University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2004 *Paper* 164

Multiple Testing Procedures: R multtest Package and Applications to Genomics

Katherine S. Pollard* Sandrine Dudoit†

Mark J. van der Laan‡

*Center for Biomolecular Science and Engineering, University of California, Santa Cruz, kpollard@gladstone.ucsf.edu

†Division of Biostatistics, School of Public Health, University of California, Berkeley, sandrine@stat.berkeley.edu

‡Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@berkeley.edu

Multiple Testing Procedures: R multtest Package and Applications to Genomics

Katherine S. Pollard, Sandrine Dudoit, and Mark J. van der Laan

Abstract

The Bioconductor R package multtest implements widely applicable resampling-based single-step and stepwise multiple testing procedures (MTP) for controlling a broad class of Type I error rates, in testing problems involving general data generating distributions (with arbitrary dependence structures among variables), null hypotheses, and test statistics. The current version of multtest provides MTPs for tests concerning means, differences in means, and regression parameters in linear and Cox proportional hazards models. Procedures are provided to control Type I error rates defined as tail probabilities for arbitrary functions of the numbers of false positives and rejected hypotheses. These error rates include tail probabilities for the number of false positives (generalized family-wise error rate, gFWER) and the proportion of false positives among the rejected hypotheses (TPPFP). Single-step and step-down common-cut-off (maxT) and common-quantile (minP) procedures, that take into account the joint distribution of the test statistics, are proposed to control the family-wise error rate (FWER), or chance of at least one Type I error. In addition, augmentation multiple testing procedures are provided to control the gFWER and TPPFP, based on any initial FWER-controlling procedure. The results of a multiple testing procedure can be summarized using rejection regions for the test statistics, confidence regions for the parameters of interest, or adjusted p-values. A key ingredient of our proposed MTPs is the test statistics null distribution (and estimator thereof) used to derive rejection regions and corresponding confidence regions and adjusted p-values. Both bootstrap and permutation estimators of the test statistics null distribution are available. The S4 class/method object-oriented programming approach was adopted to summarize the results of a MTP. The modular design of multtest allows interested users to readily extend the package's functionality. Typical testing scenarios are illustrated by applying various MTPs implemented in multtest to the Acute Lymphoblastic Leukemia (ALL)

dataset of Chiaretti et al. (2004), with the aim of identifying genes whose expression measures are associated with (possibly censored) biological and clinical outcomes.

Note. This document is an expanded version of a chapter to be published in the monograph *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* [Gentleman et al., 2005]. The reader is referred to this book for a discussion of other relevant packages developed as part of the Bioconductor project.

0.1 Introduction

0.1.1 Motivation

Current statistical inference problems in biomedical and genomic data analysis routinely involve the simultaneous test of thousands, or even millions, of null hypotheses. Examples include:

- identification of differentially expressed genes in microarray experiments, i.e., genes whose expression measures are associated with possibly censored responses or covariates;
- tests of association between gene expression measures and Gene Ontology (GO) annotation;
- identification of transcription factor binding sites in ChIP-Chip experiments [Keleş et al., 2004];
- genetic mapping of complex traits using single nucleotide polymorphisms (SNP).

The above testing problems share the following general characteristics:

- inference for high-dimensional multivariate distributions, with complex and unknown dependence structures among variables;
- broad range of parameters of interest, such as, regression coefficients and correlations;
- many null hypotheses, in the thousands or even millions;
- complex dependence structures among test statistics.

Motivated by these applications, we have developed and implemented (in R and SAS) resampling-based single-step and stepwise multiple testing procedures (MTP) for controlling a broad class of Type I error rates, in testing problems involving general data generating distributions (with arbitrary dependence structures among variables), null hypotheses (defined in terms of submodels for the data generating distribution), and test statistics (e.g., t-statistics, F-statistics). The different components of our multiple testing methodology are treated in detail in a collection of related articles [Pollard and van der Laan, 2004, Dudoit et al., 2004, van der Laan et al., 2004,,

Dudoit et al., 2004] and a book in preparation [Dudoit and van der Laan, 2004].

The early article of Pollard and van der Laan [2004] and subsequent article of Dudoit et al. [2004] establish a general statistical framework for multiple hypothesis testing. A key feature of the proposed MTPs is the *test statistics null distribution* (rather than data generating null distribution) used to derive rejection regions (i.e., cut-offs) for the test statistics and resulting confidence regions and adjusted p-values. For Type I error rates defined as arbitrary parameters $\theta(F_{V_n})$ of the distribution of the number of Type I errors V_n (e.g., the generalized family-wise error rate, $gFWER(k) = Pr(V_n > k)$, or chance of at least $(k+1)$ false positives), this null distribution is the asymptotic distribution of the vector of null value shifted and scaled test statistics. Resampling procedures (e.g., based on the non-parametric or model-based bootstrap) are proposed to conveniently obtain consistent estimators of the null distribution and the corresponding test statistic cut-offs and adjusted p-values [Pollard and van der Laan, 2004, Dudoit et al., 2004, van der Laan et al., 2004, Dudoit and van der Laan, 2004].

Pollard and van der Laan [2004] and Dudoit et al. [2004] also derive *single-step common-cut-off and common-quantile procedures* for controlling general Type I error rates of the form $\theta(F_{V_n})$.

van der Laan et al. [2004] focus on control of the family-wise error rate, $FWER = gFWER(0)$, and provide *step-down common-cut-off and common-quantile procedures*, based on maxima of test statistics (maxT) and minima of unadjusted p-values (minP), respectively. van der Laan et al. [2004], and subsequently Dudoit et al. [2004] and Dudoit and van der Laan [2004], propose general classes of *augmentation multiple testing procedures* (AMTP), obtained by adding suitably chosen null hypotheses to the set of null hypotheses already rejected by an initial MTP. In particular, given *any* FWER-controlling procedure, they show how one can trivially obtain augmentation procedures controlling tail probabilities for the number (gFWER) and proportion (TPPFP) of false positives among the rejected hypotheses. The results of a simulation study comparing augmentation procedures to existing gFWER- and TPPFP-controlling MTPs are reported in Dudoit et al. [2004]. Finally, the multiple testing methodology and applications to genomic data analysis are the subject of a book in preparation for Springer [Dudoit and van der Laan, 2004].

In order to make this general methodology broadly and readily accessible to the biomedical and genomic data analysis community, we have implemented the above MTPs in the Bioconductor R package *multtest*, which is the subject of the present paper.

0.1.2 Outline

The Bioconductor R package *multtest* provides software implementations of the multiple testing procedures mentioned in Section 0.1.1 and discussed in greater detail in Section 0.2. Specifically, given a multivariate dataset and user-supplied choices for the test statistics, Type I error rate and its target level(s), estimator of the test statistics null distribution, and procedure for deriving rejection regions, the main user-level function `MTP` returns unadjusted and adjusted p-values, cut-off vectors for the test statistics, and estimates and confidence regions for the parameters of interest. Both bootstrap and permutation estimators of the test statistics null distribution are available and can optionally be output to the user. The S4 class/method object-oriented programming approach was adopted to represent the results of a MTP. Several methods are defined to produce numerical and graphical summaries of these results. A modular programming approach, which uses function closures, allows interested users to readily extend the package's functionality, by inserting functions for new test statistics and testing procedures.

The present paper is organized as follows. Section 0.2 provides a summary of our proposed multiple testing procedures. Section 0.3 discusses their software implementation in the Bioconductor R package *multtest*. Section 0.4 describes applications of the MTPs to the Acute Lymphoblastic Leukemia (ALL) dataset of Chiaretti et al. [2004], with the aim of identifying genes whose expression measures are associated with (possibly censored) biological and clinical outcomes such as: tumor cellular subtype (B-cell vs. T-cell), tumor molecular subtype (BCR/ABL, NEG, ALL1/AF4, E2A/PBX1, p15/p16, NUP-98), and time to relapse. Finally, Section 0.5 discusses ongoing efforts.

0.2 Multiple hypothesis testing methodology

0.2.1 Multiple hypothesis testing framework

Hypothesis testing is concerned with using observed data to test hypotheses, i.e., make decisions, regarding properties of the unknown data generating distribution. For example, microarray experiments might be conducted on a sample of patients in order to identify genes whose expression levels are associated with survival. Below, we discuss in turn the main ingredients of a multiple testing problem. These include data, null and alternative hypotheses, test statistics, multiple testing procedure, Type I and Type II errors, adjusted p-values, test statistics null distribution, rejection regions. Further detail on each of these components can be found in Dudoit et al. [2004] and Dudoit and van der Laan [2004]; specific proposals of MTPs are

given in Sections 0.2.4 – 0.2.6.

Software implementation. The *multtest* package is designed so that the main components of a MTP are specified as arguments to the package's primary user-level function, `MTP`: the data via the arguments `X`, `W`, `Y`, `Z`, `Z.incl`, and `Z.test`; the test statistics via `test`, `robust`, `standardize`, `alternative`, and `psi0`; the Type I error rate via `typeone` and `alpha` (and also the error rate-specific parameters `k` and `q`); the test statistics null distribution via `nulldist` and `B`; and the MTP itself via `method`. The desired output, i.e., adjusted p-values, rejection regions, and confidence regions, are specified using the arguments `get.adjp`, `get.cutoff`, and `get.cr`, respectively. The main steps in applying a multiple testing procedure are listed in the flowchart of Table 1 and typical testing scenarios are illustrated in Section 0.4, using the ALL dataset of Chiaretti et al. [2004] as a case study.

Data. Let $X_1, \ldots, X_n$ be a *random sample* of n independent and identically distributed (i.i.d.) random variables, $X \sim P \in \mathcal{M}$, where the *data generating distribution* P is an element of a particular *statistical model* $\mathcal{M}$ (i.e., a set of possibly non-parametric distributions). In a microarray experiment, for example, X is a vector of gene expression measurements, which we observe for each of n arrays.

Null and alternative hypotheses. In order to cover a broad class of testing problems, define M null hypotheses in terms of a collection of *submodels*, $\mathcal{M}(m) \subseteq \mathcal{M}$, $m = 1, \ldots, M$, for the data generating distribution P. The M *null hypotheses* are defined as $H_0(m) \equiv \mathrm{I}(P \in \mathcal{M}(m))$ and the corresponding *alternative hypotheses* as $H_1(m) \equiv \mathrm{I}(P \notin \mathcal{M}(m))$. In many testing problems, the submodels concern *parameters*, i.e., functions of the data generating distribution P, $\Psi(P) = \psi = (\psi(m) : m = 1, \ldots, M)$, such as means, differences in means, correlation coefficients, and regression parameters. For instance, the full model $\mathcal{M}$ might refer to the set of all continuous M–variate distributions and the submodel $\mathcal{M}(m)$, corresponding to the mth null hypothesis, might be the subset of $\mathcal{M}$ for which the mth component of the mean vector $\psi = E[X]$ is non-negative, i.e., $\mathcal{M}(m) = \{P \in \mathcal{M} : \psi(m) \geq 0\}$ (further parametric restrictions, such as normality, may be imposed on the models). One distinguishes between two types of testing problems: *one-sided tests*, where $H_0(m) = \mathrm{I}(\psi(m) \leq \psi_0(m))$, and *two-sided tests*, where $H_0(m) = \mathrm{I}(\psi(m) = \psi_0(m))$. The user-supplied hypothesized *null values*, $\psi_0(m)$, are frequently zero.

Let $\mathcal{H}_0 = \mathcal{H}_0(P) \equiv \{m : H_0(m) = 1\} = \{m : P \in \mathcal{M}(m)\}$ be the set of $h_0 \equiv |\mathcal{H}_0|$ true null hypotheses, where we note that $\mathcal{H}_0$ depends on the data generating distribution P. Let $\mathcal{H}_1 = \mathcal{H}_1(P) \equiv \mathcal{H}_0^c(P) = \{m : H_1(m) = 1\} = \{m : P \notin \mathcal{M}(m)\}$ be the set of $h_1 \equiv |\mathcal{H}_1| = M - h_0$ false null hypotheses, i.e., true positives. The goal of a multiple testing procedure

is to accurately estimate the set $\mathcal{H}_0$, and thus its complement $\mathcal{H}_1$, while controlling probabilistically the number of false positives.

Test statistics. A testing procedure is a *data-driven* rule for deciding whether or not to *reject* each of the M null hypotheses $H_0(m)$, i.e., declare that $H_0(m)$ is false (zero) and hence $P \notin \mathcal{M}(m)$. The decisions to reject or not the null hypotheses are based on an M–vector of *test statistics*, $T_n = (T_n(m) : m = 1, \ldots, M)$, that are functions $T_n(m) = T(X_1, \ldots, X_n)(m)$ of the data, $X_1, \ldots, X_n$. Denote the typically unknown (finite sample) *joint distribution* of the test statistics T_n by $Q_n = Q_n(P)$.

