Tutorial in Biostatistics

Statistics
in Medicine

Received 4 August 2015, Accepted 5 July 2016 Published online in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/sim.7064

Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials

Ilya Lipkovich,[a*†] Alex Dmitrienko[b] and Ralph B. D'Agostino Sr.[c]

It is well known that both the direction and magnitude of the treatment effect in clinical trials are often affected by baseline patient characteristics (generally referred to as biomarkers). Characterization of treatment effect heterogeneity plays a central role in the field of personalized medicine and facilitates the development of tailored therapies. This tutorial focuses on a general class of problems arising in data-driven subgroup analysis, namely, identification of biomarkers with strong predictive properties and patient subgroups with desirable characteristics such as improved benefit and/or safety. Limitations of ad-hoc approaches to biomarker exploration and subgroup identification in clinical trials are discussed, and the ad-hoc approaches are contrasted with principled approaches to exploratory subgroup analysis based on recent advances in machine learning and data mining. A general framework for evaluating predictive biomarkers and identification of associated subgroups is introduced. The tutorial provides a review of a broad class of statistical methods used in subgroup discovery, including global outcome modeling methods, global treatment effect modeling methods, optimal treatment regimes, and local modeling methods. Commonly used subgroup identification methods are illustrated using two case studies based on clinical trials with binary and survival endpoints. Copyright © 2016 John Wiley & Sons, Ltd.

Keywords: clinical trials; exploratory subgroup analysis; biomarker analysis; data mining; multiplicity control.

1. Introduction

The general topic of subgroup analysis has attracted much attention in the clinical trial community. An increasing number of papers deal with statistical issues arising in the analysis of patient subgroups in late-stage clinical drug development. These subgroups are defined based on the baseline values of demographic, clinical, genomic, and other covariates that will be referred to as *biomarkers* in this tutorial. In addition, the U.S. Food and Drug Administration and European Medicines Agency have recently released guidance documents that discuss regulatory and clinical and statistical approaches to subgroup analysis [1, 2].

The need for a data-driven evaluation of patient subgroups may arise in various contexts and initiated by different stakeholders [3]:

- A sponsor is interested in 'salvaging' an experimental treatment following a failed Phase III trial by identifying a subgroup with a substantial treatment benefit.
- A sponsor is interested in identifying a subset of 'super-responders' in a successful Phase III trial.
- A policy maker is interested in identifying *optimal treatment regimes* (OTR) as rules for assigning one of the treatments available on the market to a patient with a specific biomarker profile.
- A regulatory agency needs to investigate whether a label restriction should be issued due to *inconsistent treatment effects* of a novel treatment in a setting where the overall population effect may be entirely driven by a highly significant effect in a single subgroup.
- A regulatory agency plans to investigate whether a label restriction may need to be imposed due to an *unacceptable safety profile* in a certain subgroup.

[a] *Quintiles, Inc., Durham, NC, U.S.A.*
[b] *Mediana, Inc., Overland Park, KS, U.S.A.*
[c] *Boston University, Boston, MA, U.S.A.*
**Correspondence to: Ilya Lipkovich, Quintiles Inc., 4820 Emperor Blvd, Durham, NC 27703, U.S.A.*
[†] *E-mail: ilya.lipkovich@quintiles.com*

It is important, however, to understand that, while different contexts may and often would have different regulatory implications, this may not necessarily require significant modifications of the underlying statistical methodology. For example, the same class of subgroup investigation methods can be applied to all of the settings listed previously.

To help organize available statistical approaches to biomarker evaluation and subgroup analysis in late-stage trials, the following simple classification scheme is commonly used in the clinical trial literature:

- *Confirmatory subgroup analysis* deals with a small number of prospectively defined subsets of the overall patient population.
- *Exploratory subgroup analysis* focuses on subgroup assessments that are performed in a post-hoc manner.

This classification scheme may be overly simplistic, and several extensions have been proposed in the literature [4]. This tutorial considers the following expanded scheme that was proposed in a recent survey of current subgroup analysis practices [5]:

- *Confirmatory subgroup analysis*: statistical methods aimed mainly at controlling the Type I error rate in Phase III clinical trials with a small number of pre-specified subgroups (1–2 subgroups).
- *Exploratory subgroup evaluation*: analysis of a relatively small number of pre-specified subgroups (10–20 subgroups) that focuses mostly on treatment-by-covariate interactions and consistency assessments.
- *Post-hoc subgroup evaluation*: post-hoc assessments of the treatment effect across small sets of subgroups (10–20 subgroups) that include responses to regulatory inquiries, analysis of safety issues, post-marketing activities in Phase IV trials, and assessment of heterogeneity in multi-regional studies.
- *Subgroup discovery*: Statistical methods aimed at selecting most promising subgroups with enhanced efficacy or desirable safety from a large pool of candidate subgroups (hundreds of subgroups). These methods employ data mining/machine learning algorithms to help inform the design of future trials.

Confirmatory subgroup analysis and subgroup discovery define two extremes in this classification scheme. Exploratory subgroup evaluation and post-hoc subgroup evaluation occupy the middle ground between the two extremes. The latter approaches typically rely on naive or haphazard statistical strategies; for example, it is not always clear how the candidate hypotheses were selected, how the decision rules were justified, and what the operating characteristics of these approaches are. The differentiating feature between exploratory and post-hoc subgroup assessments is that the former are pre-specified, for example, in the trial's statistical analysis plan, whereas the latter are driven by unanticipated findings *after* the clinical trial data have been collected and analyzed.

All types of subgroup analysis, except the first one, are typically implemented as strategies that explicitly (or implicitly, as is often the case with post-hoc analyses) incorporate data-driven elements. In this tutorial, we will emphasize the need of developing *principled* data-driven strategies for subgroup identification and evaluation where all data-driven elements are explicitly stated and implemented using solid statistical principles (Section 2). Our goal is to provide practitioners with a broad class of statistical methods that can be used as building blocks of such principled strategies. While many of these methods may have been motivated by (and may be more applicable to) subgroup discovery, methods considered in this tutorial would generally apply to most settings where data-driven subgroup evaluation strategies are employed. Occasionally, we will refer to such strategies as 'exploratory' in the most general sense rather than implying the class of exploratory subgroup analyses defined previously.

This tutorial will focus on strategies for subgroup assessment where subgroups are defined by one or more biomarkers selected from a set of candidate biomarkers. The biomarkers of interest are expected to exhibit predictive abilities, that is, they help identify subsets of the trial population with desirable properties such as an improved efficacy profile. By contrast, purely prognostic biomarkers can be used only for selecting patients who experience improvement or worsening irrespective of the treatment assignment. In general, identification of patient subgroups in clinical trials relies on biomarker identification, and thus, we will often refer to such activities as 'subgroup/biomarker identification'. We note that other approaches to evaluating treatment effect heterogeneity that do not involve biomarkers, for example, latent mixture models, can be used in clinical trial settings. Such methods are beyond the scope of this tutorial.

It should be mentioned that new types of multi-stage biomarker-driven designs have been proposed recently [6–9]. These designs are aimed at identifying biomarker signatures that help predict treatment response at earlier stages of the trial and performing various adaptations such as modifying the patient population at later stages. These developments are outside the scope of this tutorial as we only focus on algorithms for subgroup/biomarker identification rather than on utilizing this information to design clinical trials. This tutorial does not discuss assessment of the so-called *dynamic treatment regimes* or policies, for example, via SMART trials or analysis of observational data, which begin playing a significant role in personalized medicine. Methods in this class deal with estimating optimal treatment policies based on the data where the treatment can be assigned to the same patient at multiple decision points [10, 11]. The present review is limited to settings where the treatment decision is made at a single assessment point, that is, at the trial's baseline.

This tutorial is structured as follows. Section 2 defines the key principles of subgroup/biomarker identification that provide a foundation for a principled approach to exploratory subgroup analysis. Two case studies that are used throughout the tutorial to illustrate commonly used subgroup identification methods are introduced in Section 3. An overview of general approaches to biomarker analysis and subgroup discovery methods, including a general classification scheme, is provided in Section 4. Section 5 discusses the limitations of basic methods based on univariate and tree regression models. A detailed review of the four classes of advanced methods for evaluating predictive biomarkers and identification of associated patient subgroups is presented in Sections 6 (global outcome modeling methods), 7 (global treatment effect modeling methods), 8 (optimal treatment regimes), and 9 (local modeling methods). Selected methods from each class are illustrated in Section 10. Section 11 provides general recommendations and a summary of key features of the exploratory subgroup analysis methods highlighted in this tutorial.

2. Key principles of subgroup/biomarker discovery

In this section, we introduce key considerations in exploratory subgroup analysis or subgroup discovery and define *principled subgroup discovery*. It is well known and understood based on numerous simulation studies and purely theoretical arguments that 'undisciplined' subgroup exploration or 'data-dredging' may lead to substantial inflation of the Type I error rates and gross exaggeration of treatment effects in the selected subgroups of patients. Therefore, vast literature was generated under the heading of 'best practices for subgroup analysis' containing checklists of subgroup analysis 'do's' and 'do-not's' that are meant to improve quality of exploratory subgroup analysis and prevent researches from committing various errors. For example, a checklist with 25 rules was proposed in [12], Rothwell [13] developed 21 rules, and Sun *et al*. [14] listed seven existing and four additional criteria for assessing the credibility of subgroup analysis. The general theme of these guidelines can be expressed in several recurring items:

(1) Subgroups need to be pre-specified.
(2) Subgroups need to be biologically plausible.
(3) All significance tests should be multiplicity adjusted.
(4) No testing in a subgroup should be performed unless the associated interaction test is significant.
(5) It is often suggested that no testing in a subgroup should be performed unless the overall effect is significant.
(6) Invariably, the guidelines advise that the findings of subgroup analysis should be 'interpreted with caution'.

While this *guideline-driven* approach might have accumulated much of practical wisdom, some of the recommendations are hard to operationalize and may not be consistent with the general scientific principles/goals of a specific investigation. For example, Requirement 1 may limit the scope of scientific inquiry. As noted by Berry [15], '…there's something unscientific about requiring hypotheses to be specified in advance. Science would proceed very slowly if scientists never took data at face value'. Requirement 4 assumes that each subgroup is evaluated in a one-predictor-at-a-time fashion via a parametric model including the main effect and interaction term, which is a rather narrow and often inefficient approach to modeling. Requirement 5 ignores the very spirit of personalized medicine, which aims at recovering possible heterogeneity of the treatment effect in a clinical trial. Most importantly, the guideline-driven approach seems to be disconnected from the world of statistical science (in particular, statistical literature on model selection) and does not use principled methods for assessing and incorporating uncertainty associated with decision making in exploratory subgroup analysis.

It is important to contrast the guideline-driven and *data-driven* approaches to subgroup analysis. The latter recognizes subgroup investigation as a special case of *model selection* and thrives on statistical methodologies for model selection developed within the machine/statistical learning and related fields, including multiple comparisons and causal inference. The common thread in the data-driven approach is that it is the entire subgroup selection and evaluation strategy, rather than the final set of patient subgroups, that needs to be pre-specified. Note that this strategy is data-driven in two aspects. First, it is similar to any model selection method in that it identifies a few final models from a large set of candidate models within the model space. This approach supports the goal of identifying best subgroups as members of a broadly defined collection of candidate subgroups. Second, subgroup search strategies often include meta-parameters or tuning parameters that are not pre-defined but need to be estimated from the data. These parameters control the complexity of the model space or help the user navigate through the model space to restrict it to a subset of models supported by the data. However, the entire strategy is pre-specified in the sense that the model space and methods for estimating meta-parameters are fixed upfront.

In what follows, we present a short review of the key features of the principled subgroup analysis methods to help the reader navigate in the ocean of methods recently proposed in the literature.

Evaluating the Type I error rate/false discovery rate for the entire subgroup search strategy. Until recently, it was almost an established principle that the concept of statistical significance does not apply to machine learning and data mining where the hypotheses tested are not pre-specified but 'data-driven' or 'random'. However, more recently procedures that control for multiplicity (whether in the sense of strong familywise error rate control or false discovery rate) have been developed for some machine learning methods. For example, Meinshausen *et al*. [16] developed a significance testing approach for high-dimensional regression via multiple random data splits, and Lockhart *et al*. [17] developed an analytical procedure for testing coefficients in the lasso regression under certain assumptions. Furthermore, nowadays, many data mining procedures include statistical significance as its core element perhaps combined with other concepts such as complexity and reproducibility. Just to list a few examples, see stability selection [18], adaptive signature and cross-validated designs [19, 20] and the method for qualitative interaction assessment [21]. As for any method used in confirmatory clinical development programs, controlling an appropriately defined error rate (e.g., Type I error rate [22]) is very important for subgroup identification procedures. Therefore, it should be emphasized that multiplicity control with respect to the entire subgroup identification strategy is needed and approximate multiplicity control can be implemented by applying resampling-based methods. However, as in any exploratory analysis, the Type I error rate does not have to be controlled at the 'magic' 0.05 level, rather what is needed is a general sense of how likely the apparent treatment effects in selected subgroups could have been attributed to chance alone. As we will argue next, multiplicity control should be used in conjunction with complexity control and adjustments for selection bias.

Using complexity control to prevent data overfitting. Because the model space may be quite large, 'uncontrolled' or greedy search is likely to result in data overfitting, that is, generating subgroups that look very promising when evaluated with the same dataset that was searched but have a low chance to be confirmed with future data. Applying multiplicity adjustments following subgroup selection is an important but insufficient step, as it would not help find the right covariates 'after the fact'. As with any data mining/machine learning method, complexity control should be built into the process of model selection, rather than implemented after model selection, when possibly wrong subgroups, for example, subgroups based on irrelevant biomarkers, have already been identified. On the other hand, when complexity control is built into the model selection process (e.g., via penalized likelihood), it alleviates multiplicity burden and results in a less severe multiplicity adjustment compared with a greedy selection method with no complexity control. Therefore, multiplicity and complexity control in subgroup search are inter-related concepts and should be used in combination.

Controlling (reducing) selection bias when defining candidate subgroups. Subgroups of patients are often defined by examining all possible 'splits' of the overall population using thresholds based on specific values of candidate covariates. Because different covariates may have drastically different sets of unique values, which results in different numbers of possible subgroups for individual covariates, it is important to ensure that the probability of falsely selecting an irrelevant subgroup does not depend on the number of possible splits for different covariates. Otherwise, covariates with larger sets of unique values will have an advantage over covariates with smaller sets. This problem was studied extensively in the context of recursive partitioning algorithms by Loh and Shih [23] and Hothorn *et al*. [24].

Accounting for uncertainty of the entire subgroup search strategy. Many subgroup identification procedures are inherently multi-stage, and it is important to account for the uncertainty associated with the entire subgroup search procedure when evaluating Type I error rates and standard errors of treatment effect estimated within individual subgroups.

Reproducibility assessment is concerned with the probability of reproducing subgroup(s) identified on *training* data with *future* data. Clearly, a subgroup that has a very low chance to be confirmed with the future data is of little use. Resampling procedures such as the bootstrap and cross-validation (CV) followed by replicating the entire subgroup search strategy are often used for such assessments. Subgroups that are stable in the sense that very similar subgroups are repeatedly seen in multiple re-samples will have higher credibility for being reproduced in the future data. The degree of similarity among different subgroups can be assessed based on subgroup descriptors or proximity of subsets of patients included in each subgroup. The proximity of two subgroups can be assessed using appropriate similarity measures, for example, Jaccard's similarity index.

Obtaining 'honest' estimates of treatment effects in identified subgroups. Once subgroups have been identified, the analyst is facing the challenge of obtaining unbiased or 'honest' estimates of the effect sizes that should be anticipated in the future data. Resampling methods may often be the only feasible approach within the frequentist framework. For example, the virtual twins (VT) method presented in Section 6.4 relies on an honest estimate of the treatment effect expected to be found in the future data within the identified subgroup (in excess to that expected in the overall population) that is constructed using k-fold CV and parametric and non-parametric bootstrap procedures. Bayesian approaches for estimating effect sizes in the selected patient subgroups via hierarchical modeling or model averaging may provide attractive alternatives [8, 25].

The users of subgroup analysis methods are also interested in determining objective criteria to help select methods that perform better on their datasets. Some authors may focus on evaluation criteria that exaggerate the advantages of their methods. Without offering the 'final judgment', we also suggest key criteria for classifying different biomarker/subgroup identification strategies that were used in various publications (Table XV).

3. Case studies

This section introduces two clinical trial examples that will be used throughout this tutorial to illustrate key subgroup discovery methods and their software implementation. Both case studies provide examples of a retrospective approach to biomarker discovery and subgroup identification. Note, however, that subgroup exploration will pursue different goals in the two case studies. Because the overall treatment effect was negative in Case study 1, subgroup identification procedures could be applied in an attempt to 'rescue' this failed trial and identify one or more subgroups of patients who experience a beneficial effect. By contrast, a positive treatment difference was observed in the overall population in Case study 2. The trial's sponsor would therefore be interested in uncovering subgroups with enhanced efficacy, that is, subgroups of patients with effect sizes that exceed the effect size in the overall population.

3.1. Case study 1

A Phase III trial was conducted to examine the efficacy and safety profiles of a novel treatment for severe sepsis. A two-arm unbalanced design was utilized in the trial (novel treatment versus standard of care) with 317 patients in the treatment arm and 153 patients in the control arm. The primary endpoint in the trial was all-cause survival at 28 days (survived versus deceased), and no treatment benefit was detected in the overall population of patients. In fact, the novel treatment was associated with a lower 28-day survival rate compared with the control, and the one-sided treatment effect p-value was 0.83.

The trial's sponsor was interested in a comprehensive characterization of the treatment effect by a number of important biomarkers to verify whether a beneficial effect may exist in a subset of the overall population. The candidate set of biomarkers included demographic and clinical variables such as patient's age and disease characteristics (Table I). For example, the acute physiology and chronic health evaluation II score is a widely used tool for predicting the probability of survival in severe sepsis patients [26]. This score is known to have strong prognostic abilities, that is, it helps identify patients with a poor prognosis regardless of the treatment assignment, but its predictive properties are unknown, that is, it is not clear whether or not it can be used for selecting treatment responders.

