s_H &= \frac{\sum\limits_{J \in ch(K_2) - \mathcal{E}(\mathcal{R})} w_J}{\sum\limits_{J \in ch(K_2) - \mathcal{E}(S)} w_J} \\ &= \frac{w_{N_2}}{w_{N_2} - w_{N_{22}}} \\ &= \frac{w_{N_{21}} + w_{N_{22}}}{w_{N_{21}} + w_{N_{22}} - w_{N_{22}}} \end{aligned}$$

Bibliography

Dudoit, S. and van der Laan, M. J. (2007). *Multiple Testing Procedures with Applications to Genomics (Springer Series in Statistics)*. Springer.

Falconer, D. and Mackay, T. (1996). *Introduction to Quantitative Genetics*. Longman.

Goeman, J. and Finos, L. (2012). The inheritance procedure: multiple testing of tree-structured hypotheses. *Statistical Applications in Genetics and Molecular Biology*, **11**(1), 1–18.

Goeman, J., Oosting, J., Finos, L., and Solari, A. (2010). *The Global Test and the globaltest R package*.

Goeman, J. J. and Solari, A. (2010). The sequential rejection principle of familywise error control. *The Annals of Statistics*, **38**(6), 3782–3810.

Meinshausen, N. (2008). Hierarchical testing of variable importance. *Biometrika*, **95**(2), 265–278.

Westfall, P. H. and Young, S. (1993). *Resampling-Based Multiple Testing: Examples and Methods for P-Value Adjustment*. A Wiley-Interscience publication. Wiley.