Single-parameter null hypotheses are commonly tested using *t-statistics*, i.e., standardized differences,

$$T_n(m) \equiv \frac{\text{Estimator} - \text{Null value}}{\text{Standard error}} = \sqrt{n}\,\frac{\psi_n(m) - \psi_0(m)}{\sigma_n(m)}. \tag{1}$$

In general, the M–vector $\psi_n = (\psi_n(m) : m = 1, \ldots, M)$ denotes an asymptotically linear *estimator* of the parameter M–vector $\psi = (\psi(m) : m = 1, \ldots, M)$ and $(\sigma_n(m)/\sqrt{n} : m = 1, \ldots, M)$ denote consistent estimators of the *standard errors* of the components of ψ_n. For tests of means, one recovers the usual one-sample and two-sample t-statistics, where $\psi_n(m)$ and $\sigma_n(m)$ are based on empirical means and variances, respectively (e.g., two-sample t-statistic in Equation (3), p. vi, for the ALL microarray data analysis of Section 0.4). In some settings, it may be appropriate to use (unstandardized) *difference statistics*, $T_n(m) \equiv \sqrt{n}(\psi_n(m) - \psi_0(m))$ [Pollard and van der Laan, 2004]. Test statistics for other types of null hypotheses include F-statistics, χ^2-statistics, and likelihood ratio statistics.

Multiple testing procedure. A *multiple testing procedure* (MTP) provides *rejection regions*, $\mathcal{C}_n(m)$, i.e., sets of values for each test statistic $T_n(m)$ that lead to the decision to reject the null hypothesis $H_0(m)$. In other words, a MTP produces a random (i.e., data-dependent) subset $\mathcal{R}_n$ of rejected hypotheses that estimates $\mathcal{H}_1$, the set of true positives,

$$\mathcal{R}_n = \mathcal{R}(T_n, Q_{0n}, \alpha) \equiv \{m : H_0(m) \text{ is rejected}\} = \{m : T_n(m) \in \mathcal{C}_n(m)\}, \tag{2}$$

where $\mathcal{C}_n(m) = \mathcal{C}(T_n, Q_{0n}, \alpha)(m)$, $m = 1, \ldots, M$, denote possibly random rejection regions. The long notation $\mathcal{R}(T_n, Q_{0n}, \alpha)$ and $\mathcal{C}(T_n, Q_{0n}, \alpha)(m)$ emphasizes that the MTP depends on: (i) the *data*, $X_1, \ldots, X_n$, through the M–vector of *test statistics*, $T_n = (T_n(m) : m = 1, \ldots, M)$; (ii) a (estimated) test statistics *null distribution*, Q_{0n}, for deriving rejection regions for each $T_n(m)$ and the resulting adjusted p-values (Section 0.2.2); and (iii) the *nominal level* α, i.e., the desired upper bound for a suitably defined Type I error rate.

Unless specified otherwise, it is assumed that large values of the test statistic $T_n(m)$ provide evidence against the corresponding null hypoth-

esis $H_0(m)$, that is, we consider rejection regions of the form $\mathcal{C}_n(m) = (c_n(m), \infty)$, where $c_n(m)$ are to-be-determined *critical values*, or *cut-offs*, computed under the null distribution Q_{0n} for the test statistics T_n (Section 0.2.2).

Example. Suppose that, as in the analysis of the ALL dataset of Chiaretti et al. [2004] (Section 0.4), one is interested in identifying genes that are differentially expressed in two populations of ALL cancer patients, those with the B-cell subtype and those with the T-cell subtype. The data consist of random J–vectors X, where the first M entries of X are microarray expression measures on M genes of interest and the last entry, $X(J)$, is an indicator for the ALL subtype (1 for B-cell, 0 for T-cell). Then, the parameter of interest is an M–vector of differences in mean expression measures in the two populations, $\psi(m) = E[X(m)|X(J) = 1] - E[X(m)|X(J) = 0]$, $m = 1, \ldots, M$. To identify genes with higher mean expression measures in the B-cell compared to T-cell ALL subjects, one can test the one-sided null hypotheses $H_0(m) = \mathrm{I}(\psi(m) \leq 0)$ vs. the alternative hypotheses $H_1(m) = \mathrm{I}(\psi(m) > 0)$, using two-sample Welch t-statistics

$$T_n(m) \equiv \frac{\bar{X}_{1,n_1}(m) - \bar{X}_{0,n_0}(m)}{\sqrt{n_0^{-1}(m)\sigma^2_{0,n_0}(m) + n_1^{-1}(m)\sigma^2_{1,n_1}(m)}}, \tag{3}$$

where $n_k(m)$, $\bar{X}_{k,n_k}(m)$, and $\sigma^2_{k,n_k}(m)$ denote, respectively, the sample sizes, sample means, and sample variances, for patients with tumor subtype k, $k = 0, 1$. The null hypotheses are rejected, i.e., the corresponding genes are declared differentially expressed, for large values of the test statistics $T_n(m)$.

Type I and Type II errors. In any testing situation, two types of errors can be committed: a *false positive*, or *Type I error*, is committed by rejecting a true null hypothesis, and a *false negative*, or *Type II error*, is committed when the test procedure fails to reject a false null hypothesis. The situation can be summarized by Table 2, below, where the number of Type I errors is $V_n \equiv |\mathcal{R}_n \cap \mathcal{H}_0| = \sum_{m \in \mathcal{H}_0} \mathrm{I}(T_n(m) \in \mathcal{C}_n(m))$ and the number of Type II errors is $U_n \equiv |\mathcal{R}_n^c \cap \mathcal{H}_1| = \sum_{m \in \mathcal{H}_1} \mathrm{I}(T_n(m) \notin \mathcal{C}_n(m))$.

Note that both U_n and V_n depend on the unknown data generating distribution P through the unknown set of true null hypotheses $\mathcal{H}_0 = \mathcal{H}_0(P)$. The numbers $h_0 = |\mathcal{H}_0|$ and $h_1 = |\mathcal{H}_1| = M - h_0$ of true and false null hypotheses are *unknown parameters*, the number of rejected hypotheses $R_n \equiv |\mathcal{R}_n| = \sum_{m=1}^{M} \mathrm{I}(T_n(m) \in \mathcal{C}_n(m))$ is an *observable random variable*, and the entries in the body of the table, U_n, $h_1 - U_n$, V_n, and $h_0 - V_n$, are *unobservable random variables* (depending on P through $\mathcal{H}_0(P)$).

Ideally, one would like to simultaneously minimize both the number of Type I errors and the number of Type II errors. Unfortunately, this is not feasible and one seeks a *trade-off* between the two types of errors. A stan-

dard approach is to specify an acceptable level α for the Type I error rate and derive testing procedures, i.e., rejection regions, that aim to minimize the Type II error rate, i.e., maximize *power*, within the class of procedures with Type I error rate at most α.

Type I error rates. When testing multiple hypotheses, there are many possible definitions for the Type I error rate and power of a test procedure. Accordingly, we adopt the general framework proposed in Dudoit et al. [2004] and Dudoit and van der Laan [2004], and define Type I error rates as *parameters*, $\theta_n = \theta(F_{V_n,R_n})$, of the joint distribution F_{V_n,R_n} of the numbers of Type I errors V_n and rejected hypotheses R_n. Such a general representation covers the following commonly-used Type I error rates.

Generalized family-wise error rate (gFWER), or probability of at least $(k+1)$ Type I errors, $k = 0, \ldots, (h_0 - 1)$,

$$gFWER(k) \equiv Pr(V_n > k) = 1 - F_{V_n}(k), \tag{4}$$

where F_{V_n} is the discrete cumulative distribution function (c.d.f.) on $\{0, \ldots, M\}$ for the number of Type I errors, V_n. When $k = 0$, the gFWER is the usual *family-wise error rate* (FWER), or probability of at least one Type I error,

$$FWER \equiv Pr(V_n > 0) = 1 - F_{V_n}(0). \tag{5}$$

The FWER is controlled, in particular, by the classical Bonferroni procedure.

Per-comparison error rate (PCER), or expected value of the proportion of Type I errors among the M tests,

$$PCER \equiv \frac{1}{M} E[V_n] = \frac{1}{M} \int v dF_{V_n}(v). \tag{6}$$

Tail probabilities for the proportion of false positives (TPPFP) among the rejected hypotheses,

$$TPPFP(q) \equiv Pr(V_n/R_n > q) = 1 - F_{V_n/R_n}(q), \qquad q \in (0,1), \tag{7}$$

where F_{V_n/R_n} is the c.d.f. for the proportion V_n/R_n of false positives among the rejected hypotheses, with the convention that $V_n/R_n \equiv 0$ if $R_n = 0$.

False discovery rate (FDR), or expected value of the proportion of false positives among the rejected hypotheses,

$$FDR \equiv E[V_n/R_n] = \int q dF_{V_n/R_n}(q), \tag{8}$$

again with the convention that $V_n/R_n \equiv 0$ if $R_n = 0$ [Benjamini and Hochberg, 1995].

Note that while the gFWER is a parameter of only the *marginal* distribution F_{V_n} of the number of Type I errors V_n (tail probability, or survivor function, for V_n), the TPPFP is a parameter of the *joint* distribution of (V_n, R_n) (tail probability, or survivor function, for V_n/R_n).

Error rates based on the *proportion* of false positives (e.g., TPPFP and FDR) are especially appealing for large-scale testing problems such as those encountered in genomics, compared to error rates based on the *number* of false positives (e.g., gFWER), as they do not increase exponentially with the number of tested hypotheses.

The aforementioned error rates are part of the broad class of Type I error rates considered in Dudoit et al. [2004] and Dudoit and van der Laan [2004], and defined as tail probabilities $Pr(g(V_n, R_n) > q)$ and expected values $E[g(V_n, R_n)]$ for an arbitrary function $g(V_n, R_n)$ of the numbers of false positives V_n and rejected hypotheses R_n. The gFWER and TPPFP correspond to the special cases $g(V_n, R_n) = V_n$ and $g(V_n, R_n) = V_n/R_n$, respectively.

Adjusted p-values. The notion of p-value extends directly to multiple testing problems, as follows. Given a MTP $\mathcal{R}_n(\alpha) = \mathcal{R}(T_n, Q_{0n}, \alpha)$, the *adjusted p-value* $\widetilde{P}_{0n}(m) = \widetilde{P}(T_n, Q_{0n})(m)$, for null hypothesis $H_0(m)$, is defined as the smallest Type I error level α at which one would reject $H_0(m)$, that is,

$$\begin{aligned} \widetilde{P}_{0n}(m) &\equiv \inf\{\alpha \in [0,1] : m \in \mathcal{R}_n(\alpha)\} \\ &= \inf\{\alpha \in [0,1] : T_n(m) \in \mathcal{C}_n(m)\}, \qquad m = 1, \ldots, M. \end{aligned} \tag{9}$$

Note that *unadjusted* or *marginal p-values*, for the test of a single hypothesis, correspond to the special case $M = 1$. For a continuous null distribution Q_{0n}, the unadjusted p-value for null hypothesis $H_0(m)$ is given by $P_{0n}(m) = P(T_n(m), Q_{0n,m}) = \bar{Q}_{0n,m}(T_n(m))$, where $Q_{0n,m}$ and $\bar{Q}_{0n,m}$ denote, respectively, the marginal c.d.f.'s and survivor functions for Q_{0n}. For example, the adjusted p-values for the classical Bonferroni procedure for FWER control are given by $\widetilde{P}_{0n}(m) = \min(MP_{0n}(m), 1)$.

As in single hypothesis tests, the smaller the adjusted p-value, the stronger the evidence against the corresponding null hypothesis. If $\mathcal{R}_n(\alpha)$ is right-continuous at α, in the sense that $\lim_{\alpha' \downarrow \alpha} \mathcal{R}_n(\alpha') = \mathcal{R}_n(\alpha)$, then one has two equivalent representations for the MTP, in terms of rejection regions for the test statistics and in terms of adjusted p-values,

$$\mathcal{R}_n(\alpha) = \{m : T_n(m) \in \mathcal{C}_n(m)\} = \{m : \widetilde{P}_{0n}(m) \leq \alpha\}. \tag{10}$$

Reporting the results of a MTP in terms of adjusted p-values, as opposed to the binary decisions to reject or not the hypotheses, offers several advantages. (i) Adjusted p-values can be defined for *any Type I error rate* (gFWER, TPPFP, FDR, etc.). (ii) They reflect the strength of the evi-

dence against each null hypothesis in terms of the Type I error rate for the *entire MTP*. (iii) They are *flexible summaries* of a MTP, in that results are supplied for *all levels* α, i.e., the level α need not be chosen ahead of time. (iv) Finally, adjusted p-values provide convenient benchmarks to *compare* different MTPs, whereby smaller adjusted p-values indicate a less conservative procedure.

Confidence regions. For the test of single-parameter null hypotheses and for any Type I error rate of the form $\theta(F_{V_n})$, Pollard and van der Laan [2004] and Dudoit and van der Laan [2004] provide results on the correspondence between single-step MTPs and θ–specific *confidence regions*.

