Author: Meijer, Rosa Janna
Title: Efficient multiple testing for large structured problems
Issue Date: 2015-06-30

3

A region-based multiple testing method for hypotheses ordered in space or time

Abstract

We present a multiple testing method for hypotheses that are ordered in space or time. Given such hypotheses, the elementary hypotheses as well as regions of consecutive hypotheses are of interest. These region hypotheses not only have intrinsic meaning but testing them also has the advantage that (potentially small) signals across a region are combined in one test. Because the expected number and length of potentially interesting regions are usually not available beforehand, we propose a method that tests all possible region hypotheses as well as all individual hypotheses in a single multiple testing procedure that controls the familywise error rate. We start at testing the global null-hypothesis and when this hypothesis can be rejected we continue with further specifying the exact location/locations of the effect present. The method is implemented in the `R` package `cherry` and is illustrated on a DNA copy number data set.

This chapter has been published as: Rosa J. Meijer, Thijmen J.P. Krebs and Jelle J. Goeman. (2015) A region-based multiple testing method for hypotheses ordered in space or time. *Statistical Applications in Genetics and Molecular Biology 14* (1), 1–19

3.1 Introduction

In many biological settings, the data points measured are not only of interest individually, but also on a region level. Data examples include copy number variation data, methylation data or SNP (single nucleotide polymorphism) data. For each of these data types, groups of neighboring data points make up genomic regions that can have high biological relevance such as CpG islands, promotor regions or genes. Considering the data at the region level is not only useful because these regions can be the fundamental units of interest, but also because these regions can have an increased signal-to-noise-ratio (Benjamini and Heller, 2007). Even if individual signals are too weak to be identifiable, neighboring data points tend to contain similar signals and pooling this information can result in detectable effects.

Several innovative procedures have been developed to detect (genomic) regions associated with a certain outcome variable. These include scanning statistics, bump-hunting techniques, peak-detection methods and marked point process models (see e.g. Jaffe et al., 2012; Hatsuda, 2012; Schwartzman et al., 2011). Most of these methods have been developed however in view of specific applications and for that reason require certain predefined choices with respect to the underlying unknown number of associated regions, the length of these regions and the exact model relating the covariates to the outcome variable. Besides, the multiple testing issue raised by the search for specific regions within a space of numerous candidates is not always clearly addressed. Similar methods that control the false discovery rate in the context of spatial signals or random fields have also been developed (Pacifico et al., 2004; Benjamini and Heller, 2007), but here too assumptions on the number of clusters or restrictions on the form of the test-statistic have to be made.

We developed an alternative method in which all possible regions of all possible lengths are tested in a single multiple testing procedure. Our approach can be seen to fall in the broader category of sequentially rejective multiple testing procedures that control the familywise error rate (FWER). Other methods falling in this category can be found for example in Bretz et al. (2009); Burman et al. (2009); Meinshausen (2008); Westfall and Tobias (2007). Our proposed method strongly controls the FWER and aims to find regions as well as individual data points that are associated with a certain outcome variable, where association is measured by a user-specified hypothesis test. By using global tests that are powerful in detecting groups of covariates in which many covariates are weakly associated with the response, such as the tests developed by Goeman et al. (2004) or Mansmann and Meister (2005), our method will often enable us to find influential regions, even if individual association cannot be shown.

The method searches through the set of all possible regions, which implies that we do not have to specify the length or number of potentially interesting regions in advance and which also ensures that the individual data points (i.e. the elementary hypotheses) can be considered as units of interest themselves. The top-down testing order furthermore allows us to use information from earlier rejections in subsequent steps. Because a region can only be associated with the outcome if the same holds for one or more of its subregions, being able to reject a region null-hypothesis (stating that the region is not associated with the response) indicates that not all remaining subregion null-hypotheses can simultaneously be true. Using these logical relations, known as restricted combinations (Shaffer, 1986), improves the power of our multiple testing procedure. From this perspective, there is even an extra benefit in looking at regions in addition to looking at individual points, namely the fact that the multiple testing burden can diminish rather than increase in comparison to FWER controlling methods that only test the elementary hypotheses. Although similar behavior is known for the closed testing procedure developed by Marcus et al. (1976), this procedure cannot be used for situations in which there are more than approximately 30 elementary hypotheses, because the number of tests needed is exponential in the number of elementary hypotheses. The number of tests needed for our method is quadratic in the number of elementary hypotheses, and the time to carry out the multiple testing procedure will usually be dominated by the time that is required to perform all individual tests.

Our region method can be used for several data types, in combination with any valid hypothesis test for the regions and will control the FWER without making additional assumptions on the joint distribution of the test statistics used to calculate the individual non-multiplicity adjusted p-values. In addition, the final results enable us to derive confidence statements of the form given in Goeman and Solari (2011) on how many individual data points in a certain region have to be associated with the outcome variable.

In the next section, we will describe our method in detail. The exact algorithms used will be discussed in section 3.3 and in section 3.4 we will demonstrate our method on DNA copy number data as well as on simulated data. In the discussion, some possible extensions will be mentioned. Software is available in the `R` package `cherry`.

3.2 Region hypotheses

Suppose we have m ordered hypotheses $H_1, \dots, H_m$, the so-called elementary hypotheses. Although the exact form of these hypotheses does not have to be specified for our multiple testing method, we will, for ease of understanding, explain the theory on the basis of an example in which the elementary hypotheses are of the following form

$$H_i \colon \beta_i = 0,$$

where β_i is the regression coefficient connecting covariate $\boldsymbol{x}_i$ to an outcome variable $\boldsymbol{y}$ in some not further specified regression model. Our elementary null-hypotheses thus correspond to statements claiming there is no relation between certain covariates and the outcome variable.

Because of the ordering, the elementary hypotheses per se are not the only hypotheses of interest. All hypotheses representing *an intersection of consecutive elementary hypotheses* are worth testing as well. These intersection hypotheses, which we will refer to as *region hypotheses*, can be represented by the first and last elementary hypothesis they

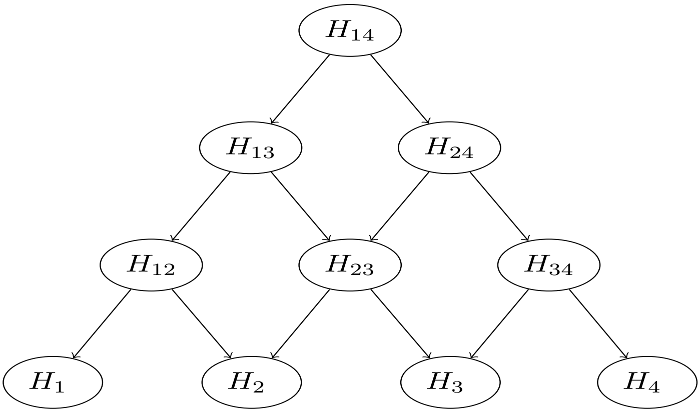


Figure 3.1: The collection of region hypotheses for $m = 4$ elementary hypotheses.

contain. The region hypothesis ranging from elementary hypotheses H_i to H_j will be denoted by H_{ij}. The complete set of region hypotheses is given by:

$$\mathcal{H} = \left\{ H_{ij} = \bigcap_{k \in \{i,\ldots,j\}} H_k, \text{ with } 1 \leq i \leq j \leq m \right\}.$$

This set can be visualized as a graph, where the nodes correspond to the different region hypotheses and the edges define the underlying subset relationships. This is illustrated in Figure 3.1 for the special case in which we have $m = 4$ elementary hypotheses. The top node H_{14} represents the overall null-hypothesis, stating that none of the four covariates are related to the outcome, whereas the leaf nodes represent the elementary hypotheses that single covariates are not related to the outcome variable. The intermediate nodes relate to statements about regions of covariates. Hypothesis H_{13} for example assumes no relation between the first three covariates and the outcome, or mathematically

$$H_{13} \colon \beta_1 = \beta_2 = \beta_3 = 0.$$

The graph has a simple structure in the sense that every node (except for the leaf nodes) has two outgoing edges. In this way, every region hypothesis H_{ij} of length $k > 1$, where length denotes the number of elementary hypotheses in the intersection or, in our example, the number of regression coefficients it assumes to be zero, is connected with the two region hypotheses $H_{(i+1)j}$ and $H_{i(j-1)}$ of length $k - 1$ that are contained in H_{ij}, its so-called children.

The graph's design allows for certain logical reasoning. Namely, when some hypothesis is false, we can be sure that at least one of its children must be false as well.