It is important to note that all eleven biomarkers listed in Table I are continuous; however, their distributions may be highly skewed with a large number of outliers. In addition, the dataset contains missing values for some covariates. The percent of missing observations does not exceed 5% for any biomarker (as shown in parentheses): X_{10} (4.9%), X_{11} (3.8%), X_9 (3.4%), X_3 (0.6%), and X_5 (0.2%).

Table I. Candidate biomarkers in Case study 1.

Biomarker	Description	Type	Range
X_1	Patient's age (years)	Continuous	(33.2, 93.3)
X_2	Time from the first organ failure to the start of drug administration (hours)	Continuous	(10, 3776)
X_3	Baseline platelets (1000/mm^3)	Continuous	(45, 650)
X_4	Baseline SOFA score (unitless)	Continuous	(3, 17)
X_5	Baseline creatinine (mg/dL)	Continuous	(1, 20)
X_6	Number of organ failures at baseline (unitless)	Continuous	(0, 5)
X_7	Pre-infusion APACHE II score (unitless)	Continuous	(19, 48)
X_8	Baseline GLASGOW coma scale score (unitless)	Continuous	(3, 15)
X_9	Baseline serum IL-6 concentration (pg/mL)	Continuous	(37, 296550)
X_{10}	Activity of daily living score at baseline (unitless)	Continuous	(0, 12)
X_{11}	Baseline bilirubin (mg/dL)	Continuous	(0.4, 20.4)

APACHE, acute physiology and chronic health evaluation

Table II. Candidate biomarkers in Case study 2.

Biomarker	Description	Type	Values
X_1	Patient's sex	Nominal	1 (Male), 2 (Female)
X_2	Patient's race	Nominal	1 (Asian), 2 (Black), 3 (White)
X_3	Cytogenetic category	Ordinal	1 (Very good), 2 (Good) 3 (Intermediate), 4 (Poor), 5 (Very poor)
X_4—X_{12}	Cytogenetic markers 1 through 9	Nominal	0 (Absent), 1 (Present)
X_{13}	Prognostic score for myelodysplastic syndromes risk assessment (IPSS-R)	Ordinal	1 (Low), 2 (Intermediate), 3 (High) 4 (Very high)
X_{14}	Outcome for patient's prior therapy	Nominal	1 (Failure), 2 (Progression), 3 (Relapse)

3.2. *Case study 2*

This case study deals with a Phase III clinical trial in patients with hematological malignancies. The patients were randomly assigned to an experimental therapy plus best supporting care (treatment arm) or best supporting care (control arm). The total sample size was 599 patients (303 patients in the treatment arm and 296 patients in the control arm). The primary endpoint in the trial was overall survival, and the treatment effect was expressed using a hazard ratio. The hazard ratio was borderline clinically and statistically significant in the overall population (hazard ratio = 0.85 with one-sided $p = 0.0367$). The trial's sponsor decided to conduct a subgroup search to investigate subsets of the overall population with an improved benefit/risk ratio. The findings could potentially lead to a decision to restrict the patient population in subsequent trials.

A number of potentially predictive biomarkers were identified by the sponsor to support a comprehensive assessment of treatment effect heterogeneity across important subsets of the overall population. The final set of candidate biomarkers selected in Case study 2 is defined in Table II. This list includes demographic variables, clinical variables related to baseline disease severity, and cytogenetic markers.

There were no continuous biomarkers in the candidate set. Two biomarkers were ordinal (X_3 and X_{13}), and the other biomarkers were measured on a nominal scale. As in Case study 1, there were missing/unknown values in this dataset. Specifically, the IPSS-R score (X_{13}) was missing for eight patients.

4. Overview of biomarker evaluation and subgroup discovery methods

This section defines the problem of biomarker evaluation and subgroup identification in clinical trials and introduces a general classification of statistical methods utilized in exploratory subgroup analysis. This taxonomy of emerging subgroup analysis methods will play an important role in Sections 6 through 9.

4.1. General setting

To facilitate the exposition of available approaches to subgroup identification in clinical trials, we will define a general subgroup discovery setting and introduce notation, which will be used throughout this tutorial. Upper-case letters will be used to refer to random variables, for example, X and Y, and lower-case letters to refer to individual observations, for example, $\mathbf{x}$. To simplify notation, lower-case letters without patient-specific subscripts may be used to refer to the observed values of random variables for an arbitrary patient, for example, $f(\mathbf{x}, t)$. However, when referring to individual outcomes or covariates as random variables, we will always use upper-case letters. For example, when computing the expectation with respect to the outcome for the ith patient, the outcome will be denoted by Y_i, and similarly, the associated vector of random covariates will be denoted by as $\mathbf{X}_i$.

Consider a clinical trial that was conducted to evaluate the efficacy and safety of an experimental therapy versus a control. Let y_i denote the outcome variable (Y) evaluated on the ith patient ($i = 1, \dots, n$), which can be continuous, binary, or based on the time to an event of interest such as death or disease progression (in the latter case, the outcome may be censored and a censoring indicator will need to be introduced). Suppose that a larger value of the outcome variable indicates a beneficial effect, for example, a larger value of Y is associated with a greater improvement in the continuous endpoint setting or longer survival in the time-to-event setting. Here, n is the total sample size in the trial. Further, let t_i denote the study arm indicator (T) for to the ith patient, that is, the patient was assigned to the control arm if $t_i = 0$ and experimental treatment arm if $t_i = 1$.

Suppose that p candidate biomarkers, denoted by $X_1, \dots, X_p$, were studied in this clinical trial. Let $\mathbf{x}_i = \{x_{i1}, \dots, x_{ip}\}$ denote a vector of observed biomarker values for the ith patient evaluated prior to the initiation of treatment. A subgroup $S(\mathbf{X})$ is defined by a rule, which selects a subset of the overall population based on the vector $\mathbf{X}$. For example,

$$S(\mathbf{X}) = I\{X_1 > c\}$$

is composed of all patients with elevated levels of the biomarker X_1. Applying this rule to a particular dataset, we obtain the estimated patient subgroup (which sometimes will be emphasized by placing a hat over the S):

$$\widehat{S}(\mathbf{X}) = \{x_{i1} > c,\ i = 1, \dots, n\}.$$

Let $f(\mathbf{x}, t) = E(Y|\mathbf{X} = \mathbf{X}, T = t)$ denote the expected response of a patient as a function of the vector $\mathbf{X}$ and treatment assignment T evaluated at points $\mathbf{x}$ and t, respectively. This is the familiar Q-function from the literature on OTRs (see Section 8 for details). Further, let $z(\mathbf{X})$ denote the treatment contrast defined at the patient's level, that is, as the function of random covariate vector $\mathbf{X}$,

$$z(\mathbf{X}) = g(f(\mathbf{X}, 1), f(\mathbf{X}, 0)),$$

where $g(\cdot)$ is a monotone function of its arguments. For example, if the outcome variable is continuous, $z(\mathbf{x})$ may be defined as the treatment difference, that is,

$$z(\mathbf{X}) = f(\mathbf{X}, 1) - f(\mathbf{X}, 0).$$

In this case, the expected outcome function can be written as

$$f(\mathbf{X}, T) = f(\mathbf{X}, 0) + z(\mathbf{X})T,$$

In a more general case, the outcome function can be represented [27] as

$$f(\mathbf{X}, T) = g(h(\mathbf{X}) + l(z(\mathbf{X})T)),$$

where $h(\cdot)$ is an arbitrary 'baseline' function of the covariate vector, $l(\cdot)$ is a monotone function, and $z(\cdot)$ summarizes the individual treatment effect. In light of this representation, the distinction between *prognostic* and *predictive* biomarkers becomes very simple. Prognostic biomarkers are defined as those that contribute only to $h(\mathbf{X})$ (i.e., 'main effects'), whereas predictive biomarkers also contribute to $z(\mathbf{X})$. All the relevant information about predictive biomarkers is included in the individual treatment contrast $z(\mathbf{X})$, and the success of personalized medicine hinges on recovering this function. By contrast, the problem of estimating the main effects in the outcome model, that is, $h(\mathbf{X})$, is not relevant in personalized medicine applications.

It is worth mentioning the concept of *potential outcomes*, which plays an important role in several classes of subgroup search methods. Considering the general setting introduced previously, two potential outcomes, denoted by $\tilde{Y}_i(0)$ and $\tilde{Y}_i(1)$, are defined for the ith patient. These random variables represent hypothetical outcomes that would have been realized had a random patient been assigned to the treatment $T = 0$ or $T = 1$, respectively. The consistency assumption states that the observed outcome is the same as the potential outcome under the treatment actually received. For example, if the ith patient was allocated to the control arm, $\tilde{Y}_i(0) = Y_i$ and $\tilde{Y}_i(1)$ is unobserved but can be estimated from the data to predict the patient's outcome, if the patient were allocated to the treatment arm. It is easy to see that, assuming consistency, the observed outcome as a random variable is connected with the two potential outcomes as follows:

$$Y_i = \tilde{Y}_i(0)(1 - T_i) + \tilde{Y}_i(1)T_i, \; i = 1, \dots, n.$$

The outcome function $f(\mathbf{X}, T)$ with $T = 0$ or 1 for any given covariate vector $\mathbf{X} = \mathbf{x}$ can be estimated as expected value of the corresponding potential outcome, $\tilde{Y}(0)$ or $\tilde{Y}(1)$, conditional on $\mathbf{X} = \mathbf{x}$. This requires the so-called 'stable unit treatment value assumption', which implies that: (i) the treatment status of any patient does not affect the potential outcomes of the other patients; and (ii) there is no hidden variation in the treatment (such as if some patients were taking adjunct medication that may affect the potential outcomes under their main treatment) [28].

4.2. *Two frameworks of personalized medicine*

Subgroup analysis procedures developed for personalized medicine applications are commonly conceptualized within the following two frameworks:

- The first framework aims at identifying the right patient for a given treatment. To give an example, consider a trial's sponsor who is interested in developing a 'salvaging strategy'. This includes identification of subgroups of patients who may still benefit from an experimental therapy versus a control given that the therapy provides minimal or no benefit in the overall population.
- The second framework deals with identifying the right treatment for a patient. Consider, for example, the problem of finding the *OTR* or policy for a given subpopulation. This framework appears to represent the society's view or public policy maker's view of personalized medicine.

Although the two frameworks are closely related to each other, there are important differences, both conceptual and statistical. The two approaches are illustrated graphically in the left-hand and right-hand panels of Figure 1, respectively. The thick and thin lines represent the expected outcome in the treatment ($T = 1$) and control ($T = 0$) arms in the figure. Larger values of the outcome indicate a beneficial effect. The horizontal axis represents a continuous biomarker X. Note that the two lines have non-zero slopes and are not parallel, which indicates that the biomarker is both *prognostic* and *predictive*.

One of the differences in the statistical formulation of the two approaches is that the first approach entails searching for predictive biomarkers exhibiting *quantitative interactions* and modifying the overall treatment effect, so that the expected treatment contrast $z(\mathbf{x})$ in the identified subgroup S is substantially larger compared with that in the overall population. If the overall effect is not significant, the trial's sponsor may wish to 'salvage' the treatment by looking for a 'bump' in the expected treatment difference as a function of the biomarker level (or, in general, multiple biomarkers). In addition, the overall population effect may be quite large, and the sponsor may be interested in identifying a subset of 'super-responders' by applying the same approach. An example of such a bump in the treatment arm is seen in the left-hand panel and suggests that patients in the subgroup $S = \{X > c_1\}$ are likely to experience a beneficial effect, whereas little treatment benefit is observed in the complementary subgroup.

The subgroup of interest may be defined formally by using a condition based on an appropriate threshold value, for example,

$$z(\mathbf{X}) > \delta \text{ implies } \mathbf{X} \in S.$$

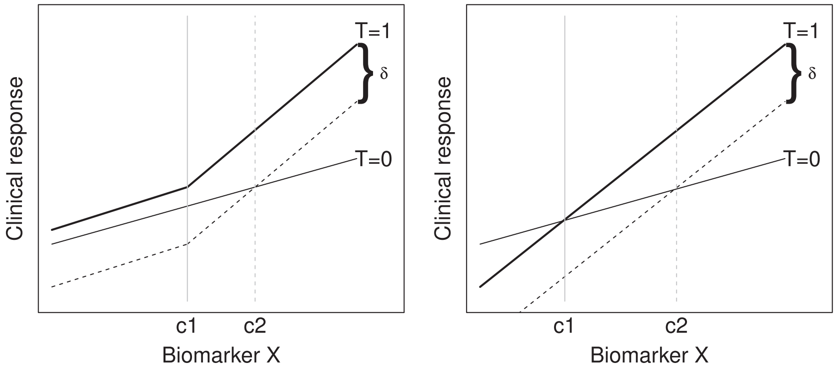

Figure 1. Two frameworks of personalized medicine: identification of the best patients for a treatment (left-hand panel) versus identification of the best treatment for a patient (right-hand panel). The thick and thin solid lines in each panel represent the expected outcome function for the experimental treatment ($T = 1$) and control ($T = 0$), respectively. The dotted lines are obtained by subtracting a fixed amount of δ from the outcome in the treatment arm, where δ can be interpreted as a clinically important difference or the 'treatment burden'.

Here, $\delta > 0$ is a clinically important difference, which can be defined as an absolute value or as a multiple of the treatment effect in the overall population, that is, $\delta = sEz(\mathbf{X})$, where $s > 1$ is a pre-specified constant and the expectation is taken with respect to the distribution of **X**. This subgroup is illustrated in the left-hand panel of Figure 1 with the vertical line drawn at c_2.

Note that a subgroup of interest may also be defined using a weaker condition that the expected treatment contrast within the subgroup exceeds a given threshold value, that is,

$$E(z(\mathbf{X})|\mathbf{X} \in S) > \delta.$$

While this criterion may appear reasonable, it can be used only as a method of subgroup *validation*, assuming that the subgroups have been produced by a procedure that ensures homogeneity of treatment effect in the selected subgroups. However, using the aforementioned condition to *define* a patient subgroup would exhibit undesirable properties; for example, the condition can be easily met by a heterogeneous subpopulation that combines patients with extremely high and modest values of the treatment contrast.

The second approach requires the identification of predictive biomarkers that exhibit *qualitative interactions* leading to different optimal treatment strategies for different types of patients. With a qualitative interaction, patients with certain values of the biomarker, known as *biomarker-positive patients*, experience a pronounced treatment effect, whereas patients in the complementary subgroup, known as *biomarker-negative patients*, benefit from the control. A natural way to define the subgroups would be to partition the entire population into the following two subsets:

- subset of patients who benefit from the experiential treatment, that is, patients with $z(\mathbf{X}) > 0$; and
- complementary subset of patients, that is, patients with $z(\mathbf{X}) \leqslant 0$.

An example of partitioning the population into two subgroups is provided in the right-hand panel of Figure 1 with a vertical line drawn at c_1. Similarly to the first framework presented previously, this condition can be generalized by introducing a threshold value δ and defining biomarker-positive patients as the patients in the subgroup $\{z(\mathbf{X}) > \delta\}$. Using the right-hand panel of Figure 1, a subgroup of this kind can be defined as $S = \{X > c_2\}$.

It is often believed and argued that qualitative treatment-by-covariate interactions are rare, and therefore, the second framework is not very relevant for drug development. Note, however, that the quantitative interaction shown with the two non-crossing expected outcome lines in the left-hand panel of Figure 1 becomes qualitative after we introduce the threshold value of δ, which is equivalent to shifting the outcome line in the treatment arm ($T = 1$) downward until it becomes the dashed line that crosses the outcome line associated with the control arm. Indeed, one can argue that the threshold value may be naturally interpreted as the 'treatment burden' either for the patient (e.g., side effects associated with the new treatment) or for the society (e.g., additional costs associated with the new treatment). This leads to a 'utility index', which may be defined as $U = Y - \delta$ for the patients who undergo the new treatment and $U = Y$ for the patients who are treated with the current standard of care. While the original outcome variable may interact with the treatment quantitatively, the utility index U is more likely to exhibit a qualitative interaction. This simple example demonstrates that qualitative interactions are, in fact, quite relevant in clinical trial settings.

4.3. Principled approaches to subgroup discovery

The notation and concepts introduced in Sections 4.1 and 4.2 provide a foundation for defining a general classification scheme for advanced methods for evaluation of predictive biomarkers and identification of associated subgroups of patients. These methods support the 'discovery spirit' of personalized medicine in the sense that they focus on data-driven subgroup exploration and, at the same time, they rely on a disciplined approach, which utilizes the key principles introduced in Section 2. The subgroup discovery methods utilize analytic strategies with known operating characteristics and thus result in reliable statistical inferences.

It is instructive to compare and contrast the principled subgroup discovery framework with basic/naive methods for biomarker evaluation. These approaches are often employed in the context of exploratory and post-hoc subgroup evaluation with respect to a small set of baseline factors (Section 1), and it is also quite common to use them in subgroup discovery applications with large sets of biomarkers. While basic methods may provide some useful insights into predictive biomarkers, these methods rely on rather ad-hoc procedures whose operating characteristics and statistical properties may be suboptimal and/or hard or even impossible to assess. In actual clinical practice, such basic methods are often combined in complex multi-stage strategies involving human intervention and 'fine-tuning', which results in procedures that are impossible to formalize, replicate, and evaluate. For this reason, these 'popular' approaches are becoming less relevant within the personalized medicine framework considered in this tutorial. As shown in multiple recent publications, including [29, 30], an application of basic approaches to biomarker evaluation in complex subgroup discovery problems leads to spurious results. Recently, Ruberg and Shen [31] emphasized the need for 'disciplined' subgroup search methods for personalized medicine and distinguished them from traditional ad-hoc subgroup assessments. Examples of basic methods based on univariate and tree regression models will be provided in Section 5.

Lipkovich and Dmitrienko [32] proposed a general taxonomy of principled approaches to biomarker evaluation and subgroup identification. A slightly modified version of this classification scheme will be used throughout this tutorial to facilitate the discussion of specific methods for subgroup discovery:

- *Global outcome modeling methods* deal with modeling the underlying outcome function $f(\mathbf{X}, T)$ (Section 6).
- *Global treatment effect modeling methods* deal with modeling the treatment contrast function $z(\mathbf{X})$ (Section 7). A very important special case of this approach is a class of methods that aim at recovering *OTRs* given a set of patient's covariates (Section 8).
- *Local modeling methods* focus on direct search for subgroups with a beneficial treatment effect, that is, identifying subgroups of patients with higher values of $z(\mathbf{X})$ (Section 9).