0.2.2 Test statistics null distribution

One of the main tasks in specifying a MTP is to derive rejection regions for the test statistics such that the Type I error rate is controlled at a desired level α, i.e., such that $\theta(F_{V_n,R_n}) \leq \alpha$, for *finite sample control*, or $\limsup_n \theta(F_{V_n,R_n}) \leq \alpha$, for *asymptotic control*. It is common practice, especially for FWER control, to set $\alpha = 0.05$. However, one is immediately faced with the problem that the *true distribution* $Q_n = Q_n(P)$ of the test statistics T_n is usually *unknown*, and hence, so are the distributions of the numbers of Type I errors, $V_n = \sum_{m \in \mathcal{H}_0} \mathrm{I}(T_n(m) \in \mathcal{C}_n(m))$, and rejected hypotheses, $R_n = \sum_{m=1}^{M} \mathrm{I}(T_n(m) \in \mathcal{C}_n(m))$. In practice, the test statistics *true distribution* $Q_n(P)$ is replaced by a *null distribution* Q_0 (or estimate thereof, Q_{0n}), in order to derive rejection regions and resulting adjusted p-values.

The choice of null distribution Q_0 is crucial, in order to ensure that (finite sample or asymptotic) control of the Type I error rate under the *assumed* null distribution Q_0 does indeed provide the required control under the *true* distribution $Q_n(P)$. For proper control, the null distribution Q_0 must be such that the Type I error rate under this assumed null distribution *dominates* the Type I error rate under the true distribution $Q_n(P)$. That is, one must have $\theta(F_{V_n,R_n}) \leq \theta(F_{V_0,R_0})$, for finite sample control, and $\limsup_n \theta(F_{V_n,R_n}) \leq \theta(F_{V_0,R_0})$, for asymptotic control, where V_0 and R_0 denote, respectively, the numbers of Type I errors and rejected hypotheses under the assumed null distribution Q_0.

For error rates $\theta(F_{V_n})$ (e.g., gFWER), defined as arbitrary parameters of the distribution of the number of Type I errors V_n, we propose as null distribution the asymptotic distribution $Q_0 = Q_0(P)$ of the M–vector Z_n of null value shifted and scaled test statistics [Pollard and van der Laan, 2004, Dudoit et al., 2004, van der Laan et al., 2004, Dudoit and van der

Laan, 2004],

$$Z_n(m) \equiv \sqrt{\min\left(1, \frac{\tau_0(m)}{Var[T_n(m)]}\right)}\Big(T_n(m) + \lambda_0(m) - E[T_n(m)]\Big). \quad (11)$$

For the test of single-parameter null hypotheses using t-statistics, the null values are $\lambda_0(m) = 0$ and $\tau_0(m) = 1$. For testing the equality of K population means using F-statistics, the null values are $\lambda_0(m) = 1$ and $\tau_0(m) = 2/(K-1)$, under the assumption of equal variances in the different populations. By shifting the test statistics $T_n(m)$ as in Equation (11), one obtains a sequence of random variables $Z_n(m)$ that are asymptotically stochastically greater than the test statistics $T_n(m)$ for the true null hypotheses. Thus, the number of Type I errors V_0 under the null distribution Q_0, is asymptotically stochastically greater than the number of Type I errors V_n under the true distribution $Q_n = Q_n(P)$. Dudoit et al. [2004] and van der Laan et al. [2004] prove that the null distribution Q_0 does indeed provide the desired asymptotic control of the Type I error rate $\theta(F_{V_n})$, for general data generating distributions (with arbitrary dependence structures among variables), null hypotheses (defined in terms of submodels for the data generating distribution), and test statistics (e.g., t-statistics, F-statistics).

For a broad class of testing problems, such as the test of single-parameter null hypotheses using t-statistics as in Equation (1), the null distribution Q_0 is an M–variate Gaussian distribution with mean vector zero and covariance matrix $\Sigma^*(P)$: $Q_0 = Q_0(P) \equiv N(0, \Sigma^*(P))$. For tests of means, where the parameter of interest is the M–dimensional mean vector $\Psi(P) = \psi = E[X]$, the estimator ψ_n is simply the M–vector of sample averages and $\Sigma^*(P)$ is the correlation matrix of $X \sim P$, $Cor[X]$. More generally, for an asymptotically linear estimator ψ_n, $\Sigma^*(P)$ is the correlation matrix of the vector influence curve (IC).

Note that the following important points distinguish our approach from existing approaches to Type I error rate control. Firstly, we are only concerned with Type I error control under the *true data generating distribution* P. The notions of weak and strong control (and associated subset pivotality, Westfall & Young [Westfall and Young, 1993], p. 42–43) are therefore irrelevant to our approach. Secondly, we propose a *null distribution for the test statistics*, $T_n \sim Q_0$, and not a data generating null distribution, $X \sim P_0 \in \cap_{m=1}^{M} \mathcal{M}(m)$. The latter practice does not necessarily provide proper Type I error control, as the test statistics' *assumed* null distribution $Q_n(P_0)$ and their *true* distribution $Q_n(P)$ may have different dependence structures, in the limit, for the true null hypotheses $\mathcal{H}_0$.

Procedure 1 Bootstrap estimation of the null distribution Q_0

1. *Let $P_n^\star$ denote an estimator of the data generating distribution P. For the* non-parametric bootstrap*, $P_n^\star$ is simply the empirical distribution*

P_n, that is, samples of size n are drawn at random, with replacement from the observed data, $X_1, \ldots, X_n$. For the model-based bootstrap*, $P_n^\star$ is based on a model $\mathcal{M}$ for the data generating distribution P, such as the family of M–variate Gaussian distributions.*

2. *Generate B bootstrap samples, each consisting of n i.i.d. realizations of a random variable $X^\# \sim P_n^\star$.*

3. *For the bth bootstrap sample, $b = 1, \ldots, B$, compute an M–vector of test statistics, $T_n^\#(\cdot, b) = (T_n^\#(m, b) : m = 1, \ldots, M)$. Arrange these bootstrap statistics in an $M \times B$ matrix, $\mathbf{T}_n^\# = \left(T_n^\#(m, b)\right)$, with rows corresponding to the M null hypotheses and columns to the B bootstrap samples.*

4. *Compute row means, $E[{T_n}^\#(m, \cdot)]$, and row variances, $Var[{T_n}^\#(m, \cdot)]$, of the matrix $\mathbf{T}_n^\#$, to yield estimates of the true means $E[T_n(m)]$ and variances $Var[T_n(m)]$ of the test statistics, respectively.*

5. *Obtain an $M \times B$ matrix, $\mathbf{Z}_n^\# = \left(Z_n^\#(m, b)\right)$, of null value shifted and scaled bootstrap statistics $Z_n^\#(m, b)$, by row-shifting and scaling the matrix $\mathbf{T}_n^\#$ as in Equation 11 using the bootstrap estimates of $E[T_n(m)]$ and $Var[T_n(m)]$ and the user-supplied null values $\lambda_0(m)$ and $\tau_0(m)$. That is, compute*

$$Z_n^\#(m, b) \equiv \sqrt{\min\left(1, \frac{\tau_0(m)}{Var[{T_n}^\#(m, \cdot)]}\right)} \times \left(T_n^\#(m, b) + \lambda_0(m) - E[{T_n}^\#(m, \cdot)]\right). \tag{12}$$

6. *The bootstrap estimate Q_{0n} of the null distribution Q_0 is the empirical distribution of the B columns $Z_n^\#(\cdot, b)$ of matrix $\mathbf{Z}_n^\#$.*

In practice, since the data generating distribution P is unknown, then so is the proposed null distribution $Q_0 = Q_0(P)$. Resampling procedures, such as the bootstrap procedure of section 1, may be used to conveniently obtain consistent estimators Q_{0n} of the null distribution Q_0 and of the corresponding test statistic cut-offs and adjusted p-values [Pollard and van der Laan, 2004, Dudoit et al., 2004, van der Laan et al., 2004, Dudoit and van der Laan, 2004]. This bootstrap procedure is implemented in the internal function `boot.resample` and may be specified via the arguments `nulldist` and `B` of the main user-level function `MTP`. The reader is referred to our earlier articles and book in preparation for further detail on the choice of test statistics T_n, null distribution Q_0, and approaches for estimating this null distribution. Accordingly, we take the test statistics T_n and their null distribution Q_0 (or estimate thereof, Q_{0n}) as given, and denote the set and number of rejected hypotheses by $\mathcal{R}_n(\alpha) = \mathcal{R}(T_n, Q_{0n}, \alpha)$ and $R_n(\alpha)$ (or the shorter $\mathcal{R}_n$ and R_n), respectively, to emphasize only the dependence on the nominal Type I error level α.

0.2.3 Rejection regions

Having selected a suitable test statistics null distribution, there remains the main task of specifying rejection regions for each null hypothesis, i.e., cut-offs for each test statistic. Among the different approaches for defining rejection regions, we distinguish between the following.

Common-cut-off vs. common-quantile multiple testing procedures. In *common-cut-off procedures*, the same cut-off c_0 is used for each test statistic (cf. FWER-controlling maxT procedures [sections 2 and 4], based on maxima of test statistics). In contrast, in *common-quantile procedures*, the cut-offs are the δ_0–quantiles of the marginal null distributions of the test statistics (cf. FWER-controlling minP procedures [sections 3 and 5], based on minima of unadjusted p-values). The latter procedures tend to be more "balanced" than the former, as the transformation to p-values places the null hypotheses on an equal footing. However, this comes at the expense of increased computational complexity.

Single-step vs. stepwise multiple testing procedures. In *single-step procedures*, each null hypothesis is evaluated using a rejection region that is independent of the results of the tests of other hypotheses. Improvement in power, while preserving Type I error rate control, may be achieved by *stepwise procedures*, in which rejection of a particular null hypothesis depends on the outcome of the tests of other hypotheses. That is, the (single-step) test procedure is applied to a sequence of successively smaller nested random (i.e., data-dependent) subsets of null hypotheses, defined by the ordering of the test statistics (common-cut-off MTPs) or unadjusted p-values (common-quantile MTPs). In *step-down procedures*, the hypotheses corresponding to the *most significant* test statistics (i.e., largest absolute test statistics or smallest unadjusted p-values) are considered successively, with further tests depending on the outcome of earlier ones. As soon as one fails to reject a null hypothesis, no further hypotheses are rejected. In contrast, for *step-up procedures*, the hypotheses corresponding to the *least significant* test statistics are considered successively, again with further tests depending on the outcome of earlier ones. As soon as one hypothesis is rejected, all remaining more significant hypotheses are rejected.

Marginal vs. joint multiple testing procedures. *Marginal multiple testing procedures* are based solely on the marginal distributions of the test statistics, i.e., on cut-off rules for individual test statistics or their corresponding unadjusted p-values (e.g., classical Bonferroni FWER-controlling single-step procedure). In contrast, *joint multiple testing procedures* take into account the dependence structure of the test statistics (e.g., gFWER-controlling single-step common-cut-off

and common-quantile procedures [sections 2 and 3], based on maxima of test statistics and minima of unadjusted p-values, respectively).

The next three sections summarize three general approaches for deriving rejection regions and corresponding adjusted p-values. The chosen procedure is specified using the `method` argument to the function `MTP`.

Single-step common-cut-off and common-quantile procedures for controlling general Type I error rates $\theta(F_{V_n})$: Procedures 2 and 3, Section 0.2.4; details in Pollard and van der Laan [2004], Dudoit et al. [2004], Dudoit and van der Laan [2004].

Step-down common-cut-off (maxT) and common-quantile (minP) procedures for controlling the FWER: Procedures 4 and 5, Section 0.2.5; details in van der Laan et al. [2004], Dudoit and van der Laan [2004].

Augmentation procedures for controlling the gFWER and TPPFP, based on an initial FWER-controlling procedure: Procedures 6 and 7, Section 0.2.6; details and extensions in van der Laan et al. [2004], Dudoit et al. [2004], Dudoit and van der Laan [2004].

0.2.4 Single-step procedures for controlling general Type I error rates $\theta(F_{V_n})$

Pollard and van der Laan [2004] and Dudoit et al. [2004] propose single-step common-cut-off and common-quantile procedures for controlling arbitrary parameters $\theta(F_{V_n})$ of the distribution of the number of Type I errors. The main idea is to substitute control of the parameter $\theta(F_{V_n})$, for the *unknown, true distribution* F_{V_n} of the number of Type I errors, by control of the corresponding parameter $\theta(F_{R_0})$, for the *known, null distribution* F_{R_0} of the number of rejected hypotheses. That is, one considers single-step procedures of the form $\mathcal{R}_n(\alpha) \equiv \{m : T_n(m) > c_n(m)\}$, where the cut-offs $c_n(m)$ are chosen so that $\theta(F_{R_0}) \leq \alpha$, for $R_0 \equiv \sum_{m=1}^{M} \mathrm{I}(Z(m) > c_n(m))$ and $Z \sim Q_0$. Among the class of MTPs that satisfy $\theta(F_{R_0}) \leq \alpha$, Pollard and van der Laan [2004] and Dudoit et al. [2004] propose two procedures, based on common cut-offs and common-quantile cut-offs, respectively (Procedures 2 and 1, in Dudoit et al. [2004]). The procedures are summarized below and the reader is referred to the articles for proofs and details on the derivation of cut-offs and adjusted p-values.