Assume for example that hypothesis H_{14} is false. This then means that at least one of the β_i's with $i \in \{1, 2, 3, 4\}$ is unequal to zero, from which it follows that hypothesis $H_{13}\colon \beta_1 = \beta_2 = \beta_3 = 0$ and hypothesis $H_{24}\colon \beta_2 = \beta_3 = \beta_4 = 0$ can no longer be simultaneously true. This observation leads to the introduction of so-called congruent sets.

We will call a set $\mathcal{R} \subseteq \mathcal{H}$ *congruent* if, given the logical relationships, it can be the complete set of false hypotheses, and *incongruent* otherwise. In other words, when a rejection set $\mathcal{R}$ is *congruent*, all hypotheses that are not yet rejected can simultaneously be true, without implying the truth of any hypothesis in $\mathcal{R}$. This same definition was used by Goeman and Finos (2012). In our example, the rejection set $\mathcal{R} = \{H_{14}\}$ is thus an incongruent set, because the falseness of H_{14} implies that at least one of the hypotheses H_{13} and H_{24} is false as well. However, the augmented set $\mathcal{R} = \{H_{14}, H_{13}\}$ is not a congruent set either, because the reasoning that applied to H_{14} and its children, applies to H_{13} and its children as well. Continuing this argumentation, we see that every congruent rejection set, apart from the empty set $\mathcal{R} = \emptyset$, will have to contain at least one elementary hypothesis, which means one leaf node. If we denote the collection of leaf nodes by $\mathcal{L} = \{H_1, \ldots, H_m\}$, and introduce notation for the graph relationships ancestors an() and offspring of() in the following way

$$\text{an}(H_{ij}) = \{H_{lk} \in \mathcal{H}\colon \{l, \ldots, k\} \supset \{i, \ldots, j\}\}$$

and

$$\text{of}(H_{ij}) = \{H_{lk} \in \mathcal{H}\colon \{l, \ldots, k\} \subset \{i, \ldots, j\}\},$$

we see that every congruent rejection set will be of the following form:

$$\mathcal{R}_{congr} = \mathcal{L}' \cup \bigcup_{H \in \mathcal{L}'} \text{an}(H) \quad \text{for some } \mathcal{L}' \subseteq \mathcal{L}.$$

Every congruent rejection set thus consists of a number of elementary hypotheses and all their ancestors. Accordingly, $\mathcal{R} = \{H\}$ can be extended to a congruent rejection set by selecting one or more of its corresponding leaf nodes, given by

$$\mathcal{L}_H = \big(\{H\} \cup \text{of}(H)\big) \cap \mathcal{L}$$

and adding those and all their ancestors to the current set $\mathcal{R}$. The concept of congruent sets and their just described construction will be used extensively in the derivation of our multiple testing method.

3.2.1 A FWER controlling multiple testing procedure, based on the sequential rejection principle

Throughout this article, we will assume that we have raw (i.e. non-multiplicity corrected) p-values p_H for every region hypothesis H. The exact statistical test used to calculate these raw p-values is not important for our multiple testing method and can, if desired, even vary between the hypotheses. Additional assumptions on the correlation structure of the test statistics or p-values are not required.

Although our multiple testing method can be used with every valid local test, we want to stress that the added value of a procedure that tests all possible region hypotheses lies in its combination with a local test that is designed to detect group effects, such as for example tests by Goeman et al. (2004) or Mansmann and Meister (2005). Those tests will be able to detect larger regions, even though the significance of the elementary hypotheses within these regions cannot be established. If a consonant local test is used, there will usually be no gain in testing all regions as compared to only testing the elementary hypotheses. If a Bonferroni test is used as local test in our proposed procedure for example, we can show (as is done in the Appendix) that our procedure reduces to Holm's procedure (Holm, 1979), which would in that case be the preferred method because it is computationally simpler. The intuition behind using a procedure that tests regions should thus always be that one expects that some effects will only be visible on a region level.

Given the raw p-values, in order to determine which hypotheses can be rejected while strongly controlling the FWER at some pre-specified α-level, we have to specify carefully which significance levels α_H can be used to test each node H. Instead of immediately distributing α over all possible region hypotheses, we will proceed iteratively; in every step we will only test those hypotheses that have all their ancestor nodes rejected.

In general, an iterative multiple testing procedure will start with an empty rejection set $\mathcal{R}_0 = \emptyset$. In every subsequent step, critical values for all hypotheses will be calculated, and those hypotheses that have p-values smaller than their assigned α-level, will be added to the current rejection set. Subsequently, new critical values will be computed, based on the new rejection set, and this procedure will continue until no further rejections can be made. Formally,

$$\mathcal{R}_{i+1} = \mathcal{R}_i \cup \{H \in \mathcal{H} \setminus \mathcal{R}_i \colon p_H \leq \alpha_H(\mathcal{R}_i)\}, \tag{3.1}$$

where $\mathcal{R}_i$ is the collection of rejected hypotheses after step i and $\alpha_H(\mathcal{R})$ is a critical value function that, based on a current rejection set $\mathcal{R} \subset \mathcal{H}$, assigns certain significance levels $\alpha_H(\mathcal{R})$ to not yet rejected hypotheses $H \in \mathcal{H}$.

To determine what α-levels can be used to test the hypotheses in subsequent steps, we base our method on the sequential rejection principle (SRP) as described by Goeman and Solari (2010). This principle tells us that, in order to strongly control the familywise error rate at level α, we can use any critical value function $\alpha_H(\mathcal{R})$, as long as it satisfies two conditions. The first condition is the *monotonicity condition* that tells us that for every $\mathcal{R} \subseteq \mathcal{S} \subset \mathcal{H}$ and for every $H \in \mathcal{H} \setminus \mathcal{S}$, we must have

$$\alpha_H(\mathcal{R}) \leq \alpha_H(\mathcal{S}). \tag{3.2}$$

This says that critical values are not allowed to decrease when more hypotheses get rejected. The second condition, the *single-step condition*, is met when for all *congruent* sets $\mathcal{R} \subset \mathcal{H}$ the following holds:

$$\sum_{H \in \mathcal{H} \setminus \mathcal{R}} \alpha_H(\mathcal{R}) \leq \alpha. \tag{3.3}$$

In every step, the α_H's distributed over possibly simultaneously true hypotheses H can thus never exceed the total α-level. A sequential procedure that satisfies these two conditions will strongly control the familywise error rate, without any further assumption on the dependence structure of the individual p-values.

For our region testing procedure, we choose $\alpha_H(\mathcal{R})$ as follows:

$$\alpha_H(\mathcal{R}) = \alpha \times r_H(\mathcal{R}),$$

where we call $r_H(\mathcal{R})$ the *ratio*, since it indicates which proportion of the overall α will be donated to hypothesis H, based on rejection set $\mathcal{R}$. This ratio can only differ from 0 for nodes that are not yet rejected, but have all their ancestor nodes rejected. We will call these *reachable* nodes the "*candidates*". To determine the exact ratio for those candidates, we distinguish between congruent and incongruent rejection sets $\mathcal{R}$. If $\mathcal{R}$ is congruent, the ratio equals

$$r_H(\mathcal{R}) = \frac{|\mathcal{L}_H|}{|\mathcal{L} \setminus \mathcal{R}|}, \tag{3.4}$$

where $|\mathcal{S}|$ denotes the cardinality of a set $\mathcal{S}$. So for every candidate H, the ratio equals its length divided by the number of non-rejected leaf nodes. This results in an α-distribution that is proportional to the length of the regions.

Whenever $\mathcal{R}$ is *incongruent*, we can benefit from the information provided by the special structure of the region hypotheses. An incongruent rejection set indicates that, given that all hypotheses in the current rejection set have been correctly rejected, not all remaining hypotheses can be simultaneously true. Exactly for that reason, the single-step condition only prescribes the sum of the α_H's of all unrejected nodes H to equal α when $\mathcal{R}$ is a congruent set, and gives no strict upper bound for the sum of these α_H's when $\mathcal{R}$ is an incongruent set. In our method it will for that reason often happen that the sum of the distributed α-levels exceeds the overall α-level. In a procedure that does not use the logical relations, in every step of the procedure all distributed α-levels have to add up to one, which makes such procedures clearly less powerful than our procedure. Using the logical relations among hypotheses to obtain an increase in power was first proposed by Shaffer (1986).