5. Basic biomarker evaluation methods

As indicated in Section 4.3, it is quite common to consider simplistic approaches to assessing the impact of multiple baseline covariates on the outcome variable in late-stage clinical trials. This includes subgroup discovery settings where the trial's sponsor is interested in identifying subgroups of patients with desirable characteristics based on large sets of candidate biomarkers. It will be shown in this section that basic univariate and even multivariate methods fail to address the main goal of subgroup discovery, namely, these methods focus on prognostic biomarkers and provide little or no information on predictive biomarkers that can be used to define subgroup with improved efficacy or safety. The basic approaches to biomarker evaluation and subgroup identification will be contrasted with advanced/principled approaches presented in Sections 6 (global outcome modeling methods), 7 (global treatment effect modeling methods), 8 (optimal treatment regimes), and 9 (local modeling methods).

We will first consider an approach that may be termed the *univariate regression approach*. Suppose that a set of candidate biomarkers that are believed to be predictive of treatment response has been pre-specified. This approach relies on fitting a series of regression models with the terms for treatment, single biomarker, and treatment-by-biomarker interaction. If the interaction term is significant at a pre-specified significance level (e.g., $\alpha = 0.1$), the corresponding biomarker is retained for the next step. The resulting promising biomarkers are then used to define patient subgroups. Subgroups based on binary biomarkers are constructed in a straightforward manner, and continuous/ordinal biomarkers are dichotomized before subgroups are set up. The process of dichotomization is based on appropriately defined 'optimal' cutoff values or clinically relevant cutoffs, for example, the cutoff of 60 years may be considered for the patient's age because it is commonly used in other clinical trials. An important feature of the univariate regression approach is that it supports only patient subgroups based on a single biomarker, and interactions among the biomarkers are not accounted for.

The second common approach utilizes tree-based regression models (e.g., CART methodology introduced in [33]) with the set of predictors comprised of the candidate biomarkers as well as the binary treatment indicator. Unlike the univariate regression approach, this *tree-based regression approach* incorporates information on higher-order interactions effects and can be applied to define subgroups based on multiple biomarkers (biomarker signatures). Second, cutoff values do not need to be pre-specified for continuous/ordinal biomarkers. The cutoffs are automatically estimated from the data in the process of fitting a tree-based model.

5.1. *Artificial example*

To illustrate the pitfalls of the two classes of basic approaches to subgroup identification defined previously, we will consider an artificial example. The example was based on a clinical trial dataset with the total of $n = 200$ patients. A balanced two-arm design was assumed and t_i defined the treatment indicator for the ith patient ($t_i = 0$, placebo arm; $t_i = 1$, treatment arm). There were two continuous biomarkers in the dataset (X_1 and X_2), and $\mathbf{x}_i = (x_{i1}, x_{i2})$ denoted their values for the ith patients. The two biomarkers were independent of each other and followed a uniform distribution on [0, 1]. The outcome variable was defined using the following model:

$$y_i = 2x_{i1} + 3x_{i2} + I(x_{i1} \leqslant 0.5)I(x_{i2} \leqslant 0.5)t_i + \varepsilon_i, \tag{1}$$

where $I(u)$ is the indicator function, that is, $I(u) = 1$ if u is true and 0 otherwise, and ε_i is a standard normal variable that is independent of x_{i1} and x_{i2}. It follows from this model that the true cutoff value for both biomarkers was equal to 0.5. Note that the covariates X_1 and X_2 affected the outcome variable directly as the main effects and via the treatment-by-covariate interaction. Therefore X_1 and X_2 carried both prognostic and predictive effects.

As in Section 4.1, the (hypothetical) individual treatment difference for the ith patient was defined as the difference between the expected treatment effect if the patient was assigned to the treatment arm and that if the patient was assigned to the placebo arm:

$$z(\mathbf{x}_i) = E(Y|\mathbf{X} = \mathbf{x}_i, T = 1) - E(Y|\mathbf{X} = \mathbf{x}_i, T = 0) = I(x_{i1} \leqslant 0.5)I(x_{i2} \leqslant 0.5).$$

The average treatment difference within an arbitrary subgroup S was defined in a similar way:

$$z(S) = E(Y|\mathbf{X} \in S, T = 1) - E(Y|\mathbf{X} \in S, T = 0).$$

The average treatment differences induced by the outcome model (1) within the key subgroups are shown in Table III. We can see that a positive treatment effect ($z(S) = 1$) was restricted to the following subgroup:

$$S = \{X_1 \leqslant 0.5 \text{ and } X_2 \leqslant 0.5\}.$$

This subgroup will be referred to as the *true subgroup*. In addition, Table III shows that the true subgroup naturally induced positive treatment differences in its supersets, including, for example,

$$S_1 = \{X_1 \leqslant 0.5\},\ S_2 = \{X_2 \leqslant 0.5\}.$$

Table III. Average treatment differences within the key subgroups and overall population (OP) in the simulated dataset.

Subgroup	$\{X_1 \leqslant 0.5\}$	$\{X_1 > 0.5\}$	OP
$\{X_2 \leqslant 0.5\}$	1.0	0.0	0.5
$\{X_2 > 0.5\}$	0.0	0.0	0.0
OP	0.5	0.0	0.25

Table IV. Identification of the treatment-by-biomarker interactions based on univariate regression models in the simulated dataset.

Label	Fitted regression model for $E(Y)$	Interaction effect p-value
Model A1	$a_0 + a_1X_1 + a_2T + a_3X_1T$	0.2830
Model B1	$a_0 + a_1X_1 + a_2T + I(X_1 \leqslant 0.5)T$	0.4200
Model C1	$a_0 + a_1I(X_1 \leqslant 0.5) + a_2T + a_3I(X_1 \leqslant 0.5)T$	0.8627
Model A2	$a_0 + a_1X_2 + a_2T + a_3X_2T$	0.6390
Model B2	$a_0 + a_1X_2 + a_2T + I(X_2 \leqslant 0.5)T$	0.0394
Model C2	$a_0 + a_1I(X_2 \leqslant 0.5) + a_2T + a_3I(X_2 \leqslant 0.5)T$	0.1730
Correct model	$a_0 + a_1X_1 + a_2X_2 + a_3T + a_4I(X_1 \leqslant 0.5)I(X_2 \leqslant 0.5)T$	0.00003

It is easy to verify that $z(S_1) = z(S_2) = 0.5$ and the average treatment difference in the overall population of patients (OP) was 0.25.

A dataset was simulated from the outcome model (1). The treatment difference in the overall patient population was trivial (one-sided $p = 0.3995$), and a significant beneficial effect was detected in the subgroup S (one-sided $p = 0.0164$).

5.2. *Performance of the univariate regression approach*

To characterize the performance of the univariate regression approach to subgroup identification in this simple setting, three models were fitted for each of the two biomarkers:

- Model A: model with the original biomarker included in the main term and interaction term;
- Model B: model with the original biomarker in the main term and dichotomized biomarker in the interaction term; and
- Model C: model with the dichotomized biomarker in the main term and interaction term.

The models are defined in Table IV. For simplicity, it was assumed in Models B and C that the true cutoff value (0.5) was known. As the benchmark, the correct model is also shown in Table IV. This model includes the outcome model (1) as a special case when $a_0 = 0$, $a_1 = 2$, $a_2 = 3$, $a_3 = 0$, and $a_4 = 1$.

Table IV lists the one-sided interaction effect p-value (i.e., p-value for testing the null hypothesis that the treatment-by-biomarker interaction coefficient is zero) for each of the seven regression models. It follows from the table that, with the exception of the correct model, the interaction effect tests did not detect the predictive effect of the selected biomarker, that is, did not detect a significant interaction between the biomarker and treatment indicator. The interaction effect p-value computed from Model B2 was likely to be significant by chance. Note that the p-value in the similar model based on the other biomarker (Model B1) was far from being significant.

These results presented in Table IV demonstrate that univariate regression modeling with interaction terms may perform quite poorly and fail to uncover predictive effects of candidate biomarkers. This is because, with one variable examined at a time, a subgroup defined by a multi-variable 'signature' is likely to be missed. While the individual predictive effects of the biomarkers X_1 and X_2 may appear fairly substantial in Table III based on the assumed outcome model (1), the sampling error in a dataset with only a hundred of patients per arm can easily render them non-existent, as was actually the case in this particular dataset. Further, even though the cutoff value for dichotomizing the biomarkers was assumed to be known in Models B and C, the interaction test within a regression model may not be meaningful if the main effect is misspecified (Models B2 and C2). To summarize, the key deficiency of univariate regression models is that they ignore potential synergistic effects of two or more biomarkers by failing to account for higher-order interaction effects.

5.3. *Performance of the tree-based regression approach*

Tree-based regression models are often used as a tool for discovering treatment-by-covariate interactions and identifying potentially predictive biomarkers in clinical trials. To examine the performance of the tree-based regression approach, it was applied to the simulated dataset defined earlier in this section. The tree-based approach was implemented using the *rpart* package (R package that implements recursive partitioning for classification and regression trees closely following the original CART method [33]).

We will use this example to quickly introduce regression trees as they will be used later in this tutorial, often as a building block for other more complex methods such as random forests used in the VT method

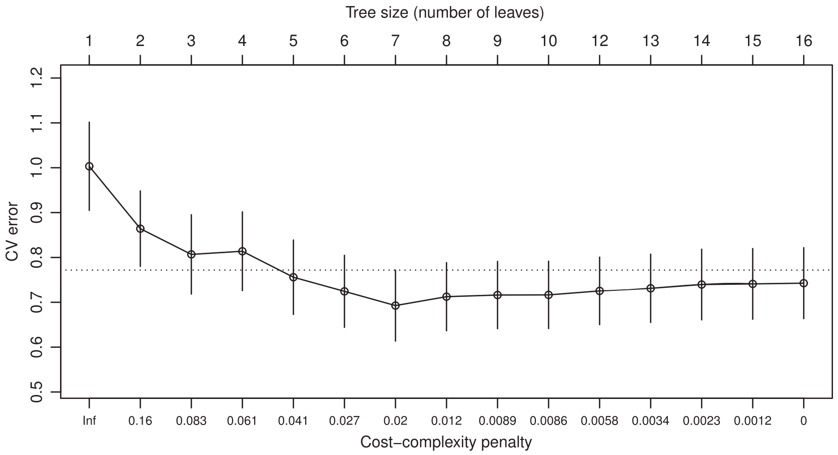

Figure 2. Cross-validation error profile for trees of different sizes in the simulated dataset.

(Section 6.4). A tree model partitions the covariate space into non-overlapping regions (leaves or 'terminal nodes') and any patient is allocated to only one region based on his or her covariates. Because a tree is organized as a decision tree, a patient is allocated to a terminal node by sequentially examining the patient's covariates starting from the first split until the patient reaches a terminal node. This is sometimes referred to as 'running a patient down the tree'. The predicted value is defined as the average outcome value within the resulting terminal node.

As a quick illustration, let us assume that a tree is fitted to a dataset with 10 covariates and consider a terminal node (patient subgroup) described by three covariates:

$$S = \{X_1 \leqslant 10 \text{ and } 1 < X_2 \leqslant 5 \text{ and } X_7 = \text{`Male'}\}.$$

Other terminal nodes may involve more or less covariates depending on the shape of the tree. Although at the first sight this may appear different from plugging the covariate vector **x** into a typical statistical model (e.g., a linear regression), fundamentally, it is the same process of evaluating a function $f(\mathbf{x})$ on a covariate vector **x**. One apparent but superficial difference is that, when a traditional statistical model is fitted to a set of covariates, all these covariates should be included in the evaluation of $f(\mathbf{x})$. In case of a tree, only those covariates involved in defining a relevant terminal node are used, for example, only three covariates were utilized to allocate patients to the terminal node defined previously. Philosophical difficulties in absorbing trees and even more complex 'black box models' by the statistical community were discussed at length in the well-known Leo Breiman's paper [34] and conceptualized as the difference between the 'data modeling culture' and 'algorithmic modeling culture'.

Returning to the simulated dataset based on the outcome model (1), an 'overgrown' tree with 10 leaves and the minimum sample size of 10 patients per node was first generated. CV was then applied to prune the tree to an optimal size. A common recipe for choosing a best-sized tree is to select the smallest (simplest) tree whose CV error is within one standard deviation of that for the tree with the lowest error (this rule is known as the 'minCV+1SE' rule). This can be accomplished via cost-complexity pruning as explained in Breiman *et al.* [33]. Figure 2 presents the CV error plot used in the pruning procedure. The lowest CV error was achieved for the tree with seven leaves, and the CV error within one standard deviation from the seven-leaf tree is represented by the dotted line. The simplest tree with the CV error below this line was the five-leaf tree, which indicates that the original tree needs to be pruned down to five leaves (the corresponding cost-complexity penalty was 0.041). The final five-leaf tree fitted to the simulated dataset is shown in Figure 3.

The splitting rules used in the final tree are shown at the top of each parent node in Figure 3. For example, the first split was made on the second biomarker with the following *child subgroups*:

- Left branch: $L_1 = \{X_2 \leqslant 0.39\}$.
- Right branch: $R_1 = \{X_2 > 0.39\}$.

After that, both child subgroups were split by the first biomarker, and so on, until five terminal subgroups were constructed. The biomarkers were selected whenever they were considered best splitters based on the reduction in the residual sums of squares (RSS) due to the split, that is, comparing the RSS 'after the

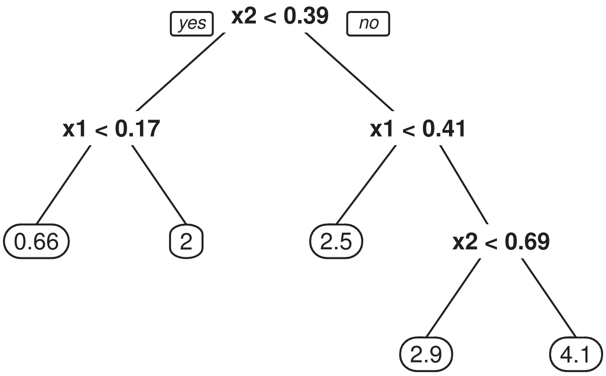

Figure 3. Pruned tree with five leaves fitted to the simulated dataset. Patients who meet the splitting condition form the left branch and those who do not form the right branch. The mean value of the outcome variable is displayed within each terminal node.

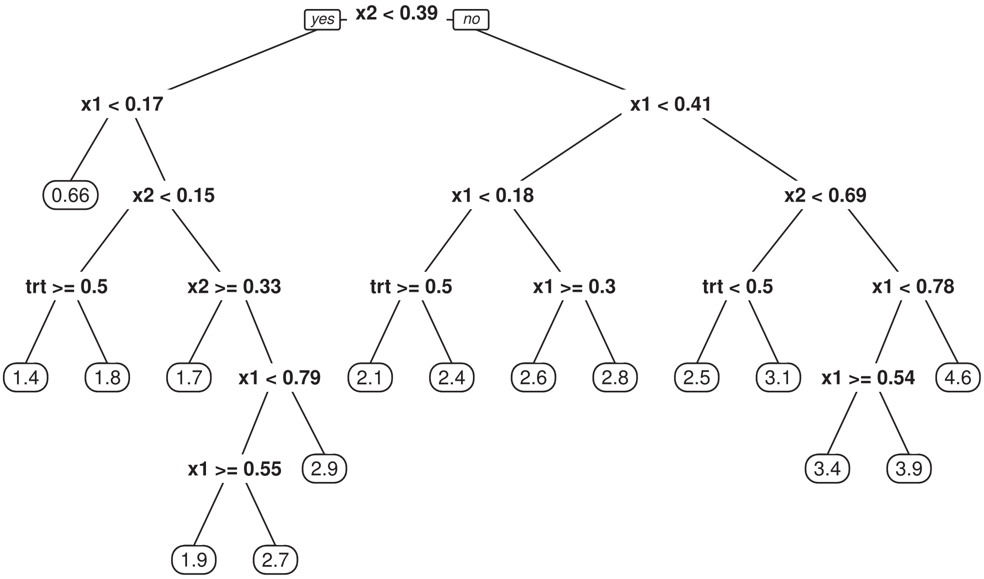

Figure 4. Example of an unpruned tree fitted to the simulated dataset.

split' versus 'before the split'. The average values of the outcome variable (Y) in each terminal subgroup displayed in Figure 3 defined a piecewise-constant fit. It is clear that larger values of the biomarkers corresponded to higher values of the outcome variable.

If the treatment indicator was selected as the splitting variable in the final tree-based model, this would help identify subgroups of patients who experience a beneficial treatment effect. However, because the overall treatment difference was rather trivial, the reduction in the sum of squares criterion based on the treatment variable alone was very modest in the beginning of the tree fitting process. As a result, the treatment variable could not compete with the strong prognostic biomarkers X_1 and X_2.

If we let the tree grow to the full size (allowing for terminal subgroups with as few as 10 patients), the treatment indicator would eventually be picked up by the tree-based regression model. The resulting tree, however, would clearly overfit the data. To give a quick example, consider the overgrown tree displayed in Figure 4. The treatment variable appeared for the first time rather late in the splitting process. For example, a subgroup defined using the treatment indicator was a subset of the following subgroup displayed in the left part of the tree: $\{X_1 > 0.17 \text{ and } X_2 \leqslant 0.15\}$. This subgroup is not even close to the true subgroup assumed in the outcome model (1), that is, $\{X_1 \leqslant 0.5 \text{ and } X_2 \leqslant 0.5\}$. It should not be surprising that this tree identified incorrect subgroups. The tree-fitting process is not aimed at recovering subgroups with a *differential treatment effect* but rather at selecting subgroups with *differential outcomes*. In other words, the tree-based regression approach supports the analysis of biomarkers with prognostic properties but provides little information on predictive biomarkers.