Procedure 2 General θ–controlling single-step common-cut-off procedure

The set of rejected hypotheses for the general θ–controlling single-step common-cut-off procedure *is of the form* $\mathcal{R}_n(\alpha) \equiv \{m : T_n(m) > c_0\}$, *where the common cut-off* c_0 *is the* smallest *(i.e., least conservative) value*

for which $\theta(F_{R_0}) \leq \alpha$. *For* $gFWER(k)$ *control (i.e.,* $\theta(F_{V_n}) = 1 - F_{V_n}(k)$*), the procedure is based on the* $(k+1)$st ordered test statistic. *The adjusted p-values for the* single-step $T(k+1)$ procedure *are given by*

$$\widetilde{p}_{0n}(m) = Pr_{Q_0}\left(Z^\circ(k+1) \geq t_n(m)\right), \qquad m = 1, \ldots, M, \tag{13}$$

where $Z^\circ(m)$ *denotes the* m*th ordered component of* $Z = (Z(m) : m = 1, \ldots, M) \sim Q_0$*, so that* $Z^\circ(1) \geq \ldots \geq Z^\circ(M)$. *For FWER control,* $k = 0$*, one recovers the* single-step maxT procedure*, based on the* maximum test statistic*,* $Z^\circ(1) = \max_m Z(m)$*, with adjusted p-values given by*

$$\widetilde{p}_{0n}(m) = Pr_{Q_0}\left(\max_{m \in \{1,\ldots,M\}} Z(m) \geq t_n(m)\right), \qquad m = 1, \ldots, M. \tag{14}$$

Procedure 3 General θ–controlling single-step common-quantile procedure

The set of rejected hypotheses for the general θ–controlling single-step common-quantile procedure *is of the form* $\mathcal{R}_n(\alpha) \equiv \{m : T_n(m) > c_0(m)\}$*, where* $c_0(m) = Q_{0,m}^{-1}(\delta_0)$ *is the* δ_0*–quantile of the marginal null distribution* $Q_{0,m}$ *of the test statistic for the* m*th null hypothesis, i.e., the smallest value* c *such that* $Q_{0,m}(c) = Pr_{Q_0}(Z(m) \leq c) \geq \delta_0$ *for* $Z \sim Q_0$*. Here,* δ_0 *is chosen as the* smallest *(i.e., least conservative) value for which* $\theta(F_{R_0}) \leq \alpha$.

For $gFWER(k)$ *control (i.e.,* $\theta(F_{V_n}) = 1 - F_{V_n}(k)$*), the procedure is based on the* $(k+1)$st ordered unadjusted p-value. *Specifically, let* $\bar{Q}_{0,m} \equiv 1 - Q_{0,m}$ *denote the survivor functions for the marginal null distributions* $Q_{0,m}$ *and define unadjusted p-values* $P_0(m) \equiv \bar{Q}_{0,m}(Z(m))$ *and* $P_{0n}(m) \equiv \bar{Q}_{0,m}(T_n(m))$*, for* $Z \sim Q_0$ *and* $T_n \sim Q_n$*, respectively. The adjusted p-values for the* single-step $P(k+1)$ procedure *are given by*

$$\widetilde{p}_{0n}(m) = Pr_{Q_0}\left(P_0^\circ(k+1) \leq p_{0n}(m)\right), \qquad m = 1, \ldots, M, \tag{15}$$

where $P_0^\circ(m)$ *denotes the* m*th ordered component of the* M*–vector of unadjusted p-values* $P_0 = (P_0(m) : m = 1, \ldots, M)$*, so that* $P_0^\circ(1) \leq \ldots \leq P_0^\circ(M)$. *For FWER control (*$k = 0$*), one recovers the* single-step minP procedure*, based on the* minimum unadjusted p-value*,* $P_0^\circ(1) = \min_m P_0(m)$*, with adjusted p-values given by*

$$\widetilde{p}_{0n}(m) = Pr_{Q_0}\left(\min_{m \in \{1,\ldots,M\}} P_0(m) \leq p_{0n}(m)\right), \qquad m = 1, \ldots, M. \tag{16}$$

0.2.5 Step-down procedures for controlling the family-wise error rate

van der Laan et al. [2004] propose step-down common-cut-off (maxT) and common-quantile (minP) procedures for controlling the family-wise error rate, FWER. These procedures are similar in spirit to their single-step

counterparts in Section 0.2.4, for the special case $\theta(F_{V_n}) = 1 - F_{V_n}(0)$, with the important step-down distinction that hypotheses are considered successively, from most significant to least significant, with further tests depending on the outcome of earlier ones. That is, the test procedure is applied to a sequence of successively smaller nested random (i.e., data-dependent) subsets of null hypotheses, defined by the ordering of the test statistics (common-cut-off MTPs) or unadjusted p-values (common-quantile MTPs).

Procedure 4 FWER-controlling step-down common-cut-off (maxT) procedure
Let $O_n(m)$ denote the indices for the ordered test statistics $T_n(m)$, so that $T_n(O_n(1)) \geq \ldots \geq T_n(O_n(M))$. The step-down common-cut-off (maxT) procedure *is based on the distributions of maxima of test statistics over the nested subsets of ordered null hypotheses $\overline{\mathcal{O}}_n(h) \equiv \{O_n(h), \ldots, O_n(M)\}$. The adjusted p-values are given by*

$$\widetilde{p}_{0n}(o_n(m)) = \max_{h=1,\ldots,m} \left\{ Pr_{Q_0} \left(\max_{l \in \overline{\iota}_n(h)} Z(l) \geq t_n(o_n(h)) \right) \right\}, \qquad (17)$$

where $Z = (Z(m) : m = 1, \ldots, M) \sim Q_0$.

Thus, unlike single-step maxT procedure, based solely on the distribution of the maximum test statistic over all M hypotheses, the step-down common cut-offs and corresponding adjusted p-values are based on the distributions of maxima of test statistics over successively smaller nested random subsets of null hypotheses. Taking maxima of the probabilities over $h \in \{1, \ldots, m\}$ enforces monotonicity of the adjusted p-values and ensures that the procedure is indeed step-down, that is, one can only reject a particular hypothesis provided all hypotheses with more significant (i.e., larger) test statistics were rejected beforehand.

Likewise, the step-down common-quantile cut-offs and corresponding adjusted p-values are based on the distributions of minima of unadjusted p-values over successively smaller nested random subsets of null hypotheses.

Procedure 5 FWER-controlling step-down common-quantile (minP) procedure
Let $O_n(m)$ denote the indices for the ordered unadjusted p-values $P_{0n}(m)$, so that $P_{0n}(O_n(1)) \leq \ldots \leq P_{0n}(O_n(M))$. The step-down common-quantile (minP) procedure *is based on the distributions of minima of unadjusted p-values over the nested subsets of ordered null hypotheses $\overline{\mathcal{O}}_n(h) \equiv \{O_n(h), \ldots, O_n(M)\}$. The adjusted p-values are given by*

$$\widetilde{p}_{0n}(o_n(m)) = \max_{h=1,\ldots,m} \left\{ Pr_{Q_0} \left(\min_{l \in \overline{\iota}_n(h)} P_0(l) \leq p_{0n}(o_n(h)) \right) \right\}, \qquad (18)$$

where $P_0(m) \equiv \bar{Q}_{0,m}(Z(m))$ *and* $P_{0n}(m) \equiv \bar{Q}_{0,m}(T_n(m))$*, for* $Z \sim Q_0$ *and* $T_n \sim Q_n$*, respectively.*

0.2.6 Augmentation multiple testing procedures for controlling tail probability error rates

van der Laan et al. [2004], and subsequently Dudoit et al. [2004] and Dudoit and van der Laan [2004], propose *augmentation multiple testing procedures* (AMTP), obtained by adding suitably chosen null hypotheses to the set of null hypotheses already rejected by an initial gFWER-controlling MTP. Specifically, given *any* initial procedure controlling the generalized family-wise error rate, augmentation procedures are derived for controlling Type I error rates defined as tail probabilities and expected values for arbitrary functions $g(V_n, R_n)$ of the numbers of Type I errors and rejected hypotheses (e.g., proportion $g(V_n, R_n) = V_n/R_n$ of false positives among the rejected hypotheses). Adjusted p-values for the AMTP are shown to be simply shifted versions of the adjusted p-values of the original MTP. The important practical implication of these results is that *any* FWER-controlling MTP and its corresponding adjusted p-values immediately provide multiple testing procedures controlling a broad class of Type I error rates and their adjusted p-values. One can therefore build on the large pool of available FWER-controlling procedures, such as the single-step and step-down maxT and minP procedures discussed in Sections 0.2.4 and 0.2.5, above.

Augmentation procedures for controlling tail probabilities of the number (gFWER) and proportion (TPPFP) of false positives, based on an initial FWER-controlling procedure, are treated in detail in van der Laan et al. [2004] and Dudoit et al. [2004], and are summarized below. The gFWER and TPPFP correspond to the special cases $g(V_n, R_n) = V_n$ and $g(V_n, R_n) = V_n/R_n$, respectively.
Denote the adjusted p-values for the initial FWER-controlling procedure $\mathcal{R}_n(\alpha)$ by $\widetilde{P}_{0n}(m)$. Order the M null hypotheses according to these p-values, from smallest to largest, that is, define indices $O_n(m)$, so that $\widetilde{P}_{0n}(O_n(1)) \leq \ldots \leq \widetilde{P}_{0n}(O_n(M))$. Then, for a nominal level α test, the initial FWER-controlling procedure rejects the following null hypotheses

$$\mathcal{R}_n(\alpha) \equiv \{m : \widetilde{P}_{0n}(m) \leq \alpha\}. \tag{19}$$

Procedure 6 gFWER-controlling augmentation multiple testing procedure

For control of $gFWER(k)$ *at level* α*, given an initial FWER-controlling procedure* $\mathcal{R}_n(\alpha)$*, reject the* $R_n(\alpha) = |\mathcal{R}_n(\alpha)|$ *null hypotheses specified by this MTP, as well as the next* $A_n(\alpha)$ *most significant hypotheses,*

$$A_n(\alpha) = \min\{k, M - R_n(\alpha)\}. \tag{20}$$

The adjusted p-values $\widetilde{P}_{0n}^{+}(O_n(m))$ for the new gFWER-controlling AMTP are simply k–shifted versions of the adjusted p-values of the initial FWER-controlling MTP, with the first k adjusted p-values set to zero. That is,

$$\widetilde{P}_{0n}^{+}(O_n(m)) = \begin{cases} 0, & \text{if } m \leq k \\ \widetilde{P}_{0n}(O_n(m-k)), & \text{if } m > k \end{cases}. \tag{21}$$

The AMTP thus guarantees at least k rejected hypotheses.

Procedure 7 TPPFP-controlling augmentation multiple testing procedure

For control of $TPPFP(q)$ at level α, given an initial FWER-controlling procedure $\mathcal{R}_n(\alpha)$, reject the $R_n(\alpha) = |\mathcal{R}_n(\alpha)|$ null hypotheses specified by this MTP, as well as the next $A_n(\alpha)$ most significant hypotheses,

$$\begin{aligned} A_n(\alpha) &= \max\left\{ m \in \{0, \ldots, M - R_n(\alpha)\} : \frac{m}{m + R_n(\alpha)} \leq q \right\} \\ &= \min\left\{ \left\lfloor \frac{qR_n(\alpha)}{1-q} \right\rfloor, M - R_n(\alpha) \right\}, \end{aligned} \tag{22}$$

where the floor $\lfloor x \rfloor$ *denotes the greatest integer less than or equal to x, i.e., $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$. That is, keep rejecting null hypotheses until the ratio of additional rejections to the total number of rejections reaches the allowed proportion q of false positives. The adjusted p-values $\widetilde{P}_{0n}^{+}(O_n(m))$ for the new TPPFP-controlling AMTP are simply mq–shifted versions of the adjusted p-values of the initial FWER-controlling MTP. That is,*

$$\widetilde{P}_{0n}^{+}(O_n(m)) = \widetilde{P}_{0n}(O_n(\lceil (1-q)m \rceil)), \qquad m = 1, \ldots, M, \tag{23}$$

where the ceiling $\lceil x \rceil$ *denotes the least integer greater than or equal to x.*

FDR-controlling procedures

Given any TPPFP-controlling procedure, van der Laan et al. [2004] derive two simple (conservative) FDR-controlling procedures. The more general and conservative procedure controls the FDR at nominal level α, by controlling $TPPFP(\alpha/2)$ at level $\alpha/2$. The less conservative procedure controls the FDR at nominal level α, by controlling $TPPFP(1 - \sqrt{1-\alpha})$ at level $1 - \sqrt{1-\alpha}$. The reader is referred to the original article for details and proofs of FDR control (Section 2.4, Theorem 3). In what follows, we refer to these two MTPs as *conservative* and *restricted*, respectively.

0.3 Software implementation: R *multtest* package

0.3.1 Overview

The MTPs proposed in Sections 0.2.4 – 0.2.6 are implemented in the latest version of the Bioconductor R package *multtest* (Version 1.6.0). New features include: an expanded class of tests, such as tests for regression parameters in linear models and in Cox proportional hazards models; control of a wider selection of Type I error rates (e.g., gFWER, TPPFP, FDR); bootstrap estimation of the test statistics null distribution; augmentation multiple testing procedures; confidence regions for the parameter vector of interest. Because of their general applicability and novelty, we focus in this section on MTPs that utilize a bootstrap estimated test statistics null distribution and that are available through the package's main user-level function, `MTP`. Note that for many testing problems, MTPs based on a permutation (rather than bootstrap) estimated null distribution are also applicable. In particular, FWER-controlling permutation-based step-down maxT and minP MTPs are implemented in the functions `mt.maxT` and `mt.minP`, respectively, and can also be applied directly through a call to the `MTP` function.