Given an incongruent rejection set $\mathcal{R}$, $r_H(\mathcal{R})$ will be taken to be the minimal ratio over all *congruent* rejection sets $\mathcal{S}$ that do not have candidate hypothesis H as an element and that are an extension of the current set $\mathcal{R}$. If we denote the set of all congruent rejections sets by Φ, we get:

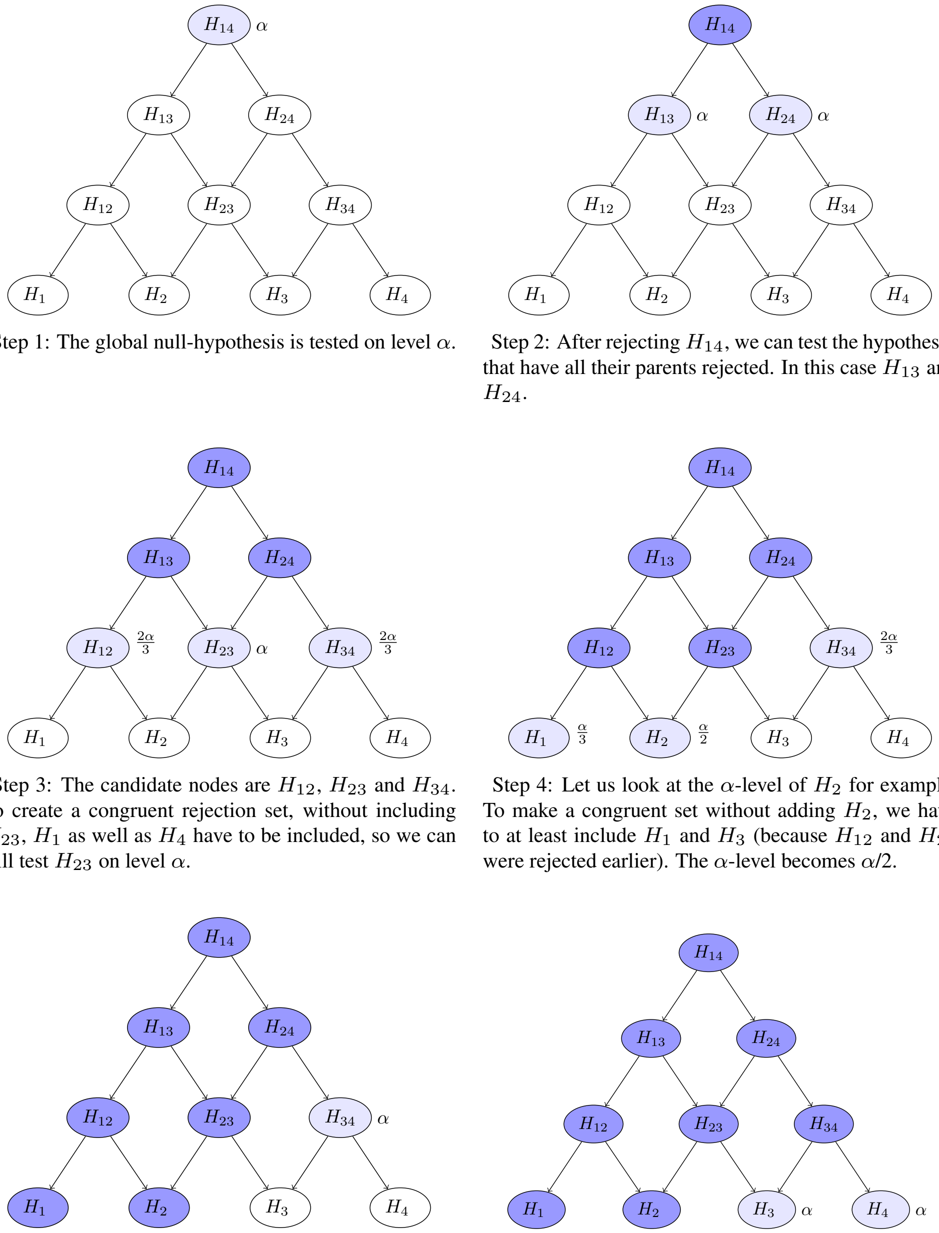
$$r_H(\mathcal{R}) = \min_{\mathcal{S} \in \Phi:\ \mathcal{R} \subset \mathcal{S}, H \notin \mathcal{S}} \frac{|\mathcal{L}_H|}{|\mathcal{L} \setminus \mathcal{S}|}. \tag{3.5}$$

The only part in the fraction that changes over the different sets $\mathcal{S}$ is the denominator. To minimize the entire fraction, the denominator has to be maximized, which means that we have to choose the congruent set $\mathcal{S}$ in such a way that the number of unrejected leaf nodes is as big as possible. This is equivalent to choosing the set $\mathcal{S}$ to have as few rejected leaf nodes as possible. Determining the ratio for a candidate H when the current rejection set is incongruent thus comes down to extending this set to a congruent set that includes as few rejected leaf nodes as possible and that does not include H. In order to extend a rejection set in this *minimal way*, we use dynamic programming techniques in which the emphasis lies on an efficient implementation in terms of computation time and memory usage. In the next section, the exact dynamic programming algorithm is described in detail.

To illustrate the just described procedure, in Figure 3.2 we use our method on a toy example in which we want to test 10 region hypotheses. At the first step, our only candidate node is node H_{14}, as indicated by the light blue color, and we start with testing this global null-hypothesis on the full α level. Because we have a congruent rejection set $\mathcal{R} = \emptyset$, we can use equation (3.4) to verify this. Let us now assume that the raw p-value of H_{14} is smaller than α, so we can reject H_{14} (indicated by a dark blue color in Figure 3.2). In step 2, we have two candidate nodes: H_{13} and H_{24}. Because the rejection set $\mathcal{R} = \{H_{14}\}$ is not a congruent set, we use equation (3.5) to find the α-levels on which we can test H_{13} and H_{24}. To find the ratio for H_{13}, we have to extend our rejection set $\mathcal{R} = \{H_{14}\}$ to a congruent set $\mathcal{S}$, without including H_{13} because this is the node we want to test and we thus assume that it can be a true hypothesis. The only possibility to extend $\{H_{14}\}$ to a congruent set without including H_{13}, is to choose $\mathcal{S} = \{H_{14}, H_{24}, H_{34}, H_4\}$. If we now divide the length of H_{13} by the number of unrejected leaf nodes ($|\mathcal{L} \setminus \mathcal{S}|$), we see that H_{13} can be tested on the full α-level and by a similar argument, the same holds for H_{24}. Here the power improvement that comes from using the logical relationships is apparent. When the logical relations are not taken into account, the α-levels of H_{13} and H_{24} should add up to α, but since we use the information from the previous rejection, we can test both hypotheses on level α while still controlling the FWER. In the remainder of the toy example, the α-levels are calculated in the same way. When we have a congruent situation, we use equation (3.4), in an incongruent situation equation (3.5) is used.

To give an example on how to extend a current rejection set in a *minimal way*, i.e. with as few rejected leaf nodes as possible, let us look at Step 3 in Figure 3.2. To calculate the α-level of node H_{12}, we have to extend the current uncongruent rejection set $\mathcal{R} = \{H_{14}, H_{13}, H_{24}\}$ to a congruent set $\mathcal{S}$ for which $H_{12} \notin \mathcal{S}$. This can either be the set $\mathcal{S} = \{H_{14}, H_{13}, H_{24}, H_{23}, H_{34}, H_3\}$ or the set $\mathcal{S} = \{H_{14}, H_{13}, H_{24}, H_{23}, H_{34}, H_3, H_4\}$. In the first case, the α-ratio will be given by 2/3 (the length of H_{12} divided by total number of unrejected leaf nodes, namely H_1, H_2 and H_4), in the second case the α-ratio will equal 1, because we have only two unrejected leaf nodes left (H_1 and H_2). It is immediately clear that, to minimize the ratio, we have to select as few leaf nodes as possible in our congruent extension $\mathcal{S}$.

It can be easily verified that our critical value function satisfies both conditions imposed by the SRP and thus strongly controls the FWER. If we look at our example, to verify the single step condition, we have to check whether the α-values add up to the overall α, in case of a congruent rejection set. In step 1 and 5, we are in a congruent situation, and we see that the α-levels indeed add up to α. This will hold in general, because



Step 1: The global null-hypothesis is tested on level α.

Step 2: After rejecting H_{14}, we can test the hypotheses that have all their parents rejected. In this case H_{13} and H_{24}.

Step 3: The candidate nodes are H_{12}, H_{23} and H_{34}. To create a congruent rejection set, without including H_{23}, H_1 as well as H_4 have to be included, so we can still test H_{23} on level α.

Step 4: Let us look at the α-level of H_2 for example. To make a congruent set without adding H_2, we have to at least include H_1 and H_3 (because H_{12} and H_{23} were rejected earlier). The α-level becomes α/2.

Step 5: A congruent situation. Only H_{34} is tested on level α.

Step 6: After rejecting H_{34}, both H_3 and H_4 can be tested on level α because at most one of them can still be true.