6. Global outcome modeling

The main approach of the biomarker evaluation and subgroup search methods in this class is to build a global model with a potentially large number of variables selected from all available candidate covariates and treatment-by-covariate interactions. The process of fitting a reliable model in this setting is quite challenging and methods of penalized and ensemble regression are commonly utilized to address this complex problem. We note that modeling of the response function can be performed either separately within each treatment arm, which results in estimating $f(\mathbf{x}, 1)$ and $f(\mathbf{x}, 0)$, or as a single model fitted to $f(\mathbf{x}, t)$, which requires explicitly modeling the treatment-by-covariate interactions. This tutorial focuses on the latter type of models. The resulting global model is applied to evaluate differential treatment effects across relevant patient subgroups and select subsets of the overall population with a beneficial effect by defining thresholds for the estimated individual treatment differences.

6.1. Overview of global outcome modeling methods

We will begin with a general overview of approaches that rely on global outcome modeling and provide a detailed description of selected methods in this class later in this section. These approaches can be grouped into parametric and non-parametric approaches. It is worth mentioning that, within the parametric framework, considerable attention was dedicated to modeling the relationship between the outcome and continuous covariates, possibly varying by treatment. A good example is an application of multivariable fractional polynomial interactions [35] to modeling treatment-by-covariate interactions in Royston and Sauerbrei [36]. While this approach may provide useful insights when dealing with a small number of candidate biomarkers, it appears less practical in complex biomarker discovery programs that require the evaluation of treatment interactions with hundreds if not thousands of candidate biomarkers.

As indicated previously, penalized regression methods are often used when modeling the outcome as a function of the prognostic effects of baseline covariates as the main effects (and possibly higher order interactions with other covariates) and their predictive effects (modeled as interactions with the treatment indicator). Because there may be a substantial number of such potential interaction effects, fitting them within a standard likelihood-based framework may not be feasible. To address this problem, various penalized methods (also known as regularization methods) that place constraints on the regression coefficients can be applied. The constraints cause the interaction effects to shrink towards zero. In particular, with the lasso penalty [37], some of the effects shrink exactly to zero, which leads to biomarker and subgroup selection methods. For example, Imai and Ratkovic [38] developed the FindIt method, which utilizes a support vector machine (SVM) classifier with separate lasso-type constrains over the predictive and prognostic effects included in the model. This approach accounts for the fact that the predictive effects are inherently weaker and need to be treated differently from the prognostic effects. We will provide more information on penalized regression approaches in Section 6.2. The penalized regression methods will be applied to Case study 1 with a binary outcome variable in Section 10.2.

Several subgroup search procedures in this class make use of the concept of potential outcomes introduced in Section 4.1. Cai *et al.* [39] developed a two-stage method for assessing the treatment effect heterogeneity at the patient level. At the first stage, a proportional hazards Cox regression model with preselected covariates is fitted to the data to estimate the treatment difference at the patient level. After that a non-parametric smoothing method is applied to construct simultaneous confidence bands for the treatment differences estimated in the first stage. Patients with a large estimated treatment difference can be included in a target subgroup. Zhao *et al.* [40] proposed a systematic method for building and evaluating different classifiers for defining patient subgroups based on thresholding estimated treatment difference scores at various cutoffs (see also [41] and [42] for related ideas).

Another important example of utilizing potential outcomes to construct subgroup identification procedures is the non-parametric VT method [43]. The underlying regression function $f(\mathbf{x}, t)$ is estimated in the first stage using random forests [44] (other methods of ensemble regression can also be applied). Once a random forest has been fitted to the data, two predicted outcomes are obtained for the ith patient, assuming that the patient was allocated to the treatment or control arms, respectively, that is,

$$\widehat{f}(\mathbf{x}_i, 0) \text{ and } \widehat{f}(\mathbf{x}_i, 1),\ i = 1, \dots, n.$$

The hypothetical treatment difference z_i is then estimated for each patient as follows:

$$\widehat{z}(\mathbf{x}_i) = \widehat{f}(\mathbf{x}_i, 1) - \widehat{f}(\mathbf{x}_i, 0).$$

The obtained treatment differences are used in the second stage of the VT method as the outcome variables for a simple regression tree with the goal of identifying a subgroup, where $z(\mathbf{x}_i)$ is expected to be larger than a clinically meaningful threshold denoted by δ. The final subgroup is formed as the union of all terminal nodes of the tree where the predicted treatment differences are greater than δ. We will provide a more detailed account of this method in Section 6.4 and apply it to Case study 1 in Section 10.3.

Hybrid strategies for subgroup identification that combine parametric and non-parametric (e.g., tree based) models were also proposed in the literature. For example, Dusseldorp *et al.* [45] proposed a method called simultaneous threshold interaction modeling algorithm (STIMA), which represents a combination of a linear multiple regression model for modeling the main effects and a tree for modeling the higher-order interaction effects. The first step of the algorithm involves fitting a linear model for the main effects with the treatment arms combined. After that the experimental treatment and control arms are analyzed separately, and the algorithm proceeds by splitting on the remaining covariates. The goal is to identify the best split by examining all candidate splits within each terminal node and choosing the split that maximizes the increase in the variance accounted for by the current model. Each new split is captured by a new interaction term, which is added to the overall regression model, and all regression coefficients are re-estimated. This procedure is repeated in a sequential manner until a pre-defined stopping condition is met, which results in generating a sequence of regression models of increasing complexity. An optimal model and corresponding tree structure can be selected using CV.

Bayesian methods for subgroup identification within the global outcome modeling framework typically focus on applying empirical Bayes or fully Bayesian methods to complex regression models with terms involving biomarker-by-treatment interactions (e.g., [46–48]).

It is important to note that frequentist methods for penalized regression discussed earlier in this section have a natural Bayesian interpretation. The penalty function corresponds to specific prior distributions placed on the model coefficients; for example, the ridge penalty corresponds to normal priors centered at zero, and the lasso penalty corresponds to Laplace priors centered at zero. The penalty factor is inversely related to the variance of the prior distribution and a larger penalty indicates a stronger belief that a particular model parameter is equal to zero. Other proposals for priors include Student's *t*-distribution, which is an intermediate case between the Gaussian and Laplace distributions [49].

'Fully Bayesian' penalized regression methods [50] were also adapted to the context of biomarker evaluation. These methods place a hyper-prior on the penalty parameter in the spirit of hierarchical Bayesian models. It is instructive to contrast the fully Bayesian approach with the frequentist penalized regression discussed in Section 6.2. While the frequentist or empirical Bayes approaches focus on selecting a *single* value of the penalty parameter by CV or other criteria, the fully Bayesian approach emphasizes the process of *averaging* multiple penalized regressions with different penalties sampled from their posterior distribution. Gu *et al.* [51] proposed a two-step biomarker selection strategy in the context of time-to-event outcomes based on the Bayesian lasso regression. At the first step, the procedure utilizes the grouped lasso penalty to select potentially relevant biomarkers (grouped with their treatment interactions), and at the second step, a Bayesian adaptive lasso is applied to refine the variable selection among biomarkers identified in the first step. In addition, non-parametric Bayesian regression methods were considered in the literature. For example, Xu *et al.* [8] proposed a Bayesian non-parametric procedure known as the SUBA procedure, which utilizes a *random partition model* generated via random splits on candidate biomarkers and is similar in spirit to Bayesian regression trees, for example, Bayesian CART developed by Chipman *et al.* [52].

It is worth noting that biomarkers may be used to define patient subgroups indirectly, for example, by affecting the probability of subgroup membership, rather than by directly defining covariate-based scores or signatures and associated thresholds. For example, Shen and He [53] proposed a latent logistic-normal mixture where the outcome is modeled via a linear model incorporating an interaction between the treatment and a latent subgroup variable. The subgroup membership probabilities are modeled using a logistic regression model with pre-selected biomarkers as covariates. The analysis includes testing the hypothesis of no subgroup effect and evaluating pre-selected biomarkers (if the former hypothesis is rejected) within a single analytic procedure. As a limitation, such models can not handle selection from a large set of candidate biomarkers.

6.2. *Penalized regression for global outcome modeling*

We begin with a brief description of the general framework of penalized regression for estimating the expected outcome function $f(\mathbf{x})$, that is, the expected patient's response given the covariate vector $\mathbf{x}$. Here, to simplify notation, we assume that the covariate vector includes, in addition to candidate covariates, the treatment indicator as well as all other relevant variables (or 'data features') that may have been created by applying appropriate transformations to the input variables, for example, interaction effects. The motivation behind penalized regression came from different areas of statistical research and evolves around the two goals:

- constructing stable and accurate predictive models in problems with a large number of (often highly correlated) covariates; and
- selecting variables (simultaneously with parameter estimation) to define a parsimonious and interpretable regression model.

Although the objectives of 'prediction' and 'interpretation' may be considered somewhat conflicting, they can be tackled under a unifying framework of penalized regression, which became a major building block for a multitude of related methods.

Within a class of parametric regression models, the expected response for the ith patient, $i = 1, \dots, n$, is a function of the vector of pre-specified covariates $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ with the vector of model parameters/regression coefficients denoted by $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)$. The general goal of penalized regression can be stated as estimation of the model parameters by minimizing an objective function. The objective function is defined as a sum of the *loss function*, which measures the discrepancy between the observed outcome and expected response, and *penalty function*, which depends on the absolute values of the model parameters. In other words, we seek $\widehat{\boldsymbol{\beta}}$ such that

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i|\boldsymbol{\beta})) + J_\lambda(\boldsymbol{\beta}) \right),$$

where $L(Y, f(\mathbf{X}))$ is the loss function, $J_\lambda(\boldsymbol{\beta})$ is the penalty function, and λ is the penalty parameter (or, in general, a vector of parameters).

While the loss function rewards regression models that closely approximate the observed response (leading to a smaller bias), the penalty term rewards parsimonious models with fewer and smaller coefficients (leading to a smaller prediction variance). The penalty parameter λ is a tuning parameter that controls the relative importance of the two conflicting goals. With $\lambda = 0$, the penalty term vanishes, and when $\lambda = \infty$, the coefficients are all shrunk to 0. Therefore, penalty parameters support a balance between the model fit and model complexity via a *bias-variance trade-off*.

Different types of penalized regression can be constructed by specifying the three major 'building blocks':

- Specifying the form of the loss function $L(Y, f(\mathbf{X}))$, which naturally depends on the outcome type (continuous, binary, time-to-event, etc). The squared-error loss function, that is, $L(Y, f(\mathbf{X})) = (Y - f(\mathbf{X}))^2$, is probably the most common choice. This function is used in least-squares estimation for continuous outcomes.
- Specifying the penalty function $J_\lambda(\boldsymbol{\beta})$. For example, the ridge penalty [54] places a penalty on the sum of squares of the regression coefficients, $J_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \beta_j^2$.
- Specifying a class of parametric functions $f(\mathbf{X}|\boldsymbol{\beta})$ to approximate the unknown expected response $f(\mathbf{X})$. An example is a linear regression, which is defined by $f(\mathbf{X}|\boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$.

For the rest of our discussion of penalized regression, we will assume that the outcome variable is binary. We will adopt a coding convention commonly used in statistical/machine learning literature, namely, the binary outcomes will be coded as -1 (negative outcome) and $+1$ (positive/desirable outcome), rather than 0 and 1, unless stated otherwise.

Hastie *et al.* [55] discuss a variety of loss functions for regression models with binary outcomes. The following two loss functions will be considered in this section:

- binomial deviance loss: $L(Y, f(\mathbf{X})) = \log(1 + \exp(-Yf(\mathbf{X}))$; and
- hinge loss: $L(Y, f(\mathbf{X})) = [1 - Yf(\mathbf{X})]_+$, where $[a]_+ = \max(0, a)$.

The first loss function is easily recognized as the familiar binomial (negative) likelihood. This loss function is used in penalized logistic regression models. The function that minimizes this loss function over the entire population, that is, the expectation of the loss is computed with respect to y, is termed a *population minimizer*. This function is given by the logit of the response probability, that is,

$$\log \frac{P(Y = +1|\mathbf{X})}{P(Y = -1|\mathbf{X})}.$$

The hinge loss is less familiar to the statistical community. This loss function is used in SVMs, which is one of the popular methods used in classification [56]. Here, the decision boundary is estimated as all such points $\mathbf{x}$ that $\hat{f}(\mathbf{x}) = 0$ and a SVM classifies a point to the classes $\{Y = -1\}$ and $\{Y = +1\}$ depending on which side of the boundary it is. The hinge loss combined with the ridge penalty leads to boundaries that maximize the width of the margin between the two classes $\{Y = -1\}$ and $\{Y = +1\}$ defined by its two 'soft' boundaries:

$$\left\{\mathbf{x} : \hat{f}(\mathbf{x}) = -1\right\} \text{ and } \left\{\mathbf{x} : \hat{f}(\mathbf{x}) = +1\right\}.$$

Intuitively, a wider margin between the two classes achieved on the training data not only means better discrimination between the patients with positive and negative outcomes but also suggests that the classification rule should work well on the new (test) data.

Popular penalties used in general penalized regression models include

- ridge or l_2 penalty: $J_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \beta_j^2$, $\lambda > 0$;
- lasso or l_1 penalty (Tibshirani [37]): $J_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} |\beta_j|$, $\lambda > 0$; and
- elastic net penalty (compromise between the l_1 and l_2 penalties [57]):

$$J_\lambda(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2,$$

where $\lambda_1 = \lambda\alpha$, $\lambda_2 = \lambda(1-\alpha)/2$, $0 \leqslant \alpha \leqslant 1$ and $\lambda > 0$.

Different penalties would result in different features of the estimated model. The ridge penalty causes the regression coefficients and correlations among the covariates to shrink towards 0 by a scale factor of $1/(1+\lambda)$. This leads to 'decorrelating' the data, and as $\lambda \to \infty$, the regression coefficients begin to behave like univariate Ordinary Least Squares (OLS) coefficients scaled down by a factor of $1/(1+\lambda)$. Generally, the ridge penalty has the effect of averaging across correlated variables.

The lasso penalty also shrinks the regression coefficients towards zero, but unlike the ridge penalty, it shrinks some of the coefficients exactly to zero. This means that parameter estimation is performed simultaneously with automatic variable selection. The lasso penalty is called a 'sparse penalty' as it enforces sparsity in the estimated coefficients. It is often argued that this 'bet on sparsity' is a reasonable strategy, especially when the covariate space is very large compared with the sample size ($p > n$), because it works well when the 'truth' is sparse (i.e., most coefficients are indeed equal to zero), and if it is not, there is no method to accurately estimate all of the coefficients anyway. The lasso penalty can be thought of as a less greedy forward variable selection method, which is closely related to least angle regression introduced in [58]. One deficiency of the lasso is that, given a set of highly correlated covariates, it arbitrarily selects only one of them. The *elastic net* penalty was developed as an attempt to remove this undesirable feature. This penalty function combines the decorrelation property of the ridge penalty with the variable selection property of the lasso penalty. Several other important variations and improvements on the lasso method that can also be applied more generally to the elastic net have been developed recently, including the *fused* lasso [59], *adaptive* lasso [60], and *grouped* lasso [61].

A natural approach to selecting the penalty parameter(s) is to try different values within a plausible range and evaluate the performance of the resulting model using an independent dataset or via CV. The parameter corresponding to the smallest CV error is chosen, or alternatively, the 'minCV+1SE' rule (Section 5.3) can be applied. Fortunately, for many penalized regression methods, for example, for the lasso [58] and some versions of SVM [62], it was shown that the estimated coefficient or regularization paths, $\beta_j(\lambda), j = 1, \ldots, p$, are piecewise linear. Efficient procedures were developed to compute the entire regularization path $\beta_j(\lambda)$ with the computational effort of a single least-squares fit [63]. Recently developed algorithms based on cyclic coordinate descent allowed computing entire regularization paths for more general classes of loss functions and penalties, covering ridge regression, elastic net penalties combined with many types of different loss functions encountered in generalized linear models [64]. These algorithms are used in the R package *glmnet*.

Once the regularization path has been computed, the optimal value of λ for 'variable selection' penalties such as the lasso and elastic net can then be found by examining only a few 'critical' or 'transition' values of λ along the path when some of the coefficients approach zero, which means that the associated covariates are unselected.

Finally, we can choose among different classes of regression functions to approximate the expected response $f(\mathbf{x})$. The idea is that the selected class of functions should be rich enough to contain the true function and allow for a variety of fits from very simple fits to very complex fits. One should not be afraid to include very complex functions because the penalty term, which plays the role of a 'constraint jacket', is likely to reduce the model complexity. Such 'expansion' of the covariate space (also known as the feature space), which is subsequently shrunk to the right size, is one of the key conceptual elements of statistical and machine learning (Sections 5.3 and 6.4). This approach reflects our lack of knowledge of the precise functional form of the true response function.

As a popular approach, functions of interest may be defined as linear expansions of basis functions [55] denoted by $h_1(\mathbf{X}), \dots, h_m(\mathbf{X})$, for example,

$$f(\mathbf{X}|\boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{m} \beta_j h_j(\mathbf{X}).$$

The basis functions could be specified explicitly, fr example, as the spline basis functions for each covariate, or implicitly by defining the so-called kernel functions [65]. Note that the basis functions could simply represent the original p covariates, that is, $h_j(\mathbf{X}) = X_j, j = 1, \dots, p$.

Returning to the problem of subgroup identification, note that the response function $f(\mathbf{X})$ also depends on the treatment assignment T, that is, $f(\mathbf{X}) = f(\mathbf{X}, T)$. When selecting the functional form of $f(\mathbf{X}, T)$, it is natural to focus on the feature space that includes all of the candidate covariates, their two-way interactions as well as all two-way and three-way interactions of covariates with the treatment variable. The main effects of the covariates and their interactions can be modeled via different functional forms by applying appropriate transformations, including power functions, piecewise polynomials, and splines.