We stress that *all* the bootstrap-based MTPs implemented in *multtest* can be performed using the main user-level function `MTP`. Note that the *multtest* package also provides several simple, marginal FWER-controlling MTPs, such as the Bonferroni, Holm [1979], Hochberg [1988], and Šidák Šidák [1967] procedures, and FDR-controlling MTPs, such as the Benjamini & Hochberg [Benjamini and Hochberg, 1995] and Benjamini & Yekutieli [Benjamini and Yekutieli, 2001] step-up procedures. These procedures are available through the `mt.rawp2adjp` function, which takes a vector of unadjusted p-values as input and returns the corresponding adjusted p-values. For greater detail on *multtest* functions, the reader is referred to the package documentation, in the form of help files, e.g., `?MTP`, and vignettes, e.g., `openVignette("multtest")`.

As detailed in Section 0.2.1, above, one needs to specify the following main ingredients when applying a MTP: the *data*, $X_1, \ldots, X_n$; suitably defined *test statistics*, T_n, for each of the null hypotheses under consideration (e.g., one-sample t-statistics, robust rank-based F-statistics, t-statistics for regression coefficients in Cox proportional hazards model); a choice of *Type I error rate*, $\theta(F_{V_n,R_n})$, providing an appropriate measure of false positives for the particular testing problem (e.g., $TPPFP(0.10)$); a proper *joint null distribution*, Q_0 (or estimate thereof, Q_{0n}), for the test statistics (e.g., bootstrap null distribution as in the bootstrap procedure of section 1); given the previously defined components, a *multiple testing procedure*, $\mathcal{R}_n = \mathcal{R}(T_n, Q_{0n}, \alpha)$, for controlling the error rate $\theta(F_{V_n,R_n})$ at a target level α.

Accordingly, the *multtest* package has adopted a modular and extensible approach to the implementation of MTPs, with the following four main types of functions.

Functions for computing the test statistics, T_n. These are internal functions (e.g., `meanX`, `coxY`), i.e., functions that are generally not called directly by the user. As shown in Section 0.3.2, below, the type of test statistic is specified by the `test` argument of the main user-level function `MTP`. Advanced users, interested in extending the class of tests available in *multtest*, can simply add their own test statistic functions to the existing library of such internal functions (see Section 0.3.4, below, for a brief discussion of the function closure approach for specifying test statistics).

Functions for obtaining the test statistics null distribution, Q_0, or an estimate thereof, Q_{0n}. The main function currently available is the internal function `boot.resample`, implementing the non-parametric version of the bootstrap procedure of section 1.

Functions for implementing the multiple testing procedure, $\mathcal{R}_n = \mathcal{R}(T_n, Q_{0n}, \alpha)$. The main user-level function is the wrapper function `MTP`, which returns rejection regions, confidence regions, and adjusted p-values, for MTPs controlling a variety of Type I error rates. In particular, it implements the single-step and step-down maxT and minP procedures for FWER control (Sections 0.2.4 and 0.2.5). The functions `fwer2gfwer`, `fwer2tppfp`, and `fwer2fdr` implement, respectively, gFWER-, TPPFP-, and FDR-controlling augmentation multiple testing procedures, based on adjusted p-values from *any* FWER-controlling procedure, and can be called via the `typeone` argument to `MTP` (Section 0.2.6).

Functions for numerical and graphical summaries of a MTP. As described in Section 0.3.3, below, a number of summary methods are available to operate on objects of class *MTP*, output from the main `MTP` function.

0.3.2 Resampling-based multiple testing procedures: `MTP` function

The main user-level function for resampling-based multiple testing is `MTP`.

```
> args(MTP)
function (X, W = NULL, Y = NULL, Z = NULL, Z.incl = NULL, Z.test = NULL,
    na.rm = TRUE, test = "t.twosamp.unequalvar", robust = FALSE,
    standardize = TRUE, alternative = "two.sided", psi0 = 0,
    typeone = "fwer", k = 0, q = 0.1, fdr.method = "conservative",
    alpha = 0.05, nulldist = "boot", B = 1000, method = "ss.maxT",
    get.cr = FALSE, get.cutoff = FALSE, get.adjp = TRUE, keep.nulldist = FALSE,
```

```
    seed = NULL)
NULL
```

INPUT.

Data. The data, `X`, consist of a J–dimensional random vector, observed on each of n sampling units (patients, cell lines, mice, etc.). These data can be stored in a $J \times n$ *matrix*, *data.frame*, or *exprs* slot of an object of class *exprSet*. In some settings, a J–vector of weights may be associated with each observation, and stored in a $J \times n$ weight *matrix*, `W` (or an n–vector `W`, if the weights are the same for each of the J variables). One may also observe a possibly censored continuous or polychotomous outcome, `Y`, for each sampling unit, as obtained, for example, from the *phenoData* slot of an object of class *exprSet*. In some studies, L additional covariates may be measured on each sampling unit and stored in `Z`, an $n \times L$ *matrix* or *data.frame*. When the tests concern parameters in regression models with covariates from `Z` (e.g., values `lm.XvsZ`, `lm.YvsXZ`, and `coxph.YvsXZ`, for the argument `test`, described below), the arguments `Z.incl` and `Z.test` specify, respectively, which covariates (i.e., which columns of `Z`, including `Z.test`) should be included in the model and which regression parameter is to be tested (only when `test="lm.XvsZ"`). The covariates can be specified either by a numeric column index or character string. If `X` is an instance of the class *exprSet*, `Y` can be a column index or character string referring to the variable in the *data.frame* `pData(X)` to use as outcome. Likewise, `Z.incl` and `Z.test` can be column indices or character strings referring to the variables in `pData(X)` to use as covariates. The argument `na.rm` controls the treatment of missing values (`NA`). It is `TRUE` by default, so that an observation with a missing value in any of the data objects' jth component ($j = 1, \ldots, J$) is excluded from the computation of any test statistic based on this jth variable.

Test statistics. The test statistics should be chosen based on the parameter of interest (e.g., location, scale, or regression parameters) and the hypotheses one wishes to test. In the current implementation of *multtest*, the following test statistics are available through the argument `test`, with default value `t.twosamp.unequalvar`, for two-sample Welch t-statistics.

- `t.onesamp`: One-sample t-statistics for tests of means.
- `t.twosamp.equalvar`: Equal variance two-sample t-statistics for tests of differences in means.
- `t.twosamp.unequalvar`: Unequal variance two-sample t-statistics for tests of differences in means (also known as two-sample Welch t-statistics).

- `t.pair`: Two-sample paired t-statistics for tests of differences in means.
- `f`: Multi-sample F-statistics for tests of equality of population means.
- `f.block`: Multi-sample F-statistics for tests of equality of population means in a block design.
- `lm.XvsZ`: t-statistics for tests of regression coefficients for variable `Z.test` in linear models each with outcome `X[j,]` ($j = 1, \ldots, J$), and possibly additional covariates `Z.incl` from the *matrix* `Z` (in the case of no covariates, one recovers the one-sample t-statistic, `t.onesamp`).
- `lm.YvsXZ`: t-statistics for tests of regression coefficients in linear models with outcome `Y` and each `X[j,]` ($j = 1, \ldots, J$) as covariate of interest, with possibly other covariates `Z.incl` from the *matrix* `Z`.
- `coxph.YvsXZ`: t-statistics for tests of regression coefficients in Cox proportional hazards survival models with outcome `Y` and each `X[j,]` ($j = 1, \ldots, J$) as covariate of interest, with possibly other covariates `Z.incl` from the *matrix* `Z`.

Robust, *rank-based* versions of the above test statistics can be specified by setting the argument `robust` to `TRUE` (the default value is `FALSE`). Consideration should be given to whether *standardized* or *unstandardized* difference statistics are most appropriate (Equation (1); see Pollard and van der Laan [2004] for a comparison). Both options are available through the argument `standardize`, by default `TRUE`. The type of alternative hypotheses is specified via the `alternative` argument: default value of `two.sided`, for two-sided test, and values of `less` or `greater`, for one-sided tests. The (common) null value for the parameters of interest is specified through the `psi0` argument, by default zero.

Type I error rate. The `MTP` function controls by default the FWER (argument `typeone="fwer"`). Augmentation procedures (Section 0.2.6), controlling other Type I error rates such as the gFWER, TPPFP, and FDR, can be specified through the argument `typeone`. Related arguments include `k` and `q`, for the allowed number and proportion of false positives for control of $gFWER(k)$ and $TPPFP(q)$, respectively, and `fdr.method`, for the type of TPPFP-based FDR-controlling procedure (i.e., `"conservative"` or `"restricted"` methods). The nominal level of the test is determined by the argument `alpha`, by default 0.05. Testing can be performed for a range of nominal Type I error rates by specifying a vector of levels `alpha`.

Test statistics null distribution. The test statistics null distribution is estimated by default using the non-parametric version of the bootstrap

procedure of section 1 (argument `nulldist="boot"`). The bootstrap procedure is implemented in the internal function `boot.resample`, which calls C to compute test statistics for each bootstrap sample. The values of the shift (λ_0) and scale (τ_0) parameters are determined by the type of test statistics (e.g., $\lambda_0 = 0$ and $\tau_0 = 1$ for t-statistics). Permutation null distributions are also available via `nulldist="perm"`. The number of resampling steps is specified by the argument `B`, by default 1,000.

Multiple testing procedures. Several methods for controlling the chosen Type I error rate are available in *multtest*.

- *FWER-controlling procedures.* The `MTP` function implements the single-step and step-down (common-cut-off) maxT and (common-quantile) minP MTPs for FWER control, described in Sections 0.2.4 and 0.2.5, and specified through the argument `method` (internal functions `ss.maxT`, `ss.minP`, `sd.maxT`, and `sd.minP`). The default MTP is the single-step maxT procedure (`method="ss.maxT"`), since it requires the least computation.
- *gFWER-, TPPFP-, and FDR-controlling augmentation procedures.* As discussed in Section 0.2.6, any FWER-controlling MTP can be trivially augmented to control additional Type I error rates, such as the gFWER and TPPFP. Two FDR-controlling procedures can then be derived from the TPPFP-controlling AMTP. AMTPs are implemented in the functions `fwer2gfwer`, `fwer2tppfp`, and `fwer2fdr`, which take FWER adjusted p-values as input and return augmentation adjusted p-values for control of the gFWER, TPPFP, and FDR, respectively. Note that the aforementioned AMTPs can be applied directly via the `typeone` argument of the main function `MTP`.

Output control. Various arguments are available to specify which combination of the following quantities should be returned: confidence regions (argument `get.cr`); cut-offs for the test statistics (argument `get.cutoff`); adjusted p-values (argument `get.adjp`); test statistics null distribution (argument `keep.nulldist`). Note that parameter estimates and confidence regions only apply to the test of single-parameter null hypotheses (i.e., not the F-tests). In addition, in the current implementation of `MTP`, parameter confidence regions and test statistic cut-offs are only provided when `typeone="fwer"`, so that `get.cr` and `get.cutoff` should be set to `FALSE` when using the error rates gFWER, TPPFP, or FDR.

OUTPUT.

The S4 class/method object-oriented programming approach was adopted to summarize the results of a MTP (Section 0.3.4). The output of the `MTP` function is an instance of the *class MTP*, with the following *slots*,

```
> slotNames("MTP")
```

```
[1] "statistic" "estimate"  "sampsize"  "rawp"      "adjp"      "conf.reg"
[7] "cutoff"    "reject"    "nulldist"  "call"      "seed"
```

MTP results. An instance of the *MTP* class contains slots for the following MTP results:

- `statistic`: The numeric M–vector of test statistics, specified by the values of the `MTP` arguments `test`, `robust`, `standardize`, and `psi0`. In many testing problems, $M = J =$ `nrow(X)`.
- `estimate`: For the test of single-parameter null hypotheses using t-statistics (i.e., not the F-tests), the numeric M–vector of estimated parameters.
- `sampsize`: The sample size, i.e., $n =$ `ncol(X)`.
- `rawp`: The numeric M–vector of unadjusted p-values.
- `adjp`: The numeric M–vector of adjusted p-values (computed only if the `get.adjp` argument is `TRUE`).
- `conf.reg`: For the test of single-parameter null hypotheses using t-statistics (i.e., not the F-tests), the numeric $M \times 2 \times$ `length(alpha)` *array* of lower and upper simultaneous confidence limits for the parameter vector, for each value of the nominal Type I error rate `alpha` (computed only if the `get.cr` argument is `TRUE`).
- `cutoff`: The numeric $M \times$ `length(alpha)` *matrix* of cut-offs for the test statistics, for each value of the nominal Type I error rate `alpha` (computed only if the `get.cutoff` argument is `TRUE`).
- `reject`: The $M \times$ `length(alpha)` *matrix* of rejection indicators (`TRUE` for a rejected null hypothesis), for each value of the nominal Type I error rate `alpha`.