Figure 3.2: The region procedure: a toy example.

in a congruent situation, candidate nodes cannot have leaf nodes in common and for that reason the sum of the numerators in equation (3.4) can never exceed the denominator.

To check whether the monotonicity condition holds in our example, we should check whether the α-level that a certain unrejected node receives can only increase when more rejections occurs. If we for example look at node H_{34}, we see that its α-level equals 0 in the first two steps, increases to $2\alpha/3$ in step 3 and 4 and increases to α in the fifth step. We see that the α-level of H_{34} does not decrease and the same holds for all other nodes in the graph. In general, the ratio of an unrejected node can never decrease when more rejection have occurred because its length stays the same (which is the numerator in equation (3.4)) while the number of unrejected leaf nodes can only decrease (which is the denominator in this same equation). A formal proof is given in the appendix.

3.2.2 Confidence sets for the number of true rejections in regions chosen post-hoc

When the procedure can make no further rejections, we have found our final rejection set $\mathcal{R}$. Although both the region hypotheses as well as the elementary hypotheses have intrinsic meaning, our reasoning will often be in terms of the latter. Some of them may be an element of $\mathcal{R}$, but even if an elementary hypothesis is no part of the rejection set $\mathcal{R}$, but a larger region containing this hypothesis is, we might still be able to make statements about the number of false elementary hypotheses in this larger region. Actually, from the set $\mathcal{R}$, we can derive statements regarding the number of false hypotheses in every arbitrarily chosen set of elementary hypotheses.

The idea to derive statements regarding the number of false (or true) hypotheses in a certain set of hypotheses, was introduced by Goeman and Solari (2011). They show that exact *simultaneous* confidence sets can be constructed for the number of true rejections incurred when rejecting any specific set of elementary hypotheses, based on the rejection set obtained from a closed testing procedure. The possibility of deriving confidence sets simultaneously stems from the fact that they are all derived from a single application of the closed testing procedure. Since all rejections within the closure are simultaneously valid with probability $1-\alpha$, the same holds for all confidence sets derived from these rejections.

Although the original reasoning was based on a closed testing procedure, this same reasoning applies to our region procedure. Because all rejections within the region graph are simultaneously valid with probability $1-\alpha$, confidence statements based on these rejections will again be simultaneously valid in a *post hoc* setting.

Suppose our final rejection set is given by $\mathcal{R}$ and we want to construct a $100(1-\alpha)\%$ confidence set for the number of false hypotheses (i.e. true findings) for a given set $\mathcal{A}$ of elementary hypotheses. Such a confidence region will have the form $\{f_\alpha(\mathcal{A}), \ldots, |\mathcal{A}|\}$, where $f_\alpha(\mathcal{A})$ is the minimal number of false hypotheses in $\mathcal{A}$ that is in accordance with our rejection set $\mathcal{R}$. This lower bound is given by

$$\min_{\mathcal{S} \supseteq \mathcal{R}} |\mathcal{A} \cap \mathcal{S}|, \tag{3.6}$$

where the minimum is taken over all possible *congruent* sets $\mathcal{S}$ that are extensions of $\mathcal{R}$. Given that the rejection set contains no type 1 errors (which is the case with probability $1 - \alpha$), one of the possible congruent extensions has to represent the true set of false hypotheses. For that reason the true number of false hypotheses in $\mathcal{A}$ can, with probability $1 - \alpha$, not be smaller than the minimal number found over all congruent extensions. To calculate $f_\alpha(\mathcal{A})$ we can again use dynamic programming, as will be described in the next section.

To give an example of the construction of a confidence set, let us look at Figure 3.2 again. Say our final rejection set is the rejection set as given in Step 4. Even though none of the elementary hypotheses are rejected, we can construct a confidence set for the number of false hypotheses in the set $\mathcal{A} = \{H_1, H_2, H_3, H_4\}$. To find $f_\alpha(\mathcal{A})$, we have to construct all possible congruent extensions $\mathcal{S} \supseteq \mathcal{R}$, count the overlap between $\mathcal{S}$ and $\mathcal{A}$ and take the minimum of that. There are many possible congruent extensions of $\mathcal{R}$, but since H_{12} and H_{23} are contained in $\mathcal{R}$, whereas this holds for none of their children, we will at least have to include one node from $\{H_1, H_2\}$ and one from $\{H_2, H_3\}$ in $\mathcal{S}$ to make it a congruent extension of $\mathcal{R}$. Meeting this condition, while minimizing the overlap between $\mathcal{S}$ and $\mathcal{A}$, will come down to choosing $\mathcal{S}$ as $\mathcal{R} \cup H_2$. There is no congruent extension of $\mathcal{R}$ possible that contains fewer leaf nodes and with $100(1-\alpha)\%$ confidence we thus know that at least one of the four hypotheses in $\mathcal{A}$ has to be false, which results in the confidence set $\{1, 2, 3, 4\}$.

Compared to the standard approach of only looking at the static rejection set, this new view on the outcome gives the researcher freedom to compose his or her own preferred set of hypotheses, providing information on the risk of following up on this particular set in for example a validation experiment.

3.2.3 A weighted-version of the region-method

Until now, the α-level of any candidate hypothesis only depended on the length of this hypothesis and the specific rejection set. However, in some cases we might want to have more control over the α-levels. For that reason, an extension of the method in which the user can assign different weights to the elementary hypotheses has also been constructed. Let $w_1, \ldots, w_m$ be positive weights, indicating the "importance" of every individual hypothesis. Bigger weights can for example reflect a prior belief that a specific covariate is associated with the response. Similarly, if the elementary hypotheses would themselves be regions, the weights could reflect the lengths of these regions. We would like the α-levels of hypotheses to be proportional to their weights.

To accomplish this, we introduce a new ratio for every candidate hypothesis H. Given a *congruent* rejection set $\mathcal{R}$, this new ratio equals

$$r_H(\mathcal{R}) = \frac{\sum\limits_{H_i \in \mathcal{L}_H} w_i}{\sum\limits_{H_i \in \mathcal{L} \setminus \mathcal{R}} w_i}. \tag{3.7}$$

In this calculation, the denominator thus equals the total weight of the leaf nodes that are not yet rejected, compared to the total number of unrejected leaf nodes in our previous version. Note that this new ratio is exactly equal to the one given in formula (3.4) when all weights equal 1.
Whenever the rejection set $\mathcal{R}$ is not congruent, we again extend it in a "minimal way" to a congruent set $\mathcal{S}$. This time, this means that we have to choose our set $\mathcal{S}$ in such a way that the sum of the weights of the incorporated leaf nodes is as small as possible. When all weights equal one, this again comes down to constructing $\mathcal{S}$ with as few rejected leaf nodes as possible. Our dynamic programming algorithm will minimize the weights and can for that reason be used in both situations.

3.3 Algorithms

In this section, we will show that all computations to update the current rejection set $\mathcal{R}_i$ to the new set $\mathcal{R}_{i+1}$ can be done in $O(m)$ time, given that the raw p-values are known. Because we have $O(m^2)$ nodes in total, we know that the rejection set can only be updated $O(m^2)$ times, which makes the order of the full algorithm $O(m^3)$. Calculating all $O(m^2)$ raw p-values will usually require at least the same number of computations, so given that all p-values have to be calculated, applying the whole multiple testing procedure will not increase the order of computations. This result is mainly based on characteristics of the exact algorithm that is used to calculate the ratios for all candidate hypotheses; an algorithm which we will explain in detail. However, before presenting the details, we will first introduce a new definition.

Let our current rejection set again be given by the set $\mathcal{R}$. Now we can distinguish between two important sets of nodes. The first group are the *candidates*, the nodes that have all their ancestor nodes rejected, but are not yet rejected themselves. The second group consists of those nodes that are rejected themselves but have none of their offspring nodes rejected. We will call those nodes *implications*. A rejected leaf node is by definition also an implication. The name "implication" is chosen, because these nodes *imply* which congruent sets are an extension of $\mathcal{R}$. By definition, a congruent extension $\mathcal{S}$ of $\mathcal{R}$ must contain at least one element from $\mathcal{L}_H$ for every $H \in \mathcal{R}$. This is however equivalent to the requirement that $\mathcal{S}$ contains at least one element from $\mathcal{L}_I$ for every implication $I \in \mathcal{R}$ because every $H \in \mathcal{R}$ is by definition an implication or the ancestor of an implication and if $H \in \text{an}(I)$, then $\mathcal{L}_I \subset \mathcal{L}_H$.

From now on, every rejection set $\mathcal{R}$ can thus be extended to a congruent set, by only looking at the implications. We already used this idea in subsection 3.2.2. To construct a congruent extension $\mathcal{S}$ of $\mathcal{R}$, where $\mathcal{R}$ was chosen to be the rejection set in Step 4 of Figure 3.2, we only had to look at the rejected nodes without rejected children, namely H_{12} and H_{23}, which we would now call implications.