To illustrate the general penalized regression methodology presented previously, we fitted logistic regression models with the binomial deviance loss function and lasso penalty to a slightly modified version of the simulated dataset from Section 5.1, which was generated based on the outcome model (1). The original continuous outcome was converted into a binary outcome by letting $Y = +1$ if $Y^* > c$ and $Y = -1$ otherwise (Y^* is the original outcome, $+1$ denotes the desirable outcome, and c is a constant). To make this exercise more challenging, the original set of covariates, that is, X_1 and X_2, was extended by including 10 noise variables $X_3, \dots, X_{12}$ generated independently of each other from the standard normal distribution, that is, $N(0, 1)$. Note that the analyst does not know that these covariates are irrelevant. The resulting true model for the binary outcome was given by

$$\begin{aligned} y_i^* &= 2x_{1i} + 3x_{2i} + I(x_{1i} \leqslant 0.5)I(x_{2i} \leqslant 0.5)t_i + \varepsilon_i, \\ y_i &= 2I\left(y_i^* > c\right) - 1, \end{aligned} \tag{2}$$

where x_{1i} and x_{2i} were independently simulated as $U(0, 1)$ and ε_i followed $N(0, 1)$, $i = 1, \dots, n$. The constant c was set to the expected value of the original outcome variable, that is, $c = E(Y^*) = 2.75$.

The observed treatment differences in the probability of a desirable outcome ($Y = 1$) in the simulated dataset within selected subgroups are shown in Table V. As intended, the largest positive treatment effect is concentrated in the left upper cell corresponding to the true subgroup, that is, $S = \{X_1 \leqslant 0.5$ and $X_2 \leqslant 0.5\}$.

The feature space included the following variables:

- treatment indicator, T;
- all candidate covariates $X_1, \dots, X_{12}$;
- two-way covariate interactions, including the squared covariates, that is, X_iX_j, X_i^2, $i = 1, \dots, 12$, $j = 1, \dots, 12$; and

Table V. Average treatment differences within the key subgroups and overall population (OP) in the simulated dataset (binary outcome).

Subgroup	$\{X_1 \leqslant 0.5\}$	$\{X_1 > 0.5\}$	OP
$\{X_2 \leqslant 0.5\}$	0.315	0.019	0.161
$\{X_2 > 0.5\}$	−0.428	0.021	−0.162
OP	−0.074	0.017	−0.01

- interactions of the covariates and two-way covariate interactions with the treatment indicator, i.e., $X_iT, X_iX_jT, X_i^2T, i = 1, \ldots, 12, j = 1, \ldots, 12$.

The expanded covariate space consisted of a total of 181 variables, which was comparable with the total sample size of $n = 200$ patients. Note that all covariates, including the treatment indicator, were first standardized to have zero means and unit standard deviations. The interaction terms were computed based on the standardized input variables, and the lasso procedure from the *glmnet* package was run with no internal standardization (i.e., `standardize=FALSE`). This method of standardization is similar to the approach suggested in [49].

Figure 5 shows the coefficient paths as a function of the log-transformed penalty parameter λ. The paths for the influential variables in the model, that is, the main effects (X_1 and X_2) and predictive effects (X_1X_2T, X_1T and X_2T), are shown as black curves, whereas the other paths are shown as gray curves in Figure 5. The procedure used a fast method of cyclical coordinate descent for fitting the entire lasso (and, more generally, elastic net) coefficient path for logistic regression. It follows from Figure 5 that the paths were indeed piecewise linear. As λ increased, the coefficients for most irrelevant covariates and their interactions were quickly shrunk down to zero. The figure suggests that a key component of penalized regression is the selection of an optimal value of the penalty parameter. The two horizontal lines indicate reasonable values of λ that were selected via CV, as explained in the succeeding discussion.

Figure 6 presents a cross-validated negative log-likelihood with associated standard errors as a function of the log-transformed penalty parameter λ. The cross-validated negative log-likelihood was computed as follows. First, the dataset was randomly divided into $k = 10$ sets (folds). For all patients in the lth

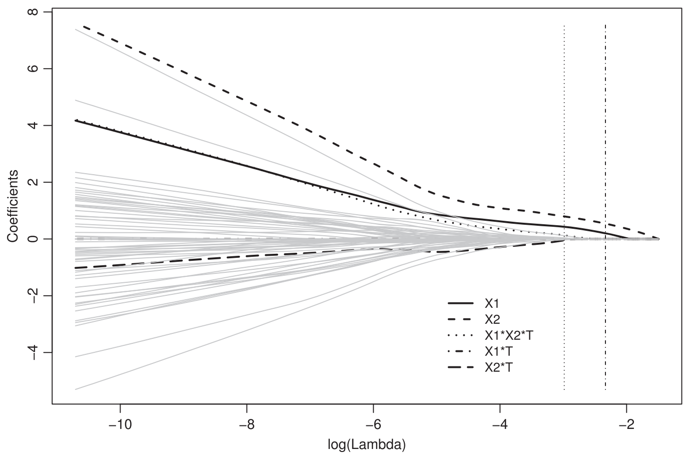

Figure 5. Lasso coefficient paths in the simulated dataset with 181 covariates. The vertical lines are drawn at $\lambda_{\min}$ (left line) and $\lambda_{\min 1se}$ (right line).

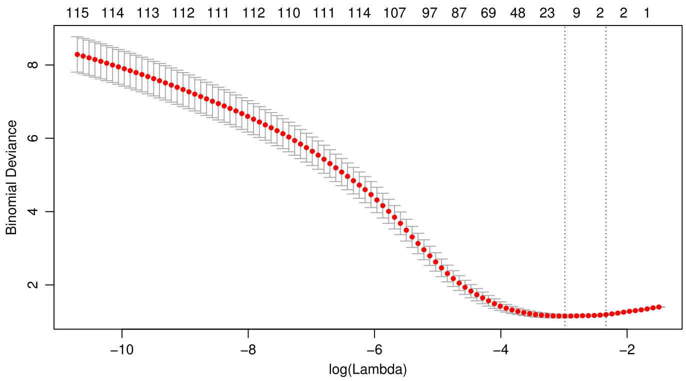

Figure 6. Ten-fold cross-validated negative log likelihood (binomial deviance) as a function of the log-transformed penalty parameter λ. The two vertical lines correspond to $\lambda_{\min}$ (left line) and $\lambda_{\min 1se}$ (right line). The error bars are based on the standard errors. The values shown in the upper horizontal axis are the numbers of non-zero coefficients resulting when the penalty ($\log(\lambda)$) is set at values shown in the lower horizontal axis.

fold (i.e., $i \in \mathcal{F}_l$), CV estimates were found from the penalized regression fit to the data with the lth fold removed, and the corresponding coefficient paths, denoted by $\widehat{\beta}_{j(-l)}(\lambda)$, were estimated. The cross-validated probability of the desirable outcome for a patient within the lth fold was computed using the inverse logit as follows:

$$\widehat{p}_i^{CV}(\lambda) = \frac{1}{1 + \exp\left[-1\left(\widehat{\beta}_{0(-l)} + \sum_{j=1}^{p} \widehat{\beta}_{j(-l)}(\lambda) x_{ij}\right)\right]},\ i \in \mathcal{F}_l.$$

The contribution of the ith patient to the CV-based negative log-likelihood function was given by

$$\mathcal{L}_i^{CV}(\lambda) = -\frac{1}{2}\left[(1+y_i)\widehat{p}_i^{CV}(\lambda) + (1-y_i)\left(1-\widehat{p}_i^{CV}(\lambda)\right)\right].$$

This contribution reflects the probability of the desirable outcome predicted from the regression model with a particular value of the penalty parameter, which was constructed by removing the fold to which the ith patient was assigned. The overall cross-validated profile was defined as a function of the penalty parameter λ by averaging the patient-specific contributions, that is,

$$\mathcal{L}^{CV}(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i^{CV}(\lambda).$$

Optimal values of λ were determined using the following two criteria based on the CV likelihood:

- $\lambda_{\min}$ was found by minimizing the overall cross-validated negative log-likelihood profile.
- $\lambda_{\min 1se}$ was selected using the 'minCV +1SE rule' rule. Specifically, $\lambda_{\min 1se}$ was defined as the largest λ such that $\mathcal{L}^{CV}(\lambda)$ is equal to $\mathcal{L}^{CV}(\lambda_{\min}) + SE$, where SE denotes the associated standard error, which was computed simply as the standard deviation of the average cross-validated negative likelihood across the k folds. This approach corresponds to the notion of choosing the *simplest* model within the 'chance variability' from the model with the smallest negative log-likelihood. Note that a larger penalty, if anything, would result in setting more coefficients to zero.

It is instructive to examine the non-zero parameters in the regression model corresponding to the two optimal values of the penalty parameter. The terms with non-zero coefficients are listed in Table VI.

It is interesting that only two terms (X_1 and X_2) had non-zero coefficients when the penalty parameter was set to $\lambda_{\min 1se}$, which means that none of the predictive variables survived such a harsh penalty.

Table VI. Penalized logistic regression models with the lasso penalty based on $\lambda_{\min}$ and $\lambda_{\min 1se}$ in the simulated data example.

	Estimated regression coefficient		
Selected model term	Based on $\lambda = \lambda_{\min}$	Based on $\lambda = \lambda_{\min 1se}$	Effect type
X_2	0.795	0.541	Prognostic (true)
X_1	0.430	0.199	Prognostic (true)
X_1X_2T	0.127	0	Predictive (true)
X_1X_2	0.117	0	Prognostic (true)
X_5X_6T	0.098	0	Predictive (noise)
X_2T	−0.046	0	Predictive (true)
X_1X_6	0.038	0	Prognostic (partially noise)
X_4^2	0.037	0	Prognostic (noise)
X_9T	0.033	0	Predictive (noise)
X_7^2	−0.030	0	Prognostic (noise)
X_9^2T	−0.026	0	Predictive (noise)
$X_4X_{11}T$	0.016	0	Predictive (noise)
X_3X_8T	0.014	0	Predictive (noise)
X_2X_6	−0.006	0	Prognostic (partially noise)
X_{10}^2T	−0.00005	0	Predictive (noise)

Considering now the regression model corresponding to less severe penalty $\lambda_{\min}$, the model terms with non-zero coefficients included the relevant predictive effects X_1X_2T and X_2T as well as the covariate and treatment-covariate interactions that involved the irrelevant (noise) effects. In general, the coefficients for the terms involving the noise covariates were smaller in magnitude than those for the terms included in the correct model, with the exception of the coefficient for the noise effect X_5X_6T that was comparable in magnitude with that for the true predictive effects. Note that the true predictive effect in the model is defined using the indicator function $I(X_1 \leqslant 0.5)I(X_2 \leqslant 0.5)$, which is quite challenging to recover within this model's feature space. As we see from Table VI, it is approximated (rather crudely) by a linear combination of the true predictive effects and a few noise effects. Because the covariates were first standardized, the predictive combination (or treatment difference score) was essentially based on

$$\left(0.127\,\frac{X_1-0.50}{0.29}\,\frac{X_2-0.49}{0.29}-0.046\,\frac{X_2-0.49}{0.29}\right)\frac{T-0.5}{0.50}+\text{noise effects}.$$

To evaluate the performance of the lasso approximation, the treatment effects can be estimated within the true subgroup $S = \{X_1 \leqslant 0.5 \text{ and } X_2 \leqslant 0.5\}$. The individual treatment contrasts can be easily estimated from a selected penalized regression model using the *predict* function in the *glmnet* package. First, the penalty parameter is set to $\lambda_{\min}$, and the predicted values are computed on a probability scale for a 'new' dataset with the same covariates $X_1, \dots, X_{12}$ as in the original dataset, the treatment indicator variable set to $(1-0.5)/0.5 = 1$ for all patients and the treatment-by-covariate interactions updated accordingly. This leads to the estimated probability of $y_i = 1$ if the ith patient is assigned to the treatment arm. Similarly, by setting the treatment indicator variable to $(0-0.5)/0.5 = -1$, the probability $y_i = 1$ can be estimated in the case when the ith patient is assigned to the control arm. The patient-level treatment contrast $z(\mathbf{x})$ is then computed as the difference between the predicted probabilities for the two potential outcomes:

$$\hat{z}(\mathbf{x}) = \frac{\exp\left[\hat{f}\left(\mathbf{x},1|\hat{\boldsymbol{\beta}}(\lambda)\right)\right]}{1+\exp\left[\hat{f}\left(\mathbf{x},1|\hat{\boldsymbol{\beta}}(\lambda)\right)\right]} - \frac{\exp\left[\hat{f}\left(\mathbf{x},0|\hat{\boldsymbol{\beta}}(\lambda)\right)\right]}{1+\exp\left[\hat{f}\left(\mathbf{x},0|\hat{\boldsymbol{\beta}}(\lambda)\right)\right]},$$

where $\hat{f}(\mathbf{x},t|\hat{\boldsymbol{\beta}}(\lambda))$ is the logit of the probability of $Y = 1$, given the covariate vector $\mathbf{X} = \mathbf{x}$ and treatment $T = t$, estimated using penalized logistic regression. Finally, the estimated values of $z(\mathbf{x})$ are averaged for the patients within the true subgroup, which results in an estimate of the expected treatment effect in the subgroup, that is, $E(z(\mathbf{X})|\mathbf{X} \in S)$. This estimate is equal to 0.0325, which, although positive, is far below the observed value of 0.315 shown in Table V.

In general, there is a great deal of uncertainty around the selection of the 'correct' value of the penalty parameter λ. However, even if a suboptimal value of λ is chosen, for example, $\lambda_{\min}$ is selected in this particular example, the terms involving irrelevant covariates will cancel out when the individual treatment contrasts are computed as long as they do not involve interactions with the treatment indicator (which is unfortunately not the case in the example discussed previously). Note that this desirable effect does not happen when we evaluate treatment contrasts on the probability scale, which is a nonlinear transformation of the estimated $f(\mathbf{X}, T)$. Therefore, operating on the logit scale adds extra protection against bias when evaluating treatment contrasts.

6.3. FindIt method

It was shown in Section 6.2 that general penalized regression methods can be used for the purpose of evaluating predictive biomarkers and patient subgroups with a differential treatment effect. This section presents a recently proposed variation on the penalized regression methodology, known as the FindIt method, which was specifically adopted for personalized medicine settings [38]. An explicit notation for the outcome function that separates the covariates and treatment indicator, that is, $f(\mathbf{X}, T)$, will be used in this section.

The FindIt method extends the penalized regression framework by selecting the following components of a general penalized regression method:

- The squared hinge loss, termed the squared loss support vector machine (L2-SVM) loss, was selected (note that L2 refers to the loss function rather than the penalty). Imai and Ratkovic [38] argued for L2-SVM 'because it returns the standard difference-in-means estimate for the treatment effect in the absence of pre-treatment covariates' similarly to the ordinary least squares regression. The 'population minimizer' of the hinge loss, that is, the function $f(\mathbf{X}, T)$, which minimizes the expected loss with the expectation taken over $Y|\mathbf{X}, T$, is given by

$$\operatorname{sign}(2P(Y = +1|\mathbf{X}, T) - 1),$$

which merely returns the predicted class label for each observation. On the other hand, the population minimizer for the squared hinge loss is $f(\mathbf{X}) = 2P(Y = +1|\mathbf{X}, T) - 1$ (table 12.1 in [55]), which, as we will see, enables computing the treatment contrast on the probability scale using the SVM estimator of $f(\mathbf{X}, T)$.

- Penalty function is the lasso penalty (l_1 penalty), but two separate penalty parameters are introduced for the covariates and their interactions (combined in the matrix $\mathbf{U}$) that represent the prognostic effects, the treatment–covariate interactions (combined in the matrix $\mathbf{V}$) that represent the predictive effects and the treatment indicator.
- The expected response $f(\mathbf{X}, T|\boldsymbol{\beta})$ is modeled as a linear function with the covariate space defined by the 'prognostic effects' ($\mathbf{U}$) and 'predictive effects' plus the treatment indicator ($\mathbf{V}$). The parameter space is $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}^{(u)}, \boldsymbol{\beta}^{(v)})$, where the length of the vectors $\boldsymbol{\beta}^{(u)}$ and $\boldsymbol{\beta}^{(v)}$ is p_u and p_v, respectively.

The proposed penalty function is given by

$$J_\lambda(\boldsymbol{\beta}) = \lambda_u \sum_{j=1}^{p_u} \left|\beta_j^{(u)}\right| + \lambda_v \sum_{j=1}^{p_v} \left|\beta_j^{(v)}\right|.$$

The two penalty parameters λ_u and λ_v are needed to ensure that variable selection is performed separately for the prognostic ($\mathbf{U}$) and predictive ($\mathbf{V}$) effects. Different penalties can be justified by the fact that prognostic factors tend to have much stronger effects than predictive factors and therefore may dominate the fit if a single penalty is used.

The selection of optimal values for the penalty parameters is based on the generalized cross-validation (GCV) criterion (Wahba [65]), which is justified as an approximation to the leave-one-out CV for linear models under the squared-error loss function. The GCV is evaluated as a function of (λ_u, λ_v), and the parameter values associated with the smallest GCV are selected.

As we already mentioned, one deficiency of SVM with the hinge loss is that, unlike logistic regression, it does not return an estimated probability of the desirable outcome ($Y = +1$). This could be viewed as a very significant limitation for this particular application because one is specifically interested in computing the individual treatment contrasts $z(\mathbf{X})$. Because the population minimizer of the squared hinge loss is $2P(Y = +1|\mathbf{X}, T) - 1$, the probabilities can be recovered from the fitted values by a simple transformation after truncating the values outside the $[-1, 1]$ interval. This leads to the proposed solution for computing the treatment contrasts for a given vector of covariates, termed the *conditional average treatment effect*. Specifically, the predicted potential outcomes are first truncated at $+1$ and -1, and then the individual treatment contrasts are computed as

$$\widehat{z}_i = \frac{1}{2}\left[\widehat{f}_{tr}(\mathbf{x}_i, 1) - \widehat{f}_{tr}(\mathbf{x}_i, 0)\right],$$

where $\widehat{f}_{tr}$ denotes the predicted value of the outcome function within $[-1, 1]$.

To illustrate the FindIt method, it was applied to the simulated dataset used in Section 5.1 with the continuous outcome converted to a binary outcome as in (2). The calculations were performed using the R package *FindIt*.