Null distribution. The `nulldist` slot contains the $M \times B$ *matrix* for the estimated test statistics null distribution. This slot is returned only if `keep.nulldist=TRUE`; option not currently available for permutation null distribution, i.e., `nulldist="perm"`. By default (i.e., for `nulldist="boot"`), the entries of `nulldist` are the null value shifted and scaled bootstrap test statistics, as defined in the bootstrap procedure of section 1.

Reproducibility. The last two slots of an *MTP* object provide information on the particular call to the `MTP` function and can be used for reproducibility in a repeat call to `MTP`. The slot `call` contains the call to the function `MTP`, and `seed` is an integer specifying the state of the random number generator used to create the resampled datasets. The seed ar-

gument is currently used only for the bootstrap null distribution (i.e., for `nulldist="boot"`).

0.3.3 Numerical and graphical summaries

The following *methods* were defined to operate on *MTP* instances and summarize the results of a MTP.

`print`: The `print` method returns a description of an object of class *MTP*, including the sample size n, the number M of tested hypotheses, the type of test performed (value of argument `test`), the Type I error rate (value of argument `typeone`), the nominal level of the test (value of argument `alpha`), the name of the MTP (value of argument `method`), the call to the function `MTP`. In addition, this method produces a table with the class, mode, length, and dimension of each slot of the *MTP* instance.

`summary`: The `summary` method provides numerical summaries of the results of a MTP and returns a list with the following three components:

- `rejections`: A *data.frame* with the number(s) of rejected hypotheses for the nominal Type I error rate(s) specified by the `alpha` argument of the function `MTP` (`NULL` values are returned if all three arguments `get.cr`, `get.cutoff`, and `get.adjp` are `FALSE`).
- `index`: A numeric M–vector of indices for ordering the hypotheses according to first `adjp`, then `rawp`, and finally the absolute value of `statistic` (not printed in the summary).
- `summaries`: When applicable (i.e., when the corresponding quantities are returned by `MTP`), a table with six number summaries of the distributions of the adjusted p-values, unadjusted p-values, test statistics, and parameter estimates.

`plot`: The `plot` method produces the following graphical summaries of the results of a MTP. The type of display may be specified via the `which` argument.

1. Scatterplot of number of rejected hypotheses vs. nominal Type I error rate.
2. Plot of ordered adjusted p-values; can be viewed as a plot of Type I error rate vs. number of rejected hypotheses.
3. Scatterplot of adjusted p-values vs. test statistics (also known as "volcano plot").
4. Plot of unordered adjusted p-values.
5. Plot of confidence regions for user-specified parameters, by default the 10 parameters corresponding to the smallest adjusted p-values (argument `top`).

6. Plot of test statistics and corresponding cut-offs (for each value of `alpha`) for user-specified hypotheses, by default the 10 hypotheses corresponding to the smallest adjusted p-values (argument `top`).

The argument `logscale` (by default equal to `FALSE`) allows one to use the negative decimal logarithms of the adjusted p-values in the second, third, and fourth graphical displays. Note that some of these plots are implemented in the older function `mt.plot`.

`[`: Subsetting method, which operates selectively on each slot of an *MTP* instance to retain only the data related to the specified hypotheses.

`as.list`: Converts an object of class *MTP* to an object of class *list*, with an entry for each slot.

0.3.4 Software design

The following features of the programming approach employed in *multtest* may be of interest to users, especially those interested in extending the functionality of the package.

Function closures. The use of *function closures*, as in the *genefilter* package, allows uniform data input for all MTPs and facilitates the extension of the package's functionality by adding, for example, new types of test statistics. Specifically, a function closure is defined for each value of the `MTP` argument `test`. The closure consists of a function for computing the test statistic (with only two arguments, a data vector `x` and a corresponding weight vector `w`, with default value of `NULL`) and its enclosing environment, with bindings for relevant additional arguments, such as null values `psi0`, outcomes `Y`, and covariates `Z`. Existing internal test statistic functions are located in the file `R/statistics.R`. Thus, new test statistics can be added to *multtest* by simply defining a new closure and adding a corresponding value for the `test` argument to `MTP`.

Class/method object-oriented programming. Like many other Bioconductor packages, *multtest* has adopted the *S4 class/method object-oriented programming approach* of Chambers [1998]. In particular, a new class, *MTP*, and associated methods, were defined to represent and operate on the results of multiple testing procedures.

Calls to C. Because resampling procedures, such as the non-parametric bootstrap implemented in *multtest*, are computationally intensive, care must be taken to ensure that the resampling steps are not prohibitively slow. The use of function closures for the test statistics, however, prevents writing the entire program in C. In the current implementation, we have

chosen to define the closure and compute the observed test statistics in R, and then call C to apply the closure to each bootstrap resampled dataset (using the R random number generator). This approach puts the for loops over bootstrap samples (B) and hypotheses (M) in the compiled code, thus speeding up this computationally expensive part of the program.

0.4 Applications: ALL microarray dataset

0.4.1 ALL data package and initial gene filtering

We illustrate some of the functionality of the *multtest* package using the Acute Lymphoblastic Leukemia (ALL) microarray dataset of Chiaretti et al. [2004], available in the data package *ALL*. The main object in this package is `ALL`, an instance of the class *exprSet*. The genes-by-subjects matrix of 12,625 Affymetrix *expression measures* (chip series HG-U95Av2) for each of 128 ALL patients is provided in the `exprs` slot of `ALL`. The `phenoData` slot contains 21 *phenotypes* (i.e., patient level responses and covariates) for each patient. Note that the expression measures have been obtained using the three-step robust multichip average (RMA) preprocessing method, implemented in the package *affy*. In particular, the expression measures have been subject to a base 2 logarithmic transformation. For greater detail, please consult the *ALL* package documentation.

```
> library("ALL")
> library("hgu95av2")
> data(ALL)
```

Our goal is to identify genes whose expression measures are associated with (possibly censored) biological and clinical outcomes such as: tumor cellular subtype (B-cell vs. T-cell), tumor molecular subtype (BCR/ABL, NEG, ALL1/AF4), and time to relapse. Alternative analyses of this dataset are discussed in Chapters **??**, **??**, **??**, **??**, and **??**. Before applying the MTPs, we perform initial gene filtering as in Chiaretti et al. [2004] and retain only those genes for which: (i) at least 20% of the subjects have a measured intensity of at least 100 and (ii) the coefficient of variation (i.e., the ratio of the standard deviation to the mean) of the intensities across samples is between 0.7 and 10. These two filtering criteria can be readily applied using functions from the *genefilter* package.

```
> ffun <- filterfun(pOverA(p = 0.2, A = 100), cv(a = 0.7, b = 10))
> filt <- genefilter(2^exprs(ALL), ffun)
> filtALL <- ALL[filt, ]

> filtX <- exprs(filtALL)
> pheno <- pData(filtALL)
```

The new filtered dataset, `filtALL`, contains expression measures on 431 genes, for 128 patients.

0.4.2 Association of expression measures and tumor cellular subtype: two-sample t-statistics

FWER-controlling step-down minP MTP with two-sample Welch t-statistics and bootstrap null distribution

Different tissues are involved in ALL tumors of the B-cell and T-cell subtypes. The phenotypic data include a variable, `BT`, which encodes the tissue type and stage of differentiation. In order to identify genes with higher mean expression measures in B-cell ALL patients compared to T-cell ALL patients, we create an indicator variable, `Bcell` (1 for B-cell, 0 for T-cell), and compute, for each gene, a two-sample Welch (unequal variance) t-statistic. We choose to control the FWER using the bootstrap-based step-down minP procedure with $B = 100$ bootstrap iterations, although more bootstrap iterations are recommended in practice.

```
> table(pData(ALL)$BT)

 B B1 B2 B3 B4  T T1 T2 T3 T4
 5 19 36 23 12  5  1 15 10  2

> Bcell <- rep(0, length(pData(ALL)$BT))
> Bcell[grep("B", as.character(pData(ALL)$BT))] <- 1

> seed <- 99
> BT.boot <- MTP(X = filtX, Y = Bcell, alternative = "greater",
+     B = 100, method = "sd.minP", seed = seed)

running bootstrap...
iteration = 100
```

Let us examine the results of the MTP stored in the object `BT.boot`.

```
> summary(BT.boot)

MTP:  sd.minP
Type I error rate:  fwer

  Level Rejections
1  0.05        194

             Min. 1st Qu. Median   Mean 3rd Qu.   Max.
adjp        0.000  0.0000 0.8700 0.5314  1.0000  1.000
rawp        0.000  0.0000 0.0300 0.3559  0.9450  1.000
statistic -34.420 -1.5690 2.0120 2.0590  5.3830 22.330
estimate   -4.655 -0.3168 0.3814 0.3258  0.9949  4.249
```

The `summary` method prints the name of the MTP (here, `sd.minP`, for step-down minP), the Type I error rate (here, `fwer`), the number of rejections at each Type I error rate level specified in `alpha` (here, 194 at level $\alpha = 0.05$), and six number summaries (mean and quantiles) of the adjusted p-values, unadjusted p-values, test statistics, and parameter estimates (here, difference in means).

The following commands may be used to obtain a list of genes that are differentially expressed in B-cell vs. T-cell ALL patients at nominal FWER level $\alpha = 0.05$, i.e., genes with adjusted p-values less than or equal to 0.05. Functions from the *annotate* and *annaffy* packages may then be used to obtain annotation information on these genes (e.g., gene names, PubMed abstracts, GO terms) and to generate HTML tables of the results. Here, we list the names of three genes only.

```
> BT.diff <- BT.boot@adjp <= 0.05
> BT.AffyID <- geneNames(filtALL)[BT.diff]
> mget(BT.AffyID[1:3], env = hgu95av2GENENAME)

$"1005_at"
[1] "dual specificity phosphatase 1"

$"1065_at"
[1] "fms-related tyrosine kinase 3"

$"1096_g_at"
[1] "CD19 antigen"
```

Various graphical summaries of the results may be obtained using the `plot` method, by selecting appropriate values of the argument `which`. Figure 1 displays four such plots. We see (top left) that the number of rejections increases slightly when nominal FWER is greater than 0.6, and then increases quickly as FWER approaches 1. Similarly, the adjusted p-values for many genes are close to either 0 or 1 (top right) and the test statistics for genes with small p-values do not overlap with those for genes with p-values close to 1 (bottom left). Together these results indicate that there is a clear separation between the rejected and accepted hypotheses, i.e., between genes that are declared differentially expressed and those that are not.

```
> par(mfrow = c(2, 2))
> plot(BT.boot)
```

Marginal FWER-controlling MTPs with two-sample Welch t-statistics and bootstrap null distribution

Given a vector of unadjusted p-values, the `mt.rawp2adjp` function computes adjusted p-values for the marginal FWER-controlling MTPs of

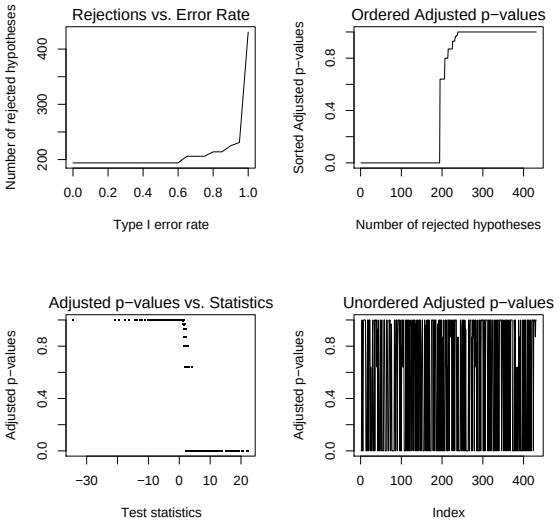

Figure 1. *B-cell vs. T-cell ALL — FWER-controlling step-down minP MTP.* By default, four graphical summaries are produced by the `plot` method for instances of the class *MTP*.

Bonferroni, Holm [1979], Hochberg [1988], and Šidák [Šidák, 1967], discussed in detail in Dudoit et al. [2003]. The `mt.plot` function may then be used to compare the different procedures in terms of their adjusted p-values.

```
> marg <- c("Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD")
> BT.marg <- mt.rawp2adjp(rawp = BT.boot@rawp, proc = marg)
> comp.marg <- cbind(BT.boot@adjp, BT.marg$adjp[order(BT.marg$index),
+     -1])

> par(mfrow = c(1, 1))
> mt.plot(adjp = comp.marg, teststat = BT.boot@statistic, proc = c("SD minP",
+     marg), leg = c(0.1, 400), col = 1:6, lty = 1:6, lwd = 3)
> tmp <- title("Comparison of FWER-controlling marginal MTPs and \n step-down minP MTP")
```

Figure 2 displays the number of rejected hypotheses vs. the nominal Type I error rate for the various FWER-controlling MTPs. For the ALL dataset, the marginal MTPs all perform similarly, making very few rejections at nominal Type I error rates near zero. The step-down minP procedure, which takes into account the joint distribution of the test statistics, leads to more rejections than the marginal methods.