3.3.1 Dynamic programming algorithm to calculate ratios

Given an incongruent rejection set $\mathcal{R}$, calculating the ratio for a given candidate H thus comes down to choosing an element from $\mathcal{L}_I$ for every implication I, without choosing any element of $\mathcal{L}_H$ (because our candidate must still be a candidate in the extended set $\mathcal{S}$) and in such a way that the sum of the weights of the chosen leaf nodes is minimal. This can be done by formulating the problem in a recursive way.

We assume that we have a set of implications $\mathcal{I} = \{I_1, \ldots, I_k\}$ in which the implications are sorted on increasing left-boundaries. We will denote the left and right-boundary of implication j by $l(I_j)$ and $r(I_j)$. We now want to construct a recursive formula "mw" (abbreviation of minimal weight), for which mw(h) denotes the smallest possible weight of a subset of the first h elementary hypotheses, containing h itself and at least one leaf node of all implications I_j with $l(I_j) \leq h$. This subset is said to "satisfy" these implications. The corresponding recursive formula is given by:

$$\text{mw}(h) \quad = \quad \begin{cases} w_h & h \leq r(I_1), \\ w_h + \displaystyle\min_{l(I_{prev}) \leq h' \leq r(I_{prev})} \text{mw}(h') & \text{otherwise}, \end{cases}$$

where I_{prev} is the rightmost implication with $r(I) < h$.

To see why this formula is correct, it suffices to divide the implications with a left-boundary $l(I) \leq h$ in two categories; those with $r(I) < h$ and those with $r(I) \geq h$. The implications in the second category are directly satisfied by including h itself in our subset of elementary hypotheses with corresponding weight w_h. The minimal weight needed to satisfy all implications in the first group is exactly the minimal weight needed to reject I_{prev} and all its predecessors, which is given by the minimum of mw(h) over all leaf nodes of I_{prev}. If there are no previous implications, adding h to the rejection set suffices, which explains the first line.

Although we only need to compute m instances of mw(h), a minimum has to be calculated repeatedly over the different implications. Calculating such a minimum will normally cost $O(m)$ steps, from which it would follow that we need $O(m^2)$ steps to calculate all values of mw(h) for $1 \leq h \leq m$. However, we can translate the recurrence relation into an efficient dynamic program that runs in $O(m)$ space and time, by "updating" the minimum instead of recalculating it in every step.

Each minimum can be calculated in turn from the currently known values mw(h) (note that the first $r(I_1)$ values of mw(h) can be calculated directly), and from the new minimum, new values of mw(h) can be calculated. Calculating all minima can be viewed as a problem in which the minimum over a sliding window has to be calculated. At each iteration, the window ranges from $l(I_i)$ to $r(I_i)$ for a certain implication I_i and both its left and right-bound will increase at the next iteration, because the implications are sorted and can by definition not be nested. Our aim is to construct a sequence Q that only has potential minima as its values and from which actual minima can be easily retrieved. In the first iteration, all instances of mw(h) where h ranges from $l(I_1)$ to $r(I_1)$ are possible nominees for inclusion in Q. They are appended one by one, but before we add a value, we check whether there are previous values that are larger (or equal) and we remove those because they can never be a minimum for any later implication. The sequence we get is thus strictly ascending. After the first iteration, the minimum of I_1 is the first value of Q. To update Q, we remove all values that have an index smaller than $l(I_2)$ and we add all values from mw(h) within $\max(r(I_1)+1, l(I_2))$ and $r(I_2)$ in the previously described way, which ensures that Q remains ascending. Continuing in this way ensures us that the desired minimum will always be at the front of the sequence at the end of each iteration, and the whole procedure can be carried out in $O(m)$ time.

Given a candidate H_{ij}, finding the value of the denominator in equation (3.7) now comes down to using our recursive formula twice. First in the way just described and then in its reverse direction, starting at hypothesis H_m instead of H_1, calculating the minimal weight needed to satisfy all implications with a right-boundary that is larger than a certain k, given that k itself will be in the rejection set. Adding mw$(i-1)$ from the original and mw$(j+1)$ from the reversed version will give the desired value. Note that hypotheses $i-1$ and $j+1$ always have to be included in the extended rejection set $\mathcal{S}$, because the parents of H_{ij} are implications which can only be satisfied by taking these hypotheses, given that no elements from $\mathcal{L}_{H_{ij}}$ can be included in $\mathcal{S}$.

By calculating both the forward and backward recurrence once for every hypothesis, we can furthermore calculate the minimal weight needed for the ratio of *every* candidate by just summing up two numbers. All ratios corresponding to rejection set $\mathcal{R}$ can thus simultaneously be calculated in linear time. Note that the same algorithm can be used when $\mathcal{R}$ is already congruent. All implications will then be leaf nodes and the extended set $\mathcal{S}$ will exactly equal $\mathcal{R}$.

3.3.2 Complexity

After having developed an algorithm that efficiently calculates ratios, we can implement the full iterative procedure, as given in formula (3.1). Starting from an empty rejection set $\mathcal{R}_0$, in each step we can identify which nodes are implications and which nodes are candidates, calculate the ratios and from that the significance levels for each candidate, and decide, based on the p-values, which candidate nodes can be rejected. These nodes are added to the current rejection set $\mathcal{R}_i$, to obtain $\mathcal{R}_{i+1}$, and we continue like this until no further rejections can be made. It might seem like we need to keep track of the full rejection set $\mathcal{R}_i$ in every step to be able to discover the new implication and candidate set, but we will show that the new set of implications and candidates, $\mathcal{I}_{i+1}$ and $\mathcal{C}_{i+1}$, can be calculated from the previous sets $\mathcal{I}_i$ and $\mathcal{C}_i$ and the set of newly rejected nodes $\mathcal{R}_{i+1} \setminus \mathcal{R}_i$. By keeping $\mathcal{I}$ and $\mathcal{C}$ sorted throughout the algorithm, the implementation can be done in $O(m)$ time. In terms of memory usage, also only $O(m)$ space is needed. The basic structure of the corresponding algorithm is given in Algorithm 1. In this algorithm we use the in section 3.2 introduced notation of(H) and an(H) to indicate the set of offspring and ancestor nodes of H and we introduce the notation pa(H) and ch(H) to indicate the parents and children of node H.

Algorithm 1 : The algorithm corresponding to the region procedure

```
Requires: $\alpha_{max}$ and a test to compute $p$-values

candidates := $\{H_{1m}\}$
implications := $\emptyset$
$\alpha := 0$

while candidates $\neq \emptyset$
    for each c in candidates
        compute ratio $r_c$
        compute pvalue $p_c$ if not done before
    /* find smallest $\alpha$ that rejects a candidate */
    $\alpha = \max(\alpha, \min_{c \in \text{candidates}}(p_c/r_c))$
    if $\alpha > \alpha_{max}$
        break
    rejected := {c $\in$ candidates | $p_c \leq r_c\alpha$}
    for each c in rejected
        adj_pvalues$_c$ := $\alpha$
    candidates := candidates $\setminus$ rejected
    candidates := candidates $\cup$ (ch(rejected) $\setminus$ of(candidates))
    implications := rejected $\cup$ (implications $\setminus$ pa(rejected))
return adj_pvalues
```

The most important part of the algorithm is the way in which the implication and candidate sets are updated. To update the implications, it is enough to note that every newly rejected node can never have rejected children and is thus a new implication by definition, whereas every old implication will stay one as long as none of its children got rejected.

To update the candidates, we should first note that every candidate that was not rejected will be a candidate in the next iteration. Furthermore, because of the new rejections, new nodes might have all their parents rejected. All children of newly rejected nodes are potential new candidates. To verify whether such a potential candidate indeed has only rejected parents, it suffices to check whether one of the previously unrejected candidates is an ancestor of this node. If this is the case, the node must have an unrejected parent, but otherwise, we can be sure that all its ancestors are rejected and the node should be included in $\mathcal{C}_{i+1}$.

From Algorithm 1 it is also clear how adjusted p-values are calculated. At the beginning, some $\alpha_{max} \leq 1$ is specified, which is the level on which the FWER will be controlled. Subsequently, in every iteration the minimal overall α-level needed to reject at least one candidate is determined, and the maximum over this and the previous level is taken to ensure that the α-level can only increase. This gives us the minimal α-value needed to reject this candidate (or candidates) and all its ancestors, i.e. its adjusted p-value. As long as this α-level stays below α_{max}, rejections will be made. If the level exceeds α_{max}, the procedure ends and all remaining adjusted p-values are set to 1.