Recall from Section 6.2 that 90 variables in the feature space were prognostic and the remaining 91 variables included the treatment indicator and thus exhibited predictive properties, which means that $p_u = 90$ and $p_v = 91$. The optimal values of the penalty parameters for the prognostic and predictive variables (λ_u and λ_v) were determined using the GCV criterion. The resulting logistic regression model included only 10 terms with non-zero coefficients. The estimated non-zero coefficients are listed in Table VII in the decreasing order of magnitude. Remarkably, the largest coefficient was associated with the true interaction effect X_1X_2T. As in the logistic regression model with a single lasso penalty (Table VI), the noise interaction effect X_5X_6T was also selected. A potential advantage of using two separate penalties is that the coefficients associated with the true predictive effects and average treatment effects in patient subgroups can be estimated more accurately. This feature would make the FindIt method more attractive for selecting predictive covariates compared with standard penalized regression approaches. Unfortunately,

Table VII. Penalized logistic regression model based on the FindIt method in the simulated data example with λ_u and λ_v selected by the generalized cross-validation criterion.

Selected model term	Estimated regression coefficient	Effect type
X_1X_2T	1.120	Predictive (true)
X_2	0.580	Prognostic (true)
X_1	0.308	Prognostic (true)
X_2X_6T	−0.084	Predictive (partially noise)
X_5X_6T	0.052	Predictive (noise)
$X_{10}X_{12}T$	0.037	Predictive (noise)
X_9^2T	−0.009	Predictive (noise)
T	−0.009	Treatment indicator
X_7^2T	−0.008	Predictive (noise)
X_4^2	0.008	Prognostic (noise)

these potential advantages of *FindIt* were not seen in this particular simulation example. The predicted effect in the true subgroup $S = \{X_1 \leqslant 0.5 \text{ and } X_2 \leqslant 0.5\}$ based on the estimates from Table VII were far below the observed values shown in Table V.

A natural question to ask at this point is, how should one define subgroups of patients who are likely to experience a beneficial treatment effect based on penalized regression methods? One possible solution is to plot the estimated treatment contrasts $\widehat{z}_i$ against the covariates with non-zero coefficients to identify reasonable cutoff values that translate into clinically interpretable subgroups. The resulting subgroups will be easier to communicate to the patient matter experts than the results of variable selection. This approach will be illustrated in the next section.

6.4. Virtual twins

This section provides a detailed description of the VT method [43]. This method combines, as its building blocks, basic elements that may be utilized in other subgroup identification approaches.

As was explained earlier in this section, VT is a two-stage procedure. The first stage involves estimating the underlying regression function $f(\mathbf{X}, T)$ and the individual treatment contrast for each patient, that is, $\widehat{z}_i = \widehat{f}(\mathbf{x}_i, 1) - \widehat{f}(\mathbf{x}_i, 0)$. At the second stage, subgroups are identified by fitting a regression tree to $\widehat{z}_i$ or a classification tree to the binary outcomes based on dichotomizing $\widehat{z}_i$. Note that Foster *et al.* [43] only considered the setting with a binary outcome; however, their methodology can be extended to other types of outcomes.

Consider a clinical trial with a binary outcome variable Y ($Y = 0$ or 1) and assume that $Y = 1$ represents a desirable outcome. In this case, $f(\mathbf{x}, t)$ denotes the conditional probability of the desirable outcome given the vector of patient's covariates $\mathbf{X} = \mathbf{x}$ and assigned treatment $T = t$. To estimate the underlying response function, Foster *et al.* [43] proposed to use random forests, although other methods of ensemble regression such as gradient boosting [64, 66] or Bayesian ensemble learning [67] can also be applied.

Random forest is an example of a 'black box' method where the resulting statistical 'model' cannot be expressed as a simple data-generating mechanism. Individual predictions from a random forest are obtained by averaging across predictions from many large (unpruned) trees grown by applying the CART method to multiple bootstrap samples of the original data. Each tree from the forest partitions the covariate space into non-overlapping regions and any particular patient is allocated to only one region based on his or her covariates. The final prediction from the forest for a patient with the covariate vector $\mathbf{x}$ is defined as the average of the predicted values across all the trees. Although predictions from each overgrown tree are unstable because of overfitting (exhibiting high variance and low bias), averaging across trees results in a substantial reduction in the prediction variance while retaining low bias. As a result, a random forest model may achieve a good trade-off between variance and bias and does not overfit the data.

To make averaging across trees even more efficient (in terms of reducing the prediction variance), an additional source of randomness is incorporated in the tree construction procedure. Specifically, when growing individual trees from bootstrap samples, the CART algorithm is modified so that the best splitting covariate is selected at each decision node from a random sample of m out of p candidate covariates (the covariates are sampled anew for each selection). The resulting trees become more diverse, which leads to a reduction in the correlations across the trees and a smaller prediction variance of the ensemble. Adding this feature actually distinguishes random forests from their precursor, namely, the *bagging* *method*, which is essentially a random forest without random variable selection. The random forest procedure can be easily automated as it is essentially controlled by only two tuning parameters: the number of trees (bootstrap samples) and the proportion m/p of covariates sampled at each node. Reasonable default values for the tuning parameters are available and implemented in the R package *randomForest*.

Returning to the VT method, the individual treatment contrasts can be computed in the first stage on the original scale, that is,

$$z_i = \hat{f}(\mathbf{x}_i, 1) - \hat{f}(\mathbf{x}_i, 0),$$

or on the logit scale,

$$z_i = \text{logit}\hat{f}(\mathbf{x}_i, 1) - \text{logit}\hat{f}(\mathbf{x}_i, 0),$$

where

$$f(\mathbf{x}_i, 1) = P(Y = 1|\mathbf{X} = \mathbf{x}_i, T = 1),\ f(\mathbf{x}_i, 0) = P(Y = 1|\mathbf{X} = \mathbf{x}_i, T = 0).$$

At the second stage, these estimated contrasts are used as 'observed' values of the response variable for growing a regression tree. The goal of this stage is to identify a patient subgroup S, where each treatment contrast is expected to be greater than a pre-specified clinically meaningful threshold denoted by δ. To prevent data overfitting, the tree size in CART is controlled using cost-complexity pruning with the complexity parameter selected via CV. The regression tree is fitted to the z's to obtain predictions denoted by $\hat{z}$. Note that the predicted outcome for a patient within a certain terminal node R is simply the average outcome value for the region associated with that node, that is,

$$\hat{z}(R) = \frac{1}{|R|}\sum_{j\in R} z_j.$$

This is a standard approach for making predictions with trees as piecewise-constant models. The final subgroup $\hat{S}$ is formed as the union of the terminal nodes, where the predicted values, that is, $\hat{z}(R)$, are greater than δ. The hat notation indicates that $\hat{S}$ is an estimate of the true subgroup of patients who experience a beneficial treatment effect.

As an alternative approach (termed VTC), a classification tree may be fitted to the binary outcome formed by dichotomizing the z's, that is, to $u_i = I(z_i > \delta)$. Then each terminal node R is classified into one of the two outcome groups based on the 'majority vote' within the node:

$$\hat{u}(R) = I\left(\frac{1}{|R|}\sum_{j\in R} u_j \geqslant 0.5\right).$$

The final subgroup $\hat{S}$ is defined as the union of the terminal nodes with the predicted outcome $\hat{u}(R) = 1$. One may reasonably argue that introducing an additional layer of uncertainty by dichotomizing the response variable may only result in sensitivity loss. Therefore, the VT method will be illustrated using the regression tree method only.

One of the key issues in any subgroup identification method is the assessment of the treatment effect within an estimated subgroup $\hat{S}$. To quantify the enhanced treatment effect, Foster et al. [43] proposed a measure of the treatment benefit denoted by $Q(S)$. The treatment benefit is defined as the 'excess' treatment effect in the true subgroup S over the overall population effect:

$$Q(S) = \{E(f(\mathbf{X}, 1)|\mathbf{X} \in S) - E(f(\mathbf{X}, 0)|\mathbf{X} \in S)\} - \{E(f(\mathbf{X}, 1)) - E(f(\mathbf{X}, 0))\}, \tag{3}$$

where the expectations are computed with respect to the covariate vector $\mathbf{X}$.

Alternatively, we can write $Q(S)$ in terms of the outcome variable Y as follows:

$$Q(S) = \{E(Y|\mathbf{X} \in S, T = 1) - E(Y|\mathbf{X} \in S, T = 0)\} - \{E(Y|T = 1) - E(Y|T = 0\}, \tag{4}$$

where the expectations are evaluated with respect to the outcome variable Y and $\mathbf{X}$.

Because the true subgroup S is unknown, the treatment benefit needs to be evaluated for the estimated subgroup $\hat{S}$, which is found using the VT method. In other words, the goal is to estimate $Q(\hat{S})$. It is tempting to simply use 'plug-in' estimates based on replacing the outcome function $f()$ with its estimate

$\widehat{f}()$ (obtained by the random forest method) in (3) and estimating the expectations in (3) and (4) using sample averages. This leads to the following *re-substitution* estimators of $Q(\widehat{S})$:

$$\widehat{Q}(\widehat{S}) = \left(\frac{1}{|\widehat{S}|} \sum_{i:\ x_i \in \widehat{S}} \widehat{f}(\mathbf{x}_i, 1) - \frac{1}{|\widehat{S}|} \sum_{i:\ x_i \in \widehat{S}} \widehat{f}(\mathbf{x}_i, 0) \right) - \left(\frac{1}{N} \sum_{i=1}^{N} \widehat{f}(\mathbf{x}_i, 1) - \frac{1}{N} \sum_{i=1}^{N} \widehat{f}(\mathbf{x}_i, 0) \right) \tag{5}$$

$$\widehat{Q}(\widehat{S}) = \left(\frac{1}{|\widehat{S}_1|} \sum_{i:\ x_i \in \widehat{S}_1} y_i - \frac{1}{|\widehat{S}_0|} \sum_{i:\ x_i \in \widehat{S}_0} y_i \right) - \left(\frac{1}{N_1} \sum_{i:\ t_i=1} y_i - \frac{1}{N_0} \sum_{i:\ t_i=0} y_i \right), \tag{6}$$

where $\widehat{S}_0$ and $\widehat{S}_1$ denote the subsets of untreated and treated patients from the subgroup $\widehat{S}$. Further, N_0 and N_1 denote the total sample sizes in the control and treatment arms, respectively, and N is the total sample size.

The re-substitution estimate (5) will be termed a *model-based* estimate and (6) a *data-based* estimate. In addition, one can consider an 'intermediate' method where the treatment effect is estimated as in (6) by contrasting the average outcomes between the two study arms; however, the actual outcomes are replaced with the predicted values computed from the model [43]. This approach is different from (5), where the potential treatment differences are computed and averaged over both study arms:

$$\widehat{Q}(\widehat{S}) = \left(\frac{1}{|\widehat{S}_1|} \sum_{i:\ x_i \in \widehat{S}_1} \widehat{f}(\mathbf{x}_i, 1) - \frac{1}{|\widehat{S}_0|} \sum_{i:\ x_i \in \widehat{S}_0} \widehat{f}(\mathbf{x}_i, 0) \right) - \left(\frac{1}{N_1} \sum_{i:\ t_i=1} \widehat{f}(\mathbf{x}_i, 1) - \frac{1}{N_0} \sum_{i:\ t_i=0} \widehat{f}(\mathbf{x}_i, 0) \right). \tag{7}$$

Unfortunately, the three estimates given in (5)–(7) are all biased estimators for $Q(\widehat{S})$. To understand why this is the case, recall that the subgroup $\widehat{S}$ was estimated from the same dataset, which is now used to compute the conditional expectations in (3) and (4) and therefore the basic re-substitution approach will likely overestimate the subgroup treatment contrast:

$$E(f(\mathbf{X}, 1)|\mathbf{X} \in S) - E(f(\mathbf{X}, 0)|\mathbf{X} \in S) \text{ or } E(Y|\mathbf{X} \in S, T = 1) - E(Y|\mathbf{X} \in S, T = 0).$$

This, in turn, will lead to a substantial optimistic bias in $Q(\widehat{S})$. This bias will be present despite the fact that the subgroup $\widehat{S}$ was selected from a reasonably pruned tree to prevent overfitting. While tree pruning via CV or other methods may ensure that $\widehat{S}$ is reasonably close to S, a separate validation dataset is generally needed to obtain an unbiased or 'honest' estimate of the treatment effect in the selected subgroup. In essence, to construct an unbiased estimate of $Q(\widehat{S})$, the expectations in (3) or (4) should be evaluated over an *independent* dataset, which is typically unavailable.

To find a solution to this problem, several resampling-based approaches to computing bias-corrected estimates of the treatment benefit $Q(\widehat{S})$ were proposed in [43]. These approaches were compared with the naive re-substitution approach defined previously using simulations. In what follows, we will introduce the bias-corrected estimate of $Q(\widehat{S})$ based on non-parametric bootstrap that was found to be most promising in [43].

First, B bootstrap samples are generated from the original dataset using sampling with replacement or other methods. Each bootstrap sample is processed using the same VT method as was applied to the original data, and a subgroup $\widehat{S}_b$ is identified along with the associated estimate $\widehat{Q}_b\left(\widehat{S}_b\right)$. The subscript b in $\widehat{Q}_b$ indicates that the Q-function in (3) or (4) is evaluated by taking the expectations over the data from the bth bootstrap sample. In addition, an estimate of the Q function for the subgroup $\widehat{S}_b$, denoted by $\widehat{Q}\left(\widehat{S}_b\right)$, is computed from the original dataset. The optimism bias of the naive re-substitution estimate $\widehat{Q}(\widehat{S})$ is estimated in the bth bootstrap sample as follows:

$$O_b = \widehat{Q}_b\left(\widehat{S}_b\right) - \widehat{Q}\left(\widehat{S}_b\right).$$

Note that the subgroup $\widehat{S}_b$ identified in the bth bootstrap sample may not (and likely would not) be the same subgroup that was found on the training data, which may appear counterintuitive. Some would argue that, to evaluate overoptimism associated with a specific subgroup $\widehat{S}$, we should rather evaluate this subgroup in each bootstrap sample and compute the optimism bias as the difference in the Q-function for $\widehat{S}$ evaluated on training and bootstrap data:

$$\tilde{O}_b = \widehat{Q}(\widehat{S}) - \widehat{Q}_b(\widehat{S}).$$

This quantity, however, would not fully account for the subgroup search process, in particular, for fitting a tree and selecting its terminal nodes to define the final subgroup in the second stage of the VT method.

The estimate of bias is found by averaging the bias estimates over the bootstrap samples, that is,

$$\widehat{Bias} = B^{-1} \sum_{b=1}^{B} O_b,$$

and the bias-corrected estimate of the treatment benefit in the subgroup $\widehat{S}$ is given by

$$\widehat{Q}_{cor}(\widehat{S}) = \widehat{Q}(\widehat{S}) - \widehat{Bias}. \tag{8}$$

The algorithm presented previously is similar to that used in the computation of bootstrap-based estimates for assessing prediction errors. Using the original dataset as a 'validation' set against the bootstrap sets has a caveat that most observations ($\approx 63.24\%$) of the original data will be included in a bootstrap sample when sampling with replacement (note that, based on simple random sampling with replacement, there is a $1/N$ chance that each individual observation will be selected in each draw, therefore, the probability that a given observation remains unselected after N draws is $(1 - 1/N)^N \approx e^{-1} \approx 0.368$). One can then decide to use only a portion of the original dataset which was not included in a particular bootstrap sample (about 36.8% of the data, termed 'out-of-bag' data) as a validation set. However, as it turns out, this procedure also introduces bias, this time in the direction of pessimism. This occurs merely as the consequence of reduced sample size; there are only about 63.2% of distinct observations from the original training set that can be used in each bootstrap sample to build the model. Hence, the performance assessed using a bootstrap-based approach should underestimate the performance expected when the entire training dataset is used. To achieve a balance between the optimism bias and pessimism bias, Efron [68] proposed (in the context of estimating prediction errors) to use a weighted average of the naive re-substitution and 'out-of-bag' estimates with the weights 0.368 and 0.632, respectively. The resulting estimate is known as the *0.632 estimator*. The same idea can be applied to the bias-corrected estimate derived previously, as suggested in [43]. Let $\widehat{Q}_{-b}(\widehat{S}_b)$ denote the Q function evaluated on the observations not included in the bth bootstrap sample for the subgroup $\widehat{S}_b$ identified from that sample. The 0.632 estimator for $Q(\widehat{S})$ is defined as

$$\widehat{Q}_{0.632}(\widehat{S}) = 0.368\, \widehat{Q}(\widehat{S}) + 0.632\, \frac{1}{B} \sum_{b=1}^{B} \widehat{Q}_{-b}(\widehat{S}_b). \tag{9}$$

In general, the reader should bear in mind that the estimators of $Q(\widehat{S})$ presented previously are heuristic in nature and their validity should be assessed by simulation in each individual setting.

7. Global treatment effect modeling

Biomarker evaluation methods in this class are attractive in that they bypass the problem of estimating the 'main effects' in a model (or, in other words, identification of biomarkers with purely prognostic properties) and focus instead on estimating the treatment contrast, which is directly related to the assessment of predictive biomarkers. As a result, global treatment effect modeling methods tend to be more robust to model misspecification compared with the global outcome modeling methods discussed in Section 6.

Negassa *et al.* [69] and Su *et al.* [70, 71] proposed the interaction trees (IT) method for identifying predictive biomarkers and associated subgroups of patients who are likely to experience a beneficial treatment effect. This method essentially extends the CART methodology [33] discussed in Section 5.3. Unlike regular tree-based approaches that fail to identify predictive biomarkers, IT incorporates a *treatment-by-split interaction* in the splitting criterion. Consider, for example, a clinical trial with a continuous outcome variable. The classical CART algorithm will fit the following main effect model within each parent node:

$$y_i = a_0 + a_1 s_i + \varepsilon_i,$$

where s_i is the indicator variable associated with a candidate split and then use the splitting criterion based on reducing the error sum of squares due to the split. By contrast, the IT method utilizes the following extended model:

$$y_i = a_0 + a_1 s_i + a_2 t_i + a_3 t_i s_i + \varepsilon_i,$$

which includes the treatment-by-split-interaction term ($t_i s_i$). The splitting criterion will be based on the reduction in the error sum of squares because of this interaction term (more generally, a variety of statistics for testing the hypothesis that the coefficient a_3 associated with the treatment-by-split-interaction is equal to zero can be considered to define the splitting criterion). Therefore, IT focuses on splits that make the resulting child nodes more heterogeneous with respect to the treatment contrast $z(\mathbf{X})$, whereas CART looks for splits that increase the heterogeneity with respect to the outcome function $f(\mathbf{X})$.