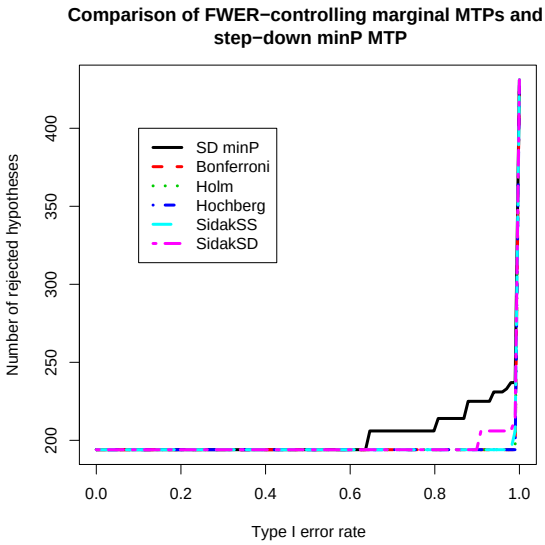

Figure 2. *B-cell vs. T-cell ALL — Marginal vs. joint FWER-controlling MTPs.* Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing bootstrap-based marginal MTPs and bootstrap-based step-down minP MTP.

FWER-controlling step-down minP MTP with two-sample Welch t-statistics and permutation null distribution

Because the sample sizes are unequal for the B-cell and T-cell patients and the expression measures may have different covariance structures in the two populations, we expect the bootstrap and permutation null distributions to yield different sets of rejected hypotheses (Pollard and van der Laan [2004]). To compare the two approaches, we apply the permutation-based step-down minP procedure, first using the `mt.minP` function and then using the new `MTP` function (which calls `mt.minP`). Please note that while the `MTP` and `mt.minP` functions produce the same results, these are presented in a different manner. In particular, for the function `MTP`, the results (e.g., test statistics, parameter estimates, unadjusted p-values, adjusted p-values, cut-offs) are given in the original order of the null hypotheses, while in the `mt.minP` function, the hypotheses are sorted first according to their adjusted p-values, next their unadjusted p-values, and finally their test statistics. In addition, the function `MTP` implements a broader range of MTPs and has adopted the S4 class/method design for representing and summarizing the results of a MTP.

```
> set.seed(99)
> BT.perm.old <- mt.minP(X = filtX, classlabel = Bcell, side = "upper",
+     B = 100)
```

```
> names(BT.perm.old)

[1] "index"    "teststat" "rawp"     "adjp"     "plower"
```

The `mt.minP` function returns a list with five components: a vector of row indices used to sort the hypotheses based on significance (`index`), the test statistics sorted according to `index` (`teststat`), unadjusted p-values (`rawp`), adjusted p-values (`adjp`), and adjusted p-values based on ignoring ties in the permutation distribution (`plower`). See `?mt.minP` for details.

```
> set.seed(99)
> BT.perm.new <- MTP(X = filtX, Y = Bcell, alternative = "greater",
+     nulldist = "perm", B = 100, method = "sd.minP")

> sum(BT.perm.old$adjp <= 0.05)

[1] 0

> sum(BT.perm.new@adjp <= 0.05)

[1] 0

> sum(BT.perm.new@adjp <= 0.05 & BT.boot@adjp <= 0.05)

[1] 0
```

At nominal FWER level $\alpha = 0.05$, the permutation step-down minP procedure identifies 0 genes as differentially expressed between patients with B-cell and T-cell ALL. In contrast, the bootstrap step-down minP procedure identifies 194 differentially expressed genes.

FWER-controlling step-down minP MTP with robust two-sample t-statistics and bootstrap null distribution

The Wilcoxon rank sum statistic (also known as the Mann-Whitney statistic) is a robust alternative to the usual two-sample t-statistic.

```
> BT.wilcox <- MTP(X = filtX, Y = Bcell, robust = TRUE, alternative = "greater",
+     B = 100, method = "sd.minP", seed = seed)

> sum(BT.wilcox@adjp <= 0.05)

[1] 193

> sum(BT.wilcox@adjp <= 0.05 & BT.boot@adjp <= 0.05)

[1] 186
```

At nominal FWER level $\alpha = 0.05$, the bootstrap step-down minP MTP, based on the robust Wilcoxon test statistic, identifies 193 genes as differentially expressed, compared to 194 genes for the same MTP based on the Welch t-statistic. 186 genes are identified by both procedures.

0.4.3 Augmentation procedures

In the context of microarray gene expression data analysis or other high-dimensional inference problems, one is often willing to tolerate some false positives, provided their number is small in comparison to the number of rejected hypotheses. In this case, the FWER is not a suitable choice of Type I error rate and one should consider other rates that lead to larger sets of rejected hypotheses. The augmentation procedures of Section 0.2.6, implemented in the function `MTP`, allow one to reject additional hypotheses, while controlling an error rate such as the generalized family-wise error rate (gFWER), the tail probability for the proportion of false positives (TPPFP), or the false discovery rate (FDR). We illustrate the use of the `fwer2gfwer`, `fwer2tppfp`, and `fwer2fdr` functions, but note that the gFWER, TPPFP, and FDR can also be controlled directly using the main `MTP` function, with appropriate choices of arguments `typeone`, `k`, `q`, and `fdr.method`.

gFWER control

```
> k <- c(5, 10, 50, 100)
> BT.gfwer <- fwer2gfwer(adjp = BT.boot@adjp, k = k)
> comp.gfwer <- cbind(BT.boot@adjp, BT.gfwer)
> mtps <- paste("gFWER(", c(0, k), ")", sep = "")
> mt.plot(adjp = comp.gfwer, teststat = BT.boot@statistic, proc = mtps,
+     leg = c(0.1, 430), col = 1:5, lty = 1:5, lwd = 3)
> tmp <- title("Comparison of gFWER(k)-controlling AMTPs \n based on SD minP MTP")
```

For gFWER-controlling AMTPs, Figure 3 illustrates that the number of rejected hypotheses increases linearly with the number k of allowed false positives, for nominal levels α such that the initial FWER-controlling MTP does not reject more than $M - k$ hypotheses. That is, the curve for the $gFWER(k)$–controlling AMTP is obtained from that of the initial FWER-controlling procedure by a simple vertical shift of k.

TPPFP control

```
> q <- c(0.05, 0.1, 0.25)
> BT.tppfp <- fwer2tppfp(adjp = BT.boot@adjp, q = q)
> comp.tppfp <- cbind(BT.boot@adjp, BT.tppfp)
> mtps <- c("FWER", paste("TPPFP(", q, ")", sep = ""))
> mt.plot(adjp = comp.tppfp, teststat = BT.boot@statistic, proc = mtps,
+     leg = c(0.1, 430), col = 1:4, lty = 1:4, lwd = 3)
> tmp <- title("Comparison of TPPFP(q)-controlling AMTPs \n based on SD minP MTP")
```

Figure 4 shows that, as expected, the number of rejections increases with the allowed proportion q of false positives when controlling $TPPFP(q)$ at a given level α.

FDR control

Given any TPPFP-controlling MTP, van der Laan et al. [2004] derive two simple (conservative) FDR-controlling MTPs. Here, we compare these two FDR-controlling approaches, based on a TPPFP-controlling augmentation of the step-down minP procedure, to the marginal Benjamini & Hochberg [Benjamini and Hochberg, 1995] and Benjamini & Yekutieli [Benjamini and Yekutieli, 2001] procedures, implemented in the function `mt.rawp2adjp`. The following code chunk first computes adjusted p-values for the augmentation procedures, then for the marginal procedures, and finally makes a plot of the numbers of rejections vs. the nominal FDR for the four MTPs.

```
> BT.fdr <- fwer2fdr(adjp = BT.boot@adjp, method = "both")$adjp
> BT.marg.fdr <- mt.rawp2adjp(rawp = BT.boot@rawp, proc = c("BY",
+     "BH"))
> comp.fdr <- cbind(BT.fdr, BT.marg.fdr$adjp[order(BT.marg.fdr$index),
+     -1])
> mtps <- c("AMTP Cons", "AMTP Rest", "BY", "BH")
> mt.plot(adjp = comp.fdr, teststat = BT.boot@statistic, proc = mtps,
+     leg = c(0.1, 430), col = c(2, 2, 3, 3), lty = rep(1:2, 2),
+     lwd = 3)
> tmp <- title("Comparison of FDR-controlling MTPs")
```

Figure 5 shows that the AMTPs based on conservative bounds for the FDR ("AMTP Cons" and "AMTP Rest") are more conservative than the Benjamini & Hochberg ("BH") MTP for nominal FDR less than 0.4, but less conservative than "BH" for larger FDR. The Benjamini & Yekutieli ("BY") MTP, a conservative version of the Benjamini & Hochberg MTP (with $\sim \log M$ penalty on the p-values), leads to the fewest rejections.

0.4.4 Association of expression measures and tumor molecular subtype: multi-sample F-statistics

The phenotype data include a variable, `mol.bio`, which records chromosomal abnormalities, such as the BCR/ABL gene rearrangement; these abnormalities concern primarily patients with B-cell ALL and may be related to prognosis. To identify genes with differences in mean expression measures between different tumor molecular subtypes (BCR/ABL, NEG, ALL1/AF4, E2A/PBX1, p15/p16), within B-cell ALL subjects, one can perform a family of F-tests. Tumor subtypes with fewer than 10 subjects are removed from the analysis. Adjusted p-values and test statistic cut-offs (for nominal levels α of 0.01 and 0.10) are computed as follows for the FWER-controlling bootstrap-based single-step maxT procedure.

```
> BX <- filtX[, Bcell == 1]
> Bpheno <- pheno[Bcell == 1, ]
```

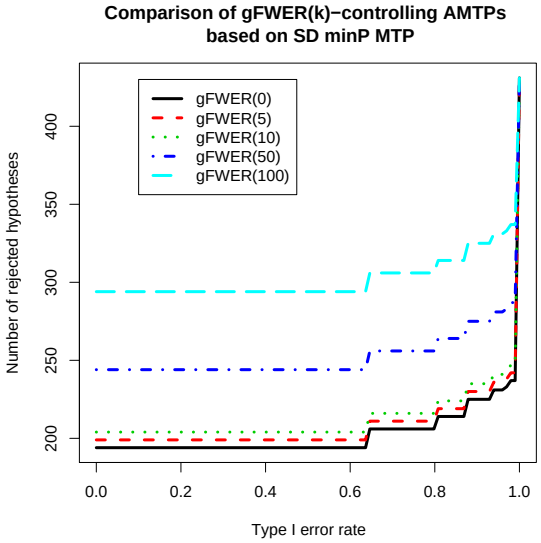

Figure 3. *B-cell vs. T-cell ALL — gFWER-controlling AMTPs.* Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing gFWER-controlling AMTPs, based on the FWER-controlling bootstrap-based step-down minP procedure, for different allowed numbers k of false positives.

```
> mb <- as.character(Bpheno$mol.biol)
> table(mb)

mb
ALL1/AF4  BCR/ABL E2A/PBX1      NEG  p15/p16
      10       37        5       42        1

> other <- c("E2A/PBX1", "p15/p16")
> mb.boot <- MTP(X = BX[, !(mb %in% other)], Y = mb[!(mb %in% other)],
+     test = "f", alpha = c(0.01, 0.1), B = 100, get.cutoff = TRUE,
+     seed = seed)

running bootstrap...
iteration = 100
```

Let us examine the results of the MTP.

```
> summary(mb.boot)

MTP:  ss.maxT
Type I error rate:  fwer

  Level Rejections
1  0.01        118
```

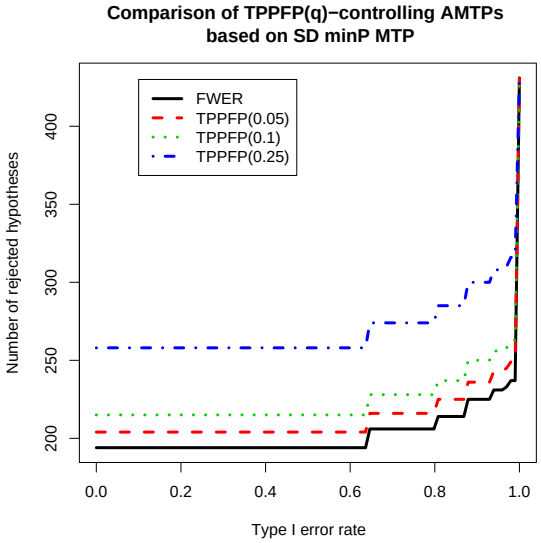

Figure 4. *B-cell vs. T-cell ALL — TPPFP-controlling AMTPs.* Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing TPPFP-controlling AMTPs, based on the FWER-controlling bootstrap-based step-down minP procedure, for different allowed proportions q of false positives.

```
2  0.10        135

               Min. 1st Qu. Median   Mean 3rd Qu.  Max.
adjp       0.0000000    0.00   0.91 0.5890   1.000  1.00
rawp       0.0000000    0.00   0.01 0.1457   0.170  1.00
statistic  0.0003624    1.29   3.30 5.8360   7.979 67.84
estimate          NA      NA     NA    NaN      NA    NA

> mb.diff <- mb.boot@adjp <= 0.01
> sum(mb.diff)

[1] 118

> sum(mb.boot@statistic >= mb.boot@cutoff[, "alpha=0.01"] & mb.diff)

[1] 118
```