Because all computations needed to update the current rejection set $\mathcal{R}_i$ to the new set $\mathcal{R}_{i+1}$ can be done in $O(m)$ time and there are $O(m^2)$ nodes in total, we know that the full procedure will have complexity $O(m^3)$. A number of factors are still of influence on the actual performance however. First of all, the complexity $O(m^3)$ is only guaranteed if the raw p-values are already calculated. When, instead of the p-values itself, a test function is provided, the algorithm will calculate a p-value for every new candidate. If the test function is fast, this will generally not have too much effect on the performance, and it can even be seen as an advantage to provide the test function instead of the p-values, because in this way only the necessary tests have to be performed. In some situations it can however be advisable to calculate all p-values in advance, because the time needed to perform all individual tests can become a large factor in the overall computation time. When the complexity of the test function itself exceeds $O(m)$, the full procedure will no longer run in $O(m^3)$. It will generally be possible to shorten the calculation time needed to obtain the p-values by using parallel computing. This will be the preferred approach when the chosen test function is time consuming.

A second factor that influences the performance of our multiple testing procedure is whether adjusted p-values are calculated. If adjusted p-values are calculated, this will often result in a multiple testing procedure in which only one or a few hypotheses are simultaneously rejected in every iteration. To speed up the procedure, one can choose to not calculate the adjusted p-values and to directly start the procedure at an α-level equal to α_{max}. In this way it will often happen that multiple nodes are rejected in every iteration, which will lower the overall number of iterations. Although the time complexity of this second method is, in the worst case, equal to the former procedure's complexity, the time gain will usually be considerable in practice.

In terms of memory usage, there is also an advantage in using a procedure that does not calculate adjusted p-values. If the adjusted p-values are not calculated, the output of our algorithm could be the final rejection set. However, because this rejection set is completely determined by its implications ($\mathcal{R} = \mathcal{I} \cup \text{an}(\mathcal{I})$), it is enough to return the final set of implications, which is more efficient in terms of memory. If adjusted p-values are calculated, it is also an option to only get the implications and their corresponding adjusted p-values returned, instead of the complete rejection set with associated adjusted p-values.

A last factor that influences the actual performance of our algorithm is the amount of signal that is present in the data. If there are only few nodes that can be rejected on a chosen α-level, the procedure will terminate shortly. However, the amount of signal in the data will usually not be known beforehand.

Even after precalculating the raw p-values, it is evident that our multiple testing procedure cannot be used for too large problems given the (worst-case) $O(m^3)$ complexity of the procedure. It will be difficult to specify the order of magnitude of the number of elementary hypotheses m for which the procedure will still be feasible, since this will depend on the factors already mentioned (whether adjusted p-values are needed, the amount of signal in the data) and on the exact specifications of the computer used. To give some indication, we would advise to only calculate adjusted p-values when the number of individual hypotheses is smaller than approximately 1000. If m greatly exceeds 10000, our region method will often be no longer feasible.

3.3.3 Confidence statements

In subsection 3.2.2, we discussed the construction of a confidence set for the number of false hypotheses in some chosen set $\mathcal{A}$, given a rejection set $\mathcal{R}$. Calculating the lower bound of this confidence set came down to calculating the minimum as given in equation (3.6).

As mentioned before, to extend $\mathcal{R}$ to a congruent set $\mathcal{S}$, we only have to satisfy all implications from the set $\mathcal{I} = \{I_1, \ldots, I_k\}$. To find the minimal number of false hypotheses in $\mathcal{A}$ we only have to consider those implications I_j that are fully contained in $\mathcal{A}$ (i.e. $\mathcal{L}_{I_j} \subseteq \mathcal{A}$), because all other implications can be satisfied without choosing any hypothesis from $\mathcal{A}$.

We will denote the set of implications that are fully contained in our set $\mathcal{A}$ by $\mathcal{I}'$. Calculating the quantity given in (3.6) now comes down to calculating the minimal number of hypotheses that are needed to satisfy all implications from $\mathcal{I}'$. This can be done efficiently by using the previously described algorithm that calculates the minimal weight needed to satisfy a given implication set. When we impose equal weights, this algorithm exactly calculates the minimal number of hypotheses needed to satisfy all these implications, or in other words, the minimal number of false hypotheses that have to lie in our set $\mathcal{A}$.

3.4 Applications

To illustrate our method, we apply it to both real and simulated data. The real data set we use comes from Carvalho et al. (2009) and consists of DNA copy number data (array CGH) of 68 colon tumors. These tumors have been classified as either "Adenoma" (33 tumors) or "Carcinoma" (35 tumors). For each chromosome we would like to know whether there are certain locations in which the DNA copy number differs between the adenoma and carcinoma tumors. To be able to test for differences we specify multiple logistic models per chromosome, in which tumor type is the dependent variable and regions of log2-based intensity ratios are used as predictors. In total we have copy number data for 4071 positions on the DNA, which results in logistic models that maximally contain a few hundred covariates each.

id	left-bound	right-bound	adj. p-value
1.	2	17	0.0445
2.	8	18	0.0450
3.	16	19	0.0436
⋮	⋮	⋮	⋮
77.	264	276	0.0455
78.	265	277	0.0463
79.	278	278	0.0318

Table 3.1: Summary of rejected regions on chromosome 13.

We use the global test (Goeman et al., 2004) to test all hypotheses of the form

$$H_{ij} \colon \beta_i = \ldots = \beta_j = 0,$$

in which β_i corresponds to the regression coefficient of the log2-based intensity ratio at the i^{th} position. With this data type, groups of covariates clearly correspond to physically meaningful entities. Because the covariates are measurements on specific locations on a chromosome, a group of neighboring covariates corresponds to a longer stretch of DNA. Both the exact locations in which the amount of DNA of the two tumor types differ as well as longer DNA stretches in which this occurs are of interest.

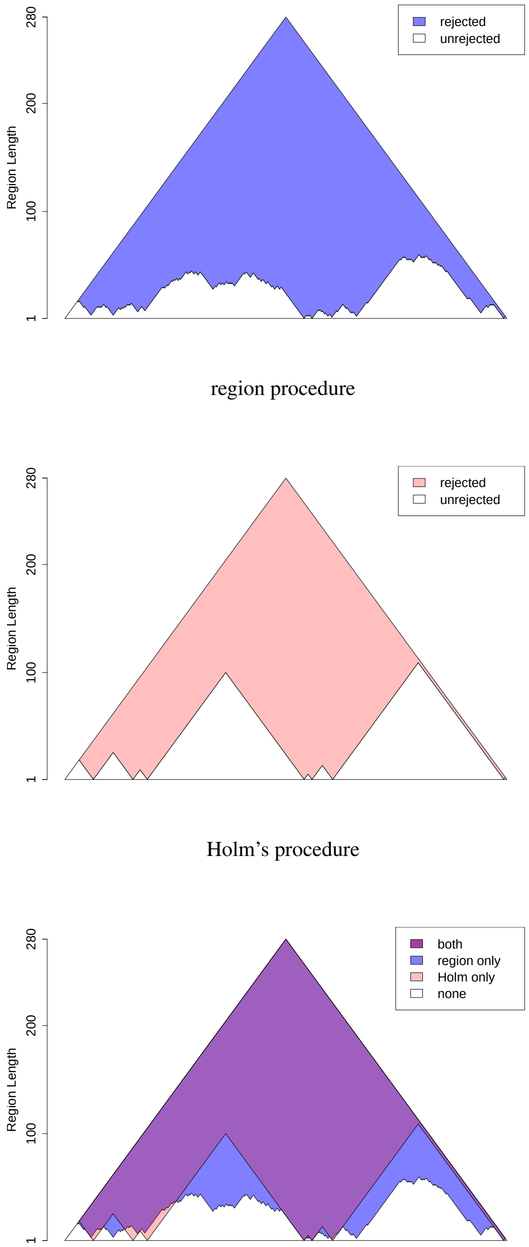
Because this application is mostly intended to illustrate how our method can be used, we will focus on chromosomes that are known to be associated with colorectal adenoma to carcinoma progression. Among others, this holds for chromosome 13 (Carvalho et al., 2009). On this chromosome, copy number ratios are available at 280 positions. After rejecting the global null-hypothesis $H_0 \colon \beta_1 = \ldots = \beta_{280} = 0$, our method tries to narrow down the exact locations on chromosome 13 in which copy number differences between the two tumor types occur. Controlling the FWER on $\alpha = 0.05$ results in many rejections which can be summarized in 79 implications (i.e. rejected regions that have no rejected child nodes). In Table 3.1, six of these implications are given together with their adjusted p-values. On the first row we see that the region ranging from covariate 2 to 17 was rejected, which means that the DNA copy number varies between the adenoma and carcinoma tumors on at least one but possibly on more of these 16 positions on the chromosome. Similar statements can be made on basis of the other rows.