The IT method is similar to other nonparametric methods based on recursive partitioning in that it supports subgroup identification within a very broad 'model space'. The model space in recursive partitioning consists of all possible configurations of selected covariates and associated cutoffs, for example, a subgroup generated by an IT may be defined as

$$S(\mathbf{X}) = I\{25 < X_1 \leqslant 40,\ X_2 = \text{'Male'}\},$$

where X_1 is the patient's age and X_2 is the patient's gender.

Interaction trees classify all patients into a collection of the non-overlapping subgroups (terminal nodes). Patients within the same terminal node experience a similar treatment benefit, which is typically represented within the subgroup by a single value (the estimated treatment contrast). Therefore, IT provide a piecewise-constant fit for the underlying treatment effect $z(\mathbf{X})$. The IT methodology proposed in [71] also includes complexity pruning, merging nodes with homogeneous treatment effects (amalgamation algorithm), and evaluating variable importance (VI) scores via random forests. Section 7.1 provides a more detailed description of the IT method, and Section 10.6 presents an application of this method to Case study 2 with a time-to-event outcome.

Loh, He, and Man [72] recently proposed a novel recursive partitioning method for subgroup identification within the GUIDE framework. GUIDE is a suite of tree-based procedures introduced in Loh [73] and is different from CART and related methods such as the IT primarily in that it employs an unbiased variable selection method. The *selection bias* in tree-based subgroup search methods arises because different biomarkers have different sets of associated candidate splits, that is, cutoffs for continuous covariates or subsets of levels for categorical covariates. Most existing tree-based procedures select the best split by exhaustively cycling through all possible splits for each biomarker. As a result, a biomarker with a larger set of possible cutoffs has an advantage over another biomarker with a smaller set of cutoffs in the sense that the former is more likely to be selected by chance even if it is not associated with the outcome [23]. Instead of an exhaustive search, GUIDE implements a two-stage selection procedure. The procedure first selects the best covariate using a simple univariate test statistic that is adjusted for the number of possible splits for a given covariate, and at the second stage, it determines the associated optimal split for the covariate selected in the first stage.

It is worth noting that similar ideas, although developed within a different theoretical framework, were employed in the *conditional inference trees* method by Hothorn *et al.* [24] and extended to a more general setting of *model-based recursive partitioning* in [74]. This methodology is implemented in the R packages *party* and *MOB* that were later subsumed in the more generic package *partykit*.

Like the IT method, the procedures proposed in [72] aim at constructing trees where terminal nodes represent subgroups of patients with a homogeneous treatment effect. Consider, for example, the Gi method that appears the most promising among several methods discussed in [72]. At the first stage, this method uses a criterion that selects the covariate with the largest *lack of fit* test statistic in a regression model that includes only the treatment and candidate covariate X (if X is ordinal or continuous, it needs to be dichotomized at its mean value within the node considered for splitting). Because the treatment-by-covariate interaction term is omitted from the model, larger values of the associated lack-of-fit statistic can be attributed to the presence of a treatment-by-covariate interaction, which justifies selecting this particular covariate as a splitter with a high potential for predicting treatment effect. This variable selection criterion is easily generalized to settings with multiple treatments and different types of outcomes. After the best covariate X^* has been chosen, the optimal cutoff is selected by minimizing the residual sum of squares from the model that includes terms for the candidate cutoff (a binary variable), treatment, and treatment-by-split interaction. Confidence intervals for the treatment differences within the subgroups defined by the tree nodes are constructed using a bootstrap-based algorithm. The authors argue that good performance and lack of 'overoptimism' in the treatment effect estimates within the subgroups identified by this method can be explained by its 'non-greedy' nature, that is, unlike IT and related methods, Gi does not try to directly maximize the treatment-by-split interaction or other functions of the treatment effect in subgroups.

Dusseldorp and Mechelen [75] recently introduced another tree-based algorithm for subgroup identification (known as QUINT) that specifically aims at recovering qualitative interactions. The goal is to partition the overall patient population into regions (terminal nodes) of the following three types: (i) the experimental treatment is sufficiently better than the control, (ii) the control is sufficiently better than the experimental treatment, and (iii) all the rest. The algorithm sequentially finds an optimal split of one of the current terminal nodes by maximizing a splitting criterion that contrasts the treatment effect in the nodes of type 1 with that in the nodes of type 2 while also incorporating the associated sample sizes in the splitting criterion. The resulting full-grown tree is pruned to an optimal size using a bootstrap-based procedure.

As an example of a parametric approach within this class of methods, consider a simple method for estimating the individual treatment contrast $z(\mathbf{X})$ termed the *modified covariate* method [76]. Assuming a continuous outcome variable and equal randomization to the treatment and control arms, Tian *et al.* [76] observed that the treatment-by-covariate interaction effects can be fitted directly (without the need of modeling the main effects) by simply defining the outcome variable as $Z = 2T^*Y$, where the treatment indicator is defined as $T^* = 2T-1 \in \{-1, 1\}$, and then regressing Z on $\mathbf{X}$. This approach relies on the fact that, under a 1:1 randomization, $E(2T^*Y|\mathbf{X} = \mathbf{x}) = z(\mathbf{x})$. However, under the squared loss function for a continuous outcome, this is equivalent to fitting a model to the original outcome variable while multiplying the (mean-centered) covariate vector by $T^*/2$. Tian *et al.* [76] argued that this framework is easily extended to different types of outcomes, for example, binary, count, and survival outcome variables, supports efficient estimation methods, and deals with high-dimensional data by regularization. This simple approach produces a patient-specific predictive score, which can be used to stratify populations by the expected treatment effect and identify subgroups of patients who experience treatment benefit or harm.

Bayesian subgroup analysis can also be performed within the global treatment effect modeling framework. It focuses on modeling the data at the treatment contrast level rather than the treatment outcome level. For example, Jones *et al.* [77] presented a general framework for Bayesian subgroup analysis, which subsumes the models proposed in [46] and [47] as special cases. Within this framework, a hierarchical Bayesian model is introduced for the treatment contrast in a candidate subgroup. The treatment effect is shared by all patients within the subgroup (cell), and to simplify the notation, the subgroup and patient indices will be dropped. The treatment effect, denoted by θ, may be the mean difference, log-odds ratio, log-hazard ratio, and so on. The subgroup effects defined by single covariates are modeled simply by including their main effects, and subgroup effects defined by m covariates would require $(m-1)$-order interaction effects. As with any cell-mean modeling, this approach works best with a relatively small number of biomarkers.

As an example, consider the process of modeling the treatment effect as a function of binary biomarkers based on the patient's age (X_1), gender (X_2), and race (X_3). The resulting model for the treatment effect in this subgroup is defined as the sum of the overall effect, subgroup effects of the three biomarkers and associated second-order and third-order interactions as follows:

$$\begin{aligned}\theta = \mu &+ \gamma_1 I(X_1 \leqslant 50) + \gamma_2 I(X_2 = \text{`Male'}) + \gamma_3 I(X_3 = \text{`White'}) \\ &+ \delta_1 I(X_1 \leqslant 50) I(X_2 = \text{`Male'}) + \delta_2 I(X_1 \leqslant 50) I(X_3 = \text{`White'}) \\ &+ \delta_3 I(X_2 = \text{`Male'}) I(X_3 = \text{`White'}) + \alpha I(X_1 \leqslant 50) I(X_2 = \text{`Male'}) I(X_3 = \text{`White'}).\end{aligned}$$

The subgroup effects, second-order, and third-order interaction effects, that is, $\gamma = (\gamma_1, \gamma_2, \gamma_3)$, $\delta = (\delta_1, \delta_2, \delta_3)$ and α, are modeled using independent normal priors with zero means and separate variances. They can also be modeled as random variables within a Bayesian hierarchical model, allowing for differential amount of shrinkage for the associated subgroup effects. Note that modeling at the outcome level would require including covariate-by-treatment interactions up to the fourth order, as well as all prognostic effects. The connection between the observed data and unobservable subgroup treatment effects θ in the left-hand side of the aforementioned equation occurs via the first level of the hierarchy. For example, when modeling a binary outcome using the same three biomarkers, we can assume that the observed log-odds ratio in each candidate subgroup is generated by independent normal distributions centered around θ, that is, $\widehat{\theta} \sim N\left(\theta, \sigma_\theta^2\right)$.

7.1. *Interaction trees*

The IT method for survival outcomes [70] extends the methodology of survival trees developed in [78], which in turn borrows heavily from the original CART algorithm by Breiman *et al.* [33]. Like its predecessors, the IT procedure includes these three fundamental steps:

- Step 1: Growing a large initial tree.
- Step 2: Constructing a nested sequence of pruned trees.
- Step 3: Selecting the best-sized subtree from the sequence.

As mentioned in Section 4, the key feature of IT is that the splitting criterion is based on the test statistic for the treatment-by-split interaction in the model fitted 'locally' within each parent node intended for splitting. The model used for survival outcomes is a semiparametric Cox proportional hazards regression model with the terms for treatment (t), provisional split indicator (s), and their interaction:

$$h(u|t_i, s_i) = h_0(u)\exp(a_1 s_i + a_2 t_i + a_3 s_i t_i), \tag{10}$$

where $h_0(u)$ is the node-specific baseline hazard function (estimated non-parametrically) and u is the time from the initiation of treatment to the event of interest. The splitting criterion can be defined as the Wald test statistic or the likelihood-ratio test (LRT) statistic for testing the null hypothesis that for $a_3 = 0$. The former is faster to compute because it requires fitting a single model, that is, the outcome model (10), whereas the LRT statistic requires fitting the original model (10) and the following reduced model:

$$h(u|t_i, s_i) = h_0(u)\exp(b_1 s_i + b_2 t_i). \tag{11}$$

The LRT statistic is computed as $G(s) = -2(l_2 - l_1)$, where l_1 and l_2 are the partial log-likelihood values based on the models (10) and (11), respectively. While the Wald test statistic can be used in the tree-growing step (Step 1), the LRT statistic will be needed for pruning and selection of the best-sized tree (Steps 2 and 3) based on the interaction-complexity criterion, as will be explained shortly. Therefore, we assume that $G(s)$ is used throughout the entire process. Specifically, the tree growing algorithm in Step 1 proceeds as follows: for each parent node, the split s^* is selected to maximize $G(s)$ over all allowable splits for all candidate covariates. The algorithm is repeated recursively until a large tree T_0 is grown (given the pre-specified restrictions on the minimal sample size in each terminal node and other parameters that control the depth of the tree).

For Steps 2 and 3, we need to define the *interaction-complexity criterion* for a tree structure $\mathcal{T}$. Let $\mathcal{T}_{term}$ denote the set of terminal nodes of $\mathcal{T}$, $\mathcal{T} - \mathcal{T}_{term}$ the set of internal nodes, and $|\mathcal{T}|$ the number of nodes in $\mathcal{T}$. Note that $|\mathcal{T}|$ can be written as the number of terminal nodes is less than one and is equal to the number of splits needed to grow $\mathcal{T}$ from the root node. The interaction-complexity criterion is defined as

$$G_\alpha(\mathcal{T}) = G(\mathcal{T}) - \alpha(|\mathcal{T}_{term}| - 1), \tag{12}$$

where

$$G(\mathcal{T}) = \sum_{s \in \mathcal{T} - \mathcal{T}_{term}} G(s)$$

is the sum of the LRT statistics over all internal nodes of $\mathcal{T}$, which can be viewed as the total amount of treatment heterogeneity represented by the tree structure $\mathcal{T}$. Further, α is the amount of penalty associated with each additional split.

Starting from the initial tree $\mathcal{T}_0$, the algorithm of repeatedly removing the 'weakest link', that is, a branch of the tree the removal of which causes the smallest reduction in $G(\mathcal{T})$, is applied to produce a nested sequence of trees:

$$\mathcal{T}_M \subset \mathcal{T}_{M-1} \subset \ldots \subset \mathcal{T}_0,$$

where $\mathcal{T}_M$ is the tree consisting of the root node. This is essentially the same pruning procedure that was developed for CART by Breiman *et al.* [33]. The only difference is that the interaction-complexity criterion is used instead of the cost-complexity criterion utilized in the CART algorithm. The trees in the nested sequence are formed by maximizing the interaction-complexity criterion in (12) over all α's increasing from 0 to ∞. For example, this criterion is maximized by the initial tree ($\mathcal{T}_0$) if no penalty is applied ($\alpha = 0$). On the other hand, this criterion is maximized by the trivial tree, which consists of the root node if $\alpha = \infty$.

As shown in [33], in order to obtain all distinct trees that maximize the criterion, one does not need to examine the infinite set of α's. It is sufficient to start with $\alpha = 0$ and consider a finite sequence of the penalty values obtained by incrementing the previous penalty by the amount needed to remove the 'weakest link' in the current tree. Using this key observation, Breiman *et al.* [33] developed an efficient pruning algorithm for constructing a nested sequence of trees.

As a by-product of the pruning algorithm, a one-to-one mapping between the sizes of trees in the sequence and associated values of α is produced. The relationship is similar to that shown in Figure 2. Therefore, selection of the size of a sub-tree on the upper horizontal axis is equivalent to selection of the cost-complexity parameter, which is a scaled version of α shown on the lower horizontal axis. As we saw in Section 5, this task is accomplished in CART by means of CV, and the cost-complexity parameter is selected by using the 'minCV+1SE' rule. The final tree is easily constructed as the one that maximizes (12) with α set to the value estimated from CV.

Loosely speaking, the tree selection criterion in CART can be thought of as the penalized likelihood, which places a penalty at each split and the amount of penalty per split is estimated from the data (via CV or independent test data if available). Note that, in the development of survival trees by LeBlanc and Crowley [78], a different route was taken for selecting the final tree from the set of pruned trees. This approach was later adopted in [70]. In this approach, the penalty per split is a pre-specified constant, denoted by α_c, rather than a data-dependent quantity. This is similar to the popular Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) used in model selection, for example, $\alpha_c = 2$ is used in the AIC or $\alpha_c = \log(n)$ in the BIC. However, 'additional' penalty is applied to the partial likelihood $G(\mathcal{T})$—the amount of treatment heterogeneity associated with all the splits within a given tree structure $\mathcal{T}$—to account for the fact that the splits are not pre-specified but formed adaptively using the same dataset. The final sub-tree is selected from the nested sequence as the one that maximizes the following criterion:

$$\widehat{G}_{\alpha_c}(\mathcal{T}_i) = \widehat{G}(\mathcal{T}_i) - \alpha_c(|\mathcal{T}_{term,i}| - 1), \tag{13}$$

where $\widehat{G}(\mathcal{T}_i)$ is a bias-corrected estimate of $G(\mathcal{T}_i)$, which is obtained using a resampling-based method (see [70, 78] for details. Therefore, while α_c penalizes for increasing complexity (number of splits) of the tree, the bias-corrected estimate $\widehat{G}(\mathcal{T})$ accounts for overoptimism in $G(\mathcal{T})$ when it is computed by 'resubstitution', that is, from the same data that was used to construct the tree.

8. Modeling optimal treatment regimes

This section provides a brief overview of methods that fall within the second framework of methods for personalized medicine (Section 4.2) that emphasizes identifying an optimal treatment for a given patient rather than finding the 'best patient' for a given treatment. As we will see, estimating methods for developing OTRs often simplify to the methods aiming at direct estimation of the global treatment effect that were considered in Section 7.

Earlier in Section 4.1, we briefly introduced the concept of potential outcomes that plays an important role in defining OTRs. A *treatment regime* (also known as an *individual treatment rule*) $d(\mathbf{X})$ is defined as the function that maps a patient's covariate vector $\mathbf{X}$ to one of the available treatments choices. In the setting introduced in Section 4.1 with two treatment choices ($T = 0$, control arm; $T = 1$, experimental treatment arm), $d(\mathbf{X})$ is equal to 0 or 1. The potential outcome associated with a specific regime $d(\mathbf{X})$ is given by

$$\tilde{Y}(d(\mathbf{X})) = \tilde{Y}(1)d(\mathbf{X}) + \tilde{Y}(0)(1 - d(\mathbf{X})),$$

where $\tilde{Y}(1)$ and $\tilde{Y}(0)$ are the potential outcomes for a randomly chosen patient if this patient was allocated to the treatment and control arms, respectively. Specifically, $\tilde{Y}(d(\mathbf{x}))$ is the potential outcome for a patient with the covariate vector $\mathbf{X} = \mathbf{x}$ who exactly follows the treatment assignment rule $d(\mathbf{X})$.

In their groundbreaking paper, Qian and Murphy [79] introduced the notion of the *value function*, which is defined as the expected potential outcome for a specific treatment regime (assuming larger values of the outcome represent a patient's benefit):

$$V[d(\mathbf{X})] = E[\tilde{Y}(d(\mathbf{X}))].$$

The value function represents the *expected rewards*, which would be received if all patients followed the rule $d(\mathbf{X})$. An OTR is then defined as follows:

$$d_{opt}(\mathbf{X}) = \underset{d}{\operatorname{argmax}}\, V[d(\mathbf{X})].$$

It is easy to see how an optimal regime can be found for each patient if the outcome function $f(\mathbf{X}, T)$ were known. In this case, a patient with the covariate profile $\mathbf{x}$ should be assigned to $T = 1$ if $f(\mathbf{x}, 1) > f(\mathbf{x}, 0)$ and to $T = 0$ otherwise. Given this, a natural approach to estimating an OTR would be to utilize the following two-stage algorithm:

- Obtain an estimate of the outcome function $\widehat{f}(\mathbf{X}, T)$.
- Let $\widehat{d}_{opt}(\mathbf{X}) = I(\widehat{f}(\mathbf{X}, 1) > \widehat{f}(\mathbf{X}, 0))$.