For control of the FWER at nominal level $\alpha = 0.01$, the bootstrap-based single-step maxT procedure with F-statistics identifies 118 genes as having significant differences in mean expression measures between tumor molecular subtypes. This set can be identified through either adjusted p-values or cut-offs for the test statistics. Figure 6 is a plot of the F-statistics and corresponding cut-offs for the ten hypotheses (genes) with the smallest

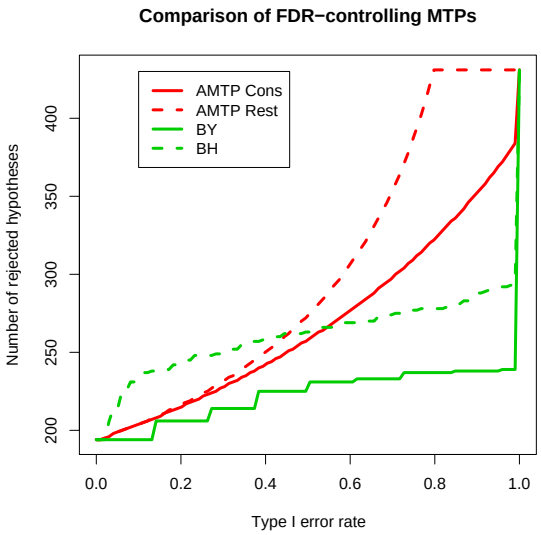

Figure 5. *B-cell vs. T-cell ALL — FDR-controlling MTPs.* Plot of number of rejected hypotheses vs. nominal Type I error rate for comparing four FDR-controlling MTPs.

adjusted p-values. Each observed F-statistic is represented by a circle and the cut-offs are represented by text indicating the corresponding nominal level (0.01 or 0.10). The plot illustrates that the F-statistics for the ten genes with the smallest adjusted p-values are much larger than expected under the null distribution. Also, cut-offs for level 0.01 and 0.10 tests are nearly identical.

```
> plot(mb.boot, which = 6, sub.caption = NULL)
```

0.4.5 Association of expression measures and time to relapse: Cox t-statistics

The bootstrap-based MTPs implemented in the main `MTP` function (`nulldist="boot"`) allow the test of hypotheses concerning regression parameters in models for which the subset pivotality condition may not hold (e.g., logistic and Cox proportional hazards models). The phenotype information in the *ALL* package includes the original remission status of the ALL patients (`remission` variable in the *data.frame* `pData(ALL)`). There are 66 B-cell ALL subjects who experienced original complete remission (`remission="CR"`) and who were followed up for remission status at a later date. We apply the single-step maxT procedure to test for a significant association between expression measures and time to relapse amongst these 66

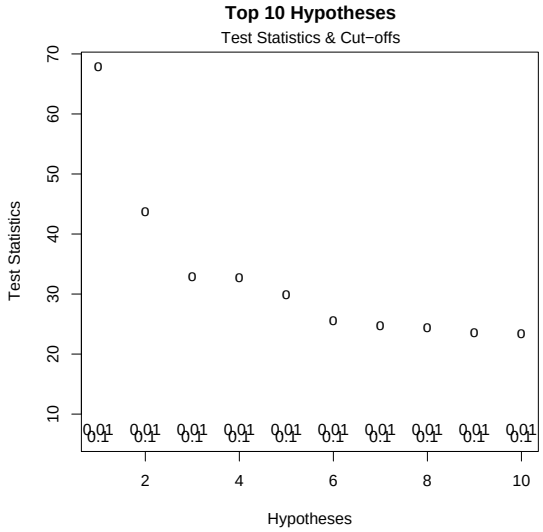

Figure 6. *Tumor molecular subtype — FWER-controlling single-step maxT MTP.* Plot of F-statistics and corresponding cut-offs for the ten genes with the smallest adjusted p-values, based on the FWER-controlling bootstrap-based single-step maxT procedure (`plot` method, `which=6`).

subjects, adjusting for sex. Note that most of the code below is concerned with extracting the (censored) time to relapse outcome and covariates from slots of the *exprSet* instance `ALL`.

```
> cr.ind <- (Bpheno$remission == "CR")
> cr.pheno <- Bpheno[cr.ind, ]
> times <- strptime(cr.pheno$"date last seen", "%m/%d/%Y") - strptime(cr.pheno$date.cr,
+     "%m/%d/%Y")
> time.ind <- !is.na(times)
> times <- times[time.ind]
> cens <- ((1:length(times)) %in% grep("CR", cr.pheno[time.ind,
+     "f.u"]))
> rel.times <- Surv(times, !cens)
> patients <- (1:ncol(BX))[cr.ind][time.ind]
> relX <- BX[, patients]
> relZ <- Bpheno[patients, ]

> cox.boot <- MTP(X = relX, Y = rel.times, Z = relZ, Z.incl = "sex",
+     Z.test = NULL, test = "coxph.YvsXZ", B = 100, get.cr = TRUE,
+     seed = seed)
```

For control of the FWER at nominal level $\alpha = 0.05$, the bootstrap-based single-step maxT procedure identifies 1 genes whose expression measures are significantly associated with time to relapse. Using the function `mget`, we examine the names of these genes.

```
> cox.diff <- cox.boot@adjp <= 0.05
> sum(cox.diff)
```

```
[1] 1
```

```
> cox.AffyID <- geneNames(filtALL)[cox.diff]
> mget(cox.AffyID, env = hgu95av2GENENAME)
```

```
$"33232_at"
[1] "cysteine-rich protein 1 (intestinal)"
```

Figure 7 is a plot of the Cox regression coefficient estimates (circles) and corresponding confidence regions (text indicating the level) for the five genes with the smallest adjusted p-values. The plot illustrates that the level $\alpha = 0.05$ confidence regions corresponding to the significant gene does not include the null value $\psi_0 = 0$ for the Cox regression parameters (red line). The confidence regions for the next four genes, do include 0.

```
> plot(cox.boot, which = 5, top = 5, sub.caption = NULL)
> abline(h = 0, col = "red")
```

0.5 Discussion

The *multtest* package implements resampling-based multiple testing procedures that can be applied to a broad range of testing problems in biomedical and genomic data analysis. Ongoing efforts involve expanding the class of MTPs implemented in *multtest*, enhancing software design and the user interface, and increasing computational efficiency. Specifically, regarding the offering of MTPs, we envisage the following new developments.

- Expanding the class of available tests, by adding test statistic closures for tests of correlations, quantiles, and parameters in generalized linear models (e.g., logistic regression).
- Expanding the class of resampling-based estimators for the test statistics null distribution (e.g., parametric bootstrap, Bayesian bootstrap), possibly using a function closure approach.
- Providing parameter confidence regions and test statistic cut-offs for other Type I error rates than the FWER.
- Implementing the new augmentation multiple testing procedures proposed in Dudoit et al. [2004] and Dudoit and van der Laan [2004],

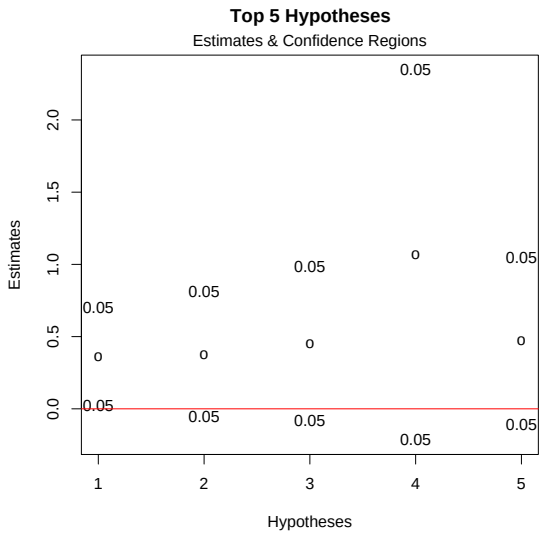

Figure 7. *Time to relapse — FWER-controlling single-step maxT MTP.* Plot of Cox regression coefficient estimates and corresponding confidence intervals for the fifteen genes with the smallest adjusted p-values, based on the FWER-controlling bootstrap-based single-step maxT procedure (`plot` method, `which=5`).

for controlling tail probabilities $Pr(g(V_n, R_n) > q)$ for an arbitrary function $g(V_n, R_n)$ of the numbers of false positives V_n and rejected hypotheses R_n.

Efforts regarding software design and the user interface include the following.

- Providing a formula interface for a symbolic description of the tests to be performed (cf. model specification in `lm`).
- Providing an `update` method for objects of class *MTP*, to facilitate the reuse of available estimates of the null distribution when implementing new MTPs.
- Extending the *MTP* class to keep track of results for several MTPs.

References

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

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple hypothesis testing under dependency. *Annals of Statistics*, 29: 1165–88, 2001.

J. Chambers. *Programming with data*. Springer Verlag, 1998.

S. Chiaretti, X. Li, R. Gentleman, et al. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103:2771–2778, 2004.

S. Dudoit, J. P. Shaffer, and J. C. Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18:71–103, 2003.

S. Dudoit and M. J. van der Laan. *Multiple Testing Procedures and Applications to Genomics*. Springer, 2004. (in preparation).

S. Dudoit, M. J. van der Laan, and M. D. Birkner. Multiple testing procedures for controlling tail probability error rates. Technical Report 166, Division of Biostatistics, University of California, Berkeley, 2004a. URL `www.bepress.com/ucbbiostat/paper166`.

S. Dudoit, M. J. van der Laan, and K. S. Pollard. Multiple testing. Part I. Single-step procedures for control of general Type I error rates. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 13, 2004b. URL `www.bepress.com/sagmb/vol3/iss1/art13`.

R. C. Gentleman, V. J. Carey, S. Dudoit, W. Huber, and R. Irizarry, editors. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer-Verlag, New York, 2005. (In preparation).

Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75:800–802, 1988.

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

S. Keleş, M. J. van der Laan, S. Dudoit, et al. Multiple testing methods for ChIP-Chip high density oligonucleotide array data. Technical Report 147, Division of Biostatistics, University of California, Berkeley, 2004. URL `www.bepress.com/ucbbiostat/paper147`.

K. S. Pollard and M. J. van der Laan. Choice of a null distribution in resampling-based multiple testing. *Journal of Statistical Planning and Inference*, 125:85–100, 2004.

M. J. van der Laan, S. Dudoit, and K. S. Pollard. Augmentation procedures for control of the generalized family-wise error rate and tail probabilities for the proportion of false positives. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 15, 2004a. URL `www.bepress.com/sagmb/vol3/iss1/art15`.

M. J. van der Laan, S. Dudoit, and K. S. Pollard. Multiple testing. Part II. Step-down procedures for control of the family-wise error rate. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 14, 2004b. URL `www.bepress.com/sagmb/vol3/iss1/art14`.

Z. Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62:626–633, 1967.

P.H. Westfall and S.S. Young. *Resampling-based multiple testing: examples and methods for p-value adjustment*. John Wiley and Sons, 1993.

Table 1. *Multiple hypothesis testing flowchart.*

Provide dataset
`MTP` arguments: `X`, `W`, `Y`, `Z`, `Z.incl`, and `Z.test`
⇓
Define parameters of interest, $\psi(m)$
⇓
Define null and alternative hypotheses, $H_0(m)$ **and** $H_1(m)$
⇓
Specify test statistics, $T_n(m)$
`MTP` arguments: `test`, `robust`, `standardize`, `alternative`, and `psi0`
⇓
Estimate test statistics null distribution, Q_{0n}
`MTP` arguments: `nulldist` and `B`
⇓
Select Type I error rate, $\theta(F_{V_n,R_n})$
`MTP` arguments: `typeone` and `alpha` (and also `k` and `q`)
⇓
Apply MTP
`MTP` argument: `method`

FWER	$Pr(V_n > 0)$	Single-step maxT procedure (Procedure 2) Single-step minP procedure (Procedure 3) Step-down maxT procedure (Procedure 4) Step-down minP procedure (Procedure 5)
gFWER	$Pr(V_n > k)$	Single-step $T(k+1)$ procedure (Procedure 2) Single-step $P(k+1)$ procedure (Procedure 3) Augmentation procedure (Procedure 6)
TPPFP	$Pr(V_n/R_n > q)$	Augmentation procedure (Procedure 7)
General	$\theta(F_{V_n})$	Single-step common-cut-off procedure (Procedure 2) Single-step common-quantile procedure (Procedure 3)

⇓
Summarize results
adjusted p-values, rejection regions, and confidence regions
`MTP` arguments: `get.adjp`, `get.cutoff`, and `get.cr`

Table 2. *Type I and Type II errors in multiple hypothesis testing.*

		Null hypotheses not rejected	Null hypotheses rejected	
Null hypotheses	true	$\lvert\mathcal{R}_n^c \cap \mathcal{H}_0\rvert$	$V_n = \lvert\mathcal{R}_n \cap \mathcal{H}_0\rvert$ (Type I errors)	$h_0 = \lvert\mathcal{H}_0\rvert$
	false	$U_n = \lvert\mathcal{R}_n^c \cap \mathcal{H}_1\rvert$ (Type II errors)	$\lvert\mathcal{R}_n \cap \mathcal{H}_1\rvert$	$h_1 = \lvert\mathcal{H}_1\rvert$
		$M - R_n$	$R_n = \lvert\mathcal{R}_n\rvert$	M