Of all 79 implications, three correspond to elementary hypotheses, which means that our method could only locate three exact locations on the chromosome in which DNA copy number differs between the two tumor groups. These locations are location 152, 157 and 278. By comparison, if we would have used Holm's procedure on the elementary hypotheses only, we would have found 7 exact locations; the aforementioned as well as location 19, 44, 53 and 170. Even though Holm's method finds more exact positions that are associated with the outcome variable, this does not mean that the results obtained with our region procedure are less informative. First of all, although our method did not find location 19, 44, 53 and 170 exactly, it did find small significant regions containing these locations. The smallest identified regions containing these four locations were respectively region $[16, 19]$, region $[44, 50]$, region $[48, 55]$ and region $[169, 170]$. Furthermore, it did find other significant regions (small as well as larger ones) that Holm's procedure could not find. These regions give information on possible important DNA stretches that can explain tumor type differences as well.

In Figure 3.3 the findings of both the region procedure as well as Holm's procedure are visualized in two graphs that are similar to the one shown in Figure 3.1, but plotted in less detail because of their size. All possible regions, ranging from length 1 to length 280, are plotted, and are depicted in color when they could be rejected at an overall α level of 0.05 and are left blank otherwise. From the third graph, that summarizes the results of the two methods in one plot, we see that our region method does approximately locate the same regions as Holm's procedure (although not always as specifically) and finds in addition other regions of potential importance. The total number of region hypotheses that Holm's procedure rejected equals 28113, whereas our method could reject 32793 region hypotheses. Since all region hypotheses are hypotheses of interest, the total number of rejections made is a relevant performance measure, and from this measure we can conclude that our region procedure has an advantage over Holm's method on this data set. Furthermore, even though we can only locate three influential covariates exactly, following the reasoning of subsection 3.2.2, we can calculate the minimal number of influential covariates that have to be present in the full set of 280. We find that we must have at least 19 of these influential covariates, which is more than the 7 locations found by using Holm's procedure.

On chromosome 13, there seem to be multiple interesting stretches of DNA that cannot always be narrowed down to one exact location. This is a situation in which our method can be very valuable, because it does not only look for exact locations and can benefit from small neighboring effects that can be detected together. However, when the elementary hypotheses itself are very significant, Holm's method will often detect as much as our method. This happens in chromosome 20 for example. On this chromosome we have 228 locations on which the DNA copy number variation is measured and Holm's method is able to find 190 of these to be significantly associated with the outcome. The results of our method are very comparable (at least 188 influential covariates, of which 187 are exactly located), but there's no advantage in using it. A note we can make here is that the test we use might also not be the best choice for this situation because it is especially designed for detecting small non-sparse effects. As already described in section 3.2, our method equals Holm when all intersection hypotheses are tested with simple Bonferroni combinations, and the fact that Holm outperforms our method in this specific situation thus also points out that our choice for the global test is a suboptimal one in this situation.

To get a better idea of the performance of our method compared to Holm's procedure, we also performed a simulation study. In our simulations, we use n-dimensional outcome



region procedure

Holm's procedure

Difference between the two procedures

Figure 3.3: Given the 280 copy number variation measurements on chromosome 13, the region graphs indicate the spots on the chromosome where differences in copy number between the two tumor types are to be expected. All significant regions, as found by two different multiple testing procedures on an overall α-level of 0.05, are depicted in color.

vectors $\boldsymbol{y}$ that depend on a pre-specified number l of "latent" variables $\boldsymbol{z}_i$ and a random term $\boldsymbol{u}$ in the following way:

$$\boldsymbol{y} = \sum_{i=1}^{l} \beta \boldsymbol{z}_i + \boldsymbol{u},$$

where β is a scalar and each latent vector $\boldsymbol{z}_i$ as well as the error term $\boldsymbol{u}$ are n-dimensional standard-normal vectors. The actual data matrix X of dimension n times p is then designed to contain both influential and noise components. Each latent variable $\boldsymbol{z}_i$, with $1 \leq i \leq l$ is used to create k influential covariates $\boldsymbol{x}_{ij}$, with $1 \leq j \leq k$, where the i indicates the corresponding latent variable:

$$\boldsymbol{x}_{ij} = \boldsymbol{z}_i + \boldsymbol{\epsilon}_{ij},$$

where $\boldsymbol{\epsilon}_{ij}$ is again standard normally distributed. This results in l groups of k correlated covariates. Subsequently, these groups are approximately equidistantly distributed over the p columns of the design matrix X. The remaining columns are filled by randomly generated variables v_i with $v_i \sim \mathcal{N}(0, 1)$. The matrix X thus contains $l \times k$ influential and $p - l \times k$ uninfluential columns.

Two important factors that can be varied are the number of variables k per group and the effect size (as given by $\boldsymbol{\beta}$). With 10 underlying variables $\boldsymbol{z}_i$, multiple group sizes k and multiple values for $\boldsymbol{\beta}$, we performed 1000 simulations per setting. The number of overall covariates p and the number of subjects n were both set to 100. The test we used was again the global test and the FWER was controlled on $\alpha = 0.05$.

In every setting we kept track of five outcome measures, namely the total number of region hypotheses that could be rejected by our procedure and by Holm's method as well as the total number of elementary hypotheses that could be rejected by those two procedures. In addition, we calculated the lower bound of the 95% confidence set for the number of influential covariates by using the final rejection set of our region procedure.

In Table 3.2 the outcomes of these experiments are given. Every number is the mean over 1000 simulations. The number in between brackets is the corresponding standard error. We see the numbers go up when the groups get larger (which of course means that there is also more to find) and when $\boldsymbol{\beta}$ increases. If we compare column 5 and 7, we see that our procedure is always able to reject more region hypotheses. However, our method is worse in specifying exactly which variables are influential. In every situation, Holm's method outperforms our method in this respect, although the differences are small. Still, our method is able to make confidence statements about the expected number of influential variables, which indicate that there are at least twice as many influential variables than detected by Holm's method in every situation. It will depend on the situation whether statements about the true number of underlying influential variables in a certain region will be preferred over a smaller but more informative number of truly detected influential variables. We repeated the procedure with 5 instead of 10 latent variables and could conclude the same.

In conclusion, our method is good at discovering whether a data set contains influential variables and at finding the areas where these variables will approximately be located.

10 latent variables						
		region procedure			Holm's procedure	
beta	group size	el.hyps implied	el.hyps rejected	reg.hyps rejected	el.hyps rejected	reg.hyps rejected
0.2	1	0.61 (0.02)	0.04 (0.006)	415 (22)	0.15 (0.01)	254 (22)
	3	1.74 (0.04)	0.12 (0.01)	1532 (33)	0.33 (0.02)	505 (30)
	5	2.54 (0.05)	0.20 (0.02)	1979 (34)	0.59 (0.03)	807 (35)
	7	2.87 (0.06)	0.24 (0.02)	2197 (32)	0.74 (0.03)	944 (37)
0.5	1	2.05 (0.03)	0.31 (0.02)	1809 (32)	0.59 (0.02)	895 (36)
	3	5.72 (0.06)	1.19 (0.04)	3434 (16)	1.64 (0.05)	1851 (40)
	5	8.32 (0.08)	2.07 (0.06)	3775 (11)	2.85 (0.07)	2380 (37)
	7	10.19 (0.10)	2.76 (0.08)	3923 (10)	3.89 (0.09)	2708 (36)
1	1	2.70 (0.03)	0.58 (0.02)	2364 (28)	0.93 (0.03)	1333 (39)
	3	7.68 (0.06)	2.15 (0.05)	3811 (11)	2.61 (0.06)	2474 (37)
	5	11.30 (0.08)	3.73 (0.08)	4083 (7.6)	4.55 (0.08)	3044 (30)
	7	14.29 (0.12)	5.11 (0.11)	4225 (7.1)	6.30 (0.11)	3332 (27)
5	1	2.98 (0.04)	0.71 (0.03)	2578 (25)	1.10 (0.03)	1508 (40)
	3	8.49 (0.06)	2.67 (0.06)	3930 (9.1)	3.14 (0.06)	2742 (33)
	5	12.69 (0.09)	4.65 (0.09)	4183 (6.8)	5.43 (0.09)	3272 (26)
	7	16.19 (0.12)	6.41 (0.12)	4322 (5.9)	7.57 (0.12)	3555 (23)

Table 3.2: Comparison region method & Holm, based on 1000 simulations per case. The third column indicates the lower bound of the 95% confidence set for the number of false elementary hypotheses, based on the total rejection set of our region method. The fourth and sixth column indicate how many of these elementary hypotheses could be located exactly by respectively the region method and Holm's method. The fifth and seventh column indicate how many region hypotheses were rejected in total by respectively the region method and Holm's method. All numbers are means over 1000 simulations. The numbers in brackets are the corresponding standard errors.

Nevertheless, when the exact locations of the influential covariates are of main interest, Holm might perform just as well (and faster). However, there are situations possible in which only groups of covariates *can* be detected. If we would have a regression setting in which p exceeds n and our hypotheses would state that a certain covariate or group of covariates is not associated with the response *given the remaining covariates*, a single covariate will never get detected. Groups of covariates on the other hand can be identifiable. In such a situation, our method would work, but Holm's method could only be used after specifying all possible interesting sets in advance. Without a clear idea of the exact signal in the data, this could lead to many groups and a severe multiple testing correction. Our method does not need a pre-specification of the interesting sets and uses the structure of the data to find them itself.

3.5 Discussion

We have presented a multiple testing method that tests all possible region hypotheses, corresponding to a set of ordered elementary hypotheses, in a hierarchical way. Because of the method's top-down approach the multiple testing burden is reduced and smaller regions are only tested when association with the outcome variable is expected based on significance of larger regions they are contained in. Given that failing to reject a certain region means that all the regions it embeds can never be tested, the choice to make the α-distribution proportional to the region lengths is intuitive. In addition, the specific structure in the hypotheses enables us to reduce the multiple testing burden further by using information on restricted combinations, i.e. by constructing congruent sets before distributing the α. Our method strongly controls the FWER, independently of the underlying correlation structure in the hypotheses and can be used with every valid hypothesis test. It is even possible to use different hypothesis tests for different regions. If one would suspect that a certain test statistic is most powerful on higher levels, whereas another is more powerful for the smaller regions, both tests could be used in different places. Furthermore, even if the final rejection set does not include (many) elementary hypotheses, our method enables us to derive statements on the minimal number of elementary hypotheses in arbitrary regions or other sets that have to be false.

In this article, we looked at a specific α-rule, which only attributes parts of the overall α to nodes that have all their parent nodes rejected. It would be interesting to also look at rules in which nodes can already be considered for testing if only one of their parents is rejected. Such rules will probably have different behavior than the one we chose, but will unfortunately be more difficult to formally state and program.

Apart from extending the current region framework, it could also be very useful to try to apply this methodology to problems in which the local hypotheses do not necessarily have to be regions, but can be more arbitrary sets of elementary hypotheses. This would lead us into the direction of directed acyclic graphs, a family of graphs of which the region-graph is a special case. The sequential rejection principle can in principle be used to construct multiple testing procedures and their corresponding α-rules for every set of hypotheses that gives rise to certain logical relationships. However, different algorithms will be required for each set of hypotheses in order to be able to efficiently calculate the α-values on which these hypotheses can be tested. In this article we described an efficient algorithm to calculate the α-levels for region hypotheses, but this algorithm is specifically designed for the logical structure as imposed by the region hypotheses, and different techniques will be needed in order to find efficient algorithms for multiple testing problems with a different structure.

3.6 Appendix

Theorem 3.6.1. *The region procedure strongly controls the FWER.*

Proof. Recall that the significance levels allocated to each hypothesis H in the region procedure are specified in the following way:

$$\alpha_H(\mathcal{R}) = \alpha \times r_H(\mathcal{R}),$$

where the ratio $r_H(\mathcal{R})$ is given by

$$r_H(\mathcal{R}) = \begin{cases} \min\limits_{\mathcal{S}\in\Phi:\ \mathcal{R}\subseteq\mathcal{S}, H\notin\mathcal{S}} \frac{\sum\limits_{H_i\in\mathcal{L}_H} w_i}{\sum\limits_{H_i\in\mathcal{L}\setminus\mathcal{S}} w_i} & \text{if pa}(H)\subseteq\mathcal{R} \text{ and } H\notin\mathcal{R} \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where Φ denotes the set of all congruent rejections sets.

In order to show that this procedure strongly controls the FWER, it suffices to show that the two conditions, as imposed by the SRP, hold. First, we will check the monotonicity condition:
Let $\mathcal{R}$ be a rejection set, $\mathcal{R}'\supseteq\mathcal{R}$ an arbitrary extension and $H\notin\mathcal{R}'$ an hypothesis.
If $r_H(\mathcal{R}) = 0$, then clearly $r_H(\mathcal{R}) \leq r_H(\mathcal{R}')$. Otherwise, we know that pa$(H)\subseteq\mathcal{R}\subseteq\mathcal{R}'$. Hence

$$\begin{aligned} r_H(\mathcal{R}) &= \min_{\mathcal{S}\in\Phi:\ \mathcal{R}\subseteq\mathcal{S}, H\notin\mathcal{S}} \frac{\sum\limits_{H_i\in\mathcal{L}_H} w_i}{\sum\limits_{H_i\in\mathcal{L}\setminus\mathcal{S}} w_i} \\ &\leq \min_{\mathcal{S}\in\Phi:\ \mathcal{R}'\subseteq\mathcal{S}, H\notin\mathcal{S}} \frac{\sum\limits_{H_i\in\mathcal{L}_H} w_i}{\sum\limits_{H_i\in\mathcal{L}\setminus\mathcal{S}} w_i} = r_H(\mathcal{R}'), \end{aligned}$$

from which it follows that (3.2) holds.

To show that the single step condition holds, first note that the first case in equation (3.8) reduces to equation (3.7) in case of a congruent rejection set $\mathcal{R}$. Given that the rejection set $\mathcal{R}$ is congruent, the following holds:

$$\sum_{H \in \mathcal{H} \setminus \mathcal{R}} r_H(\mathcal{R}) = \sum_{H \in \mathcal{H} \setminus \mathcal{R} \colon \, \mathrm{pa}(H) \subseteq \mathcal{R}} \frac{\sum\limits_{H_i \in \mathcal{L}_H} w_i}{\sum\limits_{H_i \in \mathcal{L} \setminus \mathcal{R}} w_i}$$

$$= \frac{\sum\limits_{H_i \in \mathcal{L}_H \colon \, H \in \mathcal{H} \setminus \mathcal{R}, \mathrm{pa}(H) \subseteq \mathcal{R}} w_i}{\sum\limits_{H_i \in \mathcal{L} \setminus \mathcal{R}} w_i} = 1$$

from which it follows that (3.3) holds.

In the last step, we used that whenever the rejection set $\mathcal{R}$ is congruent, candidates cannot have leaf nodes in common. Indeed, if this could happen, we would have distinct candidates H_{ij} and H_{kl} with $i \leq k \leq j \leq l$, from which it follows that H_{il} is also a region hypothesis. Because H_{ij} and H_{kl} are candidates, we know that all their ancestor nodes have to be rejected, including this hypothesis H_{il}. Because H_{il} is rejected and $\mathcal{R}$ is congruent, we have to have one of the hypotheses of $\mathcal{L}_{H_{il}}$ in $\mathcal{R}$, but we know that none of the hypotheses from $\mathcal{L}_{H_{ij}}$ and $\mathcal{L}_{H_{kl}}$ can be in $\mathcal{R}$, since H_{ij} and H_{kl} are candidates, which leads to a contradiction. □

Theorem 3.6.2. *Region procedure with Bonferroni local test is equivalent to Holm's procedure.*

Proof. Say we have m elementary hypotheses $H_1, \ldots, H_m$ with corresponding raw p-values $p_1, \ldots, p_m$, and suppose we are using our region procedure in a situation where the rejection set $\mathcal{R}$ is congruent and contains r elementary hypotheses. In such a situation, each candidate H_I gets an α-level α_I of $\frac{|I|}{m-r}\alpha$. To be able to reject any H_I with a Bonferroni test on this level, we need the following:

$$\exists i \in I \colon p_i \leq \frac{\alpha}{m-r}. \tag{3.9}$$

As soon as we have such a p_i and corresponding H_i, we cannot only reject the candidate H_I with $i \in I$, but also all smaller region hypotheses H_J with $i \in J$, because these get α-levels $\alpha_J \geq \frac{|J|}{m-r}\alpha$ and can for that reason again be rejected because of containing H_i. This means that H_i itself can be rejected as well, and that our rejection set $\mathcal{R}$ is again congruent. Given r already rejected elementary hypotheses, the region procedure in combination with a Bonferroni test can thus reject an elementary hypothesis if and only if we meet condition (3.9). The exact same criterion holds for Holm's procedure which shows their equivalence. □