A similar algorithm based on a penalized regression model with the lasso penalty was proposed in [79]. It was later pointed out by several researchers that more efficient approaches to estimating OTR can be defined by directly targeting the individual treatment contrast $z(\mathbf{X})$ rather than the global outcome function $f(\mathbf{X}, T)$ (for example, the review paper by Zhao and Zeng [80]). Indeed, if the outcome function has a simple linear form from Section 4.1, that is, $f(\mathbf{X}, T) = h(\mathbf{X}) + z(\mathbf{X})T$, the prognostic component $h(\mathbf{X})$ cancels out, and an optimal individual treatment rule depends on the covariates only through $z(\mathbf{X})$:

$$d_{opt}(\mathbf{X}) = I(z(\mathbf{X}) > 0). \tag{14}$$

To estimate OTRs without estimating $h(\mathbf{X})$, Lu, Zhang and Zeng [81] considered the following model with an arbitrary 'baseline' function $h(\mathbf{X})$ plus the predictive effects modeled as a linear function of the covariates:

$$f(\mathbf{X}, T) = h(\mathbf{X}) + z(\mathbf{X})T,$$
$$z(\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j.$$

Further, following the A-learning framework [11], a penalized regression procedure which provides a consistent estimate of the model parameters without actually fitting the baseline function $h(\mathbf{X})$ was developed. Specifically, an adaptive lasso penalty was utilized and the following loss function was used:

$$L(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \left[y_i - h(\mathbf{x}_i) - \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) (t_i - \pi(\mathbf{x}_i)) \right]^2,$$

where $h(\mathbf{x})$ is an arbitrary function capturing prognostic effects, for example, a linear function or simply a constant, and $\pi(\mathbf{x})$ is the probability of assigning a patient with an observed covariate profile $\mathbf{x}$ to the experimental treatment, that is, $\pi(\mathbf{x}) \equiv 1/2$ in a clinical trial is with 1:1 randomization (this formulation can be thought if as a generalization of [76]). Foster *et al.* [82] considered the same class of models, however with both $h(\mathbf{X})$ and $z(\mathbf{X})$ modeled non-parametrically and estimated via a back-fitting algorithm.

A different route can be taken, which is implicit but not recognized in [79]. The value function can be expressed in terms of the expectation over the *observed outcomes* via the following fundamental expression:

$$V[d(\mathbf{X})] = E\left[\frac{I(T = d(\mathbf{X}))}{P(T = d(\mathbf{X})|\mathbf{X})} Y \right], \tag{15}$$

where the random variable Y is the observed outcome and $P(T = d(\mathbf{X})|\mathbf{X})$ is the probability of being assigned to the treatment selected by the rule $d(\mathbf{X})$. The expectation is taken with respect to the joint distribution of random variables $Y, \mathbf{X}, T$ under the condition that every patient follows the rule $d(\mathbf{X})$. Because of that, we can replace $P(T = d(\mathbf{X})|\mathbf{X})$ in the denominator with $P(T = t|\mathbf{X})$, the probability that the patient is assigned the treatment that was actually observed (which will be shorthanded as $P(t|\mathbf{X})$). In other words, (15) states that the reward from following a treatment regime $d(\mathbf{X})$ is equal to the expected outcome in a subset of patients who actually followed that regime, inversely weighted by the probability of being assigned to the regime. An OTR can now be estimated by directly maximizing the rewards in (15) with respect to $d(\mathbf{X})$ or, equivalently, minimizing the expression where the expectation is evaluated for patients who *did not* follow the regime $d(\mathbf{X})$:

$$d_{opt}(\mathbf{X}) = \operatorname*{argmin}_{d} E\left[\frac{I(T \neq d(\mathbf{X}))}{P(t|\mathbf{X})} Y \right], \tag{16}$$

Substituting the representation (14) for $d(\mathbf{X})$ in (16), we obtain

$$z_{opt}(\mathbf{X}) = \operatorname*{argmin}_{z} E\left[\frac{I(T \neq I(z(\mathbf{X}) > 0))}{P(t|\mathbf{X})} Y \right]. \tag{17}$$

We now observe that (16) and (17) can be readily interpreted as the weighted classification error or misclassification loss when fitting a binary classifier to the actual treatment assignments T. As a result, the OTR problem can be viewed as a (weighted) classification problem using the familiar loss functions for binary outcomes. The classifier is modeled as the underlying treatment contrast $z(\mathbf{X})$ that needs to be known only up to its sign, that is, sign $z(\mathbf{X})$. This key observation led Zhao *et al.* [83] to develop the outcome-weighted learning (OWL) methodology for estimating OTRs. As was shown in [83], estimating optimal individualized treatment policies can be framed as a classification problem where the optimal classifier corresponds to the OTR. Within this framework, the treatment assignment variable plays a role of the outcome variable, and a certain 'outcome-based weight' is applied to each patient. Specifically, $w_i = y_i/\pi_i$ for the treated patients and $w_i = y_i/(1-\pi_i)$ for the untreated patients, where

$$\pi_i = \widehat{P}(T = 1|\mathbf{X} = \mathbf{x}_i)$$

is the estimated probability of assigning the ith patient to the experimental treatment arm (note that in a randomized treatment clinical trial it can be estimated simply as the proportion of patients with $T = 1$).

The idea is that the assigned weights take into account patients' observed outcomes in such a way that the misclassification costs will be minimized if patients with desirable outcomes are assigned to the study arm that they were actually assigned to and patients who did not experience much benefit under their current treatment are assigned to the other arm. Therefore, minimizing the weighted misclassification costs will entail assigning a patient to the treatment that provides maximum benefits given the patient's biomarker values. Any machine learning method for predicting binary or multinomial outcomes that support patient-specific weights can then be adopted for estimating an OTR within this framework. For example, the machine learning method of support vector machines was used in [83].

Zhang *et al.* [84] also considered the problem of estimating OTRs as a classification problem under a more general framework. They proposed to fit a weighted classifier, for example, a CART or SVM model, to the class labels that are defined by evaluating the sign of the individual treatment contrast, that is, $I(z(\mathbf{x}_i) > 0)$. As in the VT method presented in Section 6.4, the treatment contrast is estimated using the framework of potential outcomes; however, the outcome model is combined with the probability of a treatment model in a doubly robust augmented inverse probability weighted estimator (AIPWE), which is defined as follows (see also [85]):

$$\widehat{z}_{AIPWE}(\mathbf{x}_i) = \frac{t_i}{\pi_i} y_i - \frac{1-t_i}{1-\pi_i} y_i - \frac{t_i - \pi_i}{\pi_i} \widehat{f}(\mathbf{x}_i, 1) - \frac{t_i - \pi_i}{1-\pi_i} \widehat{f}(\mathbf{x}_i, 0).$$

The weighted classifier therefore will produce decision rules for discriminating patients who benefit from the experimental treatment from those who benefit from the control treatment. The patient-specific weights are taken as the absolute values of the estimated treatment contrasts, that is, $|\widehat{z}_{AIPWE}(\mathbf{x}_i)|$. Thus, the patients for whom the choice of treatment does not make much difference exert less influence on the decision rule.

As shown in [84], the OWL method developed in [83] can be considered as a special case of their approach, when the treatment contrast $z(\mathbf{x})$ is estimated using the inverse probability weighted estimator rather than AIPWE. The approach of Zhang *et al.* [84] can also be viewed as a generalization of the VT method in that the AIPWE estimator of the hypothetical treatment difference, that is, $\widehat{z}_{AIPWE}(\mathbf{x}_i)$, subsumes the estimator of $z(\mathbf{x}_i)$ in the VT method. The advantage of AIPWE is that it is more efficient and is consistent even when the outcome model may be misspecified but the treatment model is not. Note that this condition trivially holds for a randomized clinical trial where the probability of treatment assignment is known. When the data come from an observational study with non-random treatment assignments, the doubly robust AIPWE estimator protects against model misspecification as long as at least one of the two models (models for the outcome and treatment assignment) is correctly specified.

The optimal rule for selecting patients to be treated is estimated as $d(\mathbf{X}) = I(z(\mathbf{X}) > 0)$ and may be quite complex depending on the richness of the function space used to estimate $z(\mathbf{X})$ (see the next section). One approach to render the rule more clinically interpretable and manageable is to enhance the OTR procedure with an additional step where the estimated rule $I(z(\mathbf{X}) > 0)$ (or $I(z(\mathbf{X}) > \delta)$, where $\delta > 0$ is the clinically meaningful effect size) will be approximated with a simpler rule or set of rules, involving a smaller number of covariates and assuming a more natural form corresponding to the notion of subpopulation. A procedure of this kind, known as simple optimal regime approximation (SORA), was recently proposed and implemented in [82]. 'Simple rules' were defined using multidimensional boxes or regions formed by the intersection of univariate regions such as $\{x \leqslant a\}$ or $\{x > a\}$, where a

is selected from a grid of pre-defined values. The final multidimensional box can be defined using up to three covariates and constructed in a 'stepwise' fashion.

Fu *et al*. [86] used an exhaustive search procedure for evaluating simple rules similar to those defined in [82] and choosing the one that directly maximizes the associated value function estimated using the outcome-weighted criterion similar to (17), which was made more stable by subtracting an estimate of the mean function from the outcome: $Y - m(\mathbf{X})$. The mean function $m(\mathbf{X})$ does not have to be correctly specified. As reported in [86], using a simple linear regression with no treatment variable resulted in substantial improvement in numerical stability. Laber and Zhao [87] developed a tree-based procedure that recursively splits each parent node into two nodes such that all patients who fell into the same child node are assigned the same treatment. An optimal split and associated treatment assignments for the two child nodes are found by maximizing the value function, again estimated using the outcome-weighted criterion based on (17) and stabilized by subtracting the estimated mean function $m(\mathbf{X})$ from the outcome. Their choice of the mean function was more involved than in [86] and based on predicted outcome under a crude estimate of the optimal rule $d_{cr}(\mathbf{X})$, that is,

$$m(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x}, T = d_{cr}(\mathbf{x})).$$

This mean function was estimated non-parametrically by the random forest method. See also [88] where decision lists were used to construct interpretable and parsimonious treatment regimes.

8.1. *Penalized regression framework for OTR*

In this section, we consider the OWL method for estimating OTRs developed in [83]. We further focus on a simpler implementation of this method, which is similar in spirit to [27].

Several comments can be made regarding the problem of minimizing the expected misclassification loss function in (17). First, direct minimization of (17) is a computationally challenging problem because it leads to non-convex optimization. Therefore, as is often performed in the classification literature, direct minimization of the misclassification error (0–1 loss) is replaced with minimization of an appropriately selected smooth convex loss function. The implication of replacing the 0–1 loss with a 'surrogate loss' for (17) is that, instead of estimating the true $d_{opt}(\mathbf{x})$, a slightly different target $d^*_{opt}(\mathbf{x})$ induced by the surrogate loss function is estimated, namely, $L(t, z(\mathbf{x}))$. See Section 6.2 where several loss functions for binary outcomes (the treatment indicator in this context), such as the (negative) binomial log-likelihood or hinge loss, are considered. The solution based on the 'smooth' version of the loss function is $z(\mathbf{X}|\boldsymbol{\beta}^*)$, with

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, E[L(T, z(\mathbf{X}|\boldsymbol{\beta}))W],$$

where $W = Y/P(t|\mathbf{X})$, $z(\mathbf{X}|\boldsymbol{\beta})$ is the treatment contrast function parameterized using a vector of the regression coefficients $\boldsymbol{\beta}$, and the optimal treatment assignment rule is

$$d^*_{opt}(\mathbf{X}) = I(z(\mathbf{X}|\boldsymbol{\beta}^*) > 0).$$

While this change in the estimation target (from $\boldsymbol{\beta}$ to $\boldsymbol{\beta}^*$) is generally rather negligible, it immediately opens the gates for computationally efficient methods for estimating $d_{opt}(\mathbf{X})$. The problem is reduced to a (weighted) regression for a binary outcome.

Secondly, as in any regression problem with a large number of candidate covariates, some form of regularization is desirable, which is typically accomplished by introducing penalties as described in Section 6.2 (e.g., l_1, l_2, and elastic net penalties). Note that the feature space for estimating an OTR using OWL is reduced compared with that considered in the examples presented in Section 6.2 because the treatment-by-covariate interactions do not need to be included in the model. However, this may still be a high-dimensional problem due to a large number of covariate-by-covariate interactions.

Finally, as in penalized regression models considered in Section 6.2, one needs to define a class of functions that will be used in regression modeling, that is, specify the form of $z(\mathbf{X}|\boldsymbol{\beta})$. This class will ultimately define the form of treatment assignment rules. On the one extreme, one can consider using basis function expansions, for example, natural splines, wavelets, trees, or kernel methods of feature expansion, leading to 'black box' models with uninterpretable rules. On the other extreme, an OTR may be constrained to lie in the space defined by simple linear functions of the covariates or a single-tree model that can be readily interpreted by clinical trial researchers.

We note from this discussion that the OWL approach essentially reduces the problem of finding an OTR to a class of penalized regression problems for binary outcomes. Therefore, many different subtypes of OWL can be devised as one can come up with many different ways to fit weighted regression models for a binary outcome by making different choices of the loss, penalty, and basis expansion functions.

As a simple example, consider a randomized clinical trial with a 1:1 randomization and let π denote the probability of assigning a patient to the experimental treatment arm. We can construct parsimonious models for determining an OTR in a continuous outcome setting using the following steps:

- Fit a weighted logistic regression model with the lasso penalty and (negative) binomial loss to the treatment labels, that is,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ n^{-1} \sum\nolimits_{i=1}^{n} L[t_i, z(\mathbf{x}_i|\boldsymbol{\beta})] w_i + \lambda \sum\nolimits_{j=1}^{p} |\beta_j| \right\},$$
$$w_i = \begin{array}{c} y_i/\pi, \text{ if } t_i = 1, \\ y_i/(1-\pi), \text{ if } t_i = 0, \end{array}$$

and $z(\mathbf{x}_i|\boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$ is modeled as a linear predictor for the probability of treatment selection on the logit scale.

- Optimal treatment assignment rule for patient with covariate profile $\mathbf{X} = \mathbf{x}$ is estimated as

$$\hat{d}^*_{opt}(\mathbf{x}) = I\left(z\left(\mathbf{x}|\hat{\boldsymbol{\beta}}\right) > 0\right).$$

As mentioned earlier, Zhao *et al.* [83] pioneered the OWL approach and proposed using the hinge loss with the l_2 penalty and kernel basis expansion to model $z(\mathbf{x})$ in problems with a continuous outcome variable. Huang and Fong [89] considered binary outcome settings and proposed a novel *ramp loss* function, which better mimics the original 0–1 classification loss and l_2 penalty expanded by using non-linear kernels. An interesting feature of the Huang–Fong method is that the use of the 'treatment/disease burden ratio' δ, $0 \leqslant \delta \leqslant 1$, in treatment assignment rules. This ratio is defined as a fraction of disease burden caused by a single event. For example, $\delta = 0.05$ means that the treatment induces a burden (e.g., in terms of patient's safety or extra costs) equivalent to 5% of the burden caused by the occurrence of one undesirable event. In this case, the reduction in the probability of an undesirable event by 5% with a new treatment would be essentially washed out by the 5% increase in the treatment burden. The challenge of introducing the treatment burden parameter on the probability scale is in that the OTR for the case of $\delta > 0$ cannot be expressed based on a simple linear model for $z(\mathbf{X})$. On the other hand, if δ was defined on the same scale, which is used to model the treatment contrast $z(\mathbf{X})$, for example, a logit scale as in our example of penalized logistic regression, the rule for $\delta > 0$ could be expressed in terms of a similar simple model, for example, as $\hat{d}^*_{opt}(\mathbf{x}) = I(z(\mathbf{x}|\hat{\boldsymbol{\beta}}) > \delta)$. This consideration justified modeling $z(\mathbf{X})$ as a complex function via nonlinear kernels in [89]. One may, however, argue that specifying the treatment burden on the logit scale (even if somewhat less clinically interpretable) would be desirable from the modeling simplicity perspective.

On the opposite extreme in the landscape of OWL-based methods, we find the proposal of Xu *et al.* [27] who considered the simplicity and interpretability of the treatment assignment rule as the main goal. The regularized outcome weighted subgroup identification (ROWSi) method introduced by Xu *et al.* [27] is based on fitting simple linear models with the weighted (negative) binomial loss and lasso penalty for continuous predictors. This approach ensures sparseness in the solution and facilitates the interpretability of the resulting rules. Predictive biomarkers can be easily identified by inspecting the effects with non-zero coefficients. The following example can be used to illustrate the argument for using simple linear models for modeling the treatment contrast function. Assume that the outcome model belongs to the following class of models, which includes generalized linear models,

$$E(Y|\mathbf{X}, T) = g\{h(\mathbf{X}) + l(z(\mathbf{X})T)\}, \tag{18}$$

where $z(\mathbf{X})$ is a simple linear model of the covariates, that is, $z(\mathbf{X}|\boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j$, $g(\cdot)$ and $l(\cdot)$ are increasing functions and $h(\cdot)$ is an arbitrary function. The optimal treatment assignment rule can be expressed as $I(z(\mathbf{X}|\boldsymbol{\beta}) > 0)$ and the regression coefficients can be consistently estimated by minimizing a 'smooth' version of the loss function $L(T, z(\mathbf{X}))$ with an appropriate penalty to accommodate a high-dimensional covariate space (as illustrated in the aforementioned example). Note that, while $h(\mathbf{X})$ may depend on the covariates in a very complex way, the optimal rule is quite simple and is recovered from the data without estimating $h(\mathbf{X})$. Under relatively mild assumptions about the form of the outcome function, which includes generalized linear models (18) as a special case, the optimal rule depends only on the sign of a linear function of the covariates (see Proposition 1 of Xu *et al*. [27]).

The framework of Xu *et al*. [27] provides a powerful extension of the original proposal by Zhao *et al*. [83] to different types of outcomes. Specifically, time-to-event outcomes with censored times can be modeled within the OWL framework as easily as continuous outcomes because many common models for survival outcomes assume a functional form covered by Proposition 1 of Xu *et al*. [27]. To illustrate, let Y denote a time-to-event outcome variable and consider the Aalen–Cox model with the hazard function given by

$$\lambda_Y(y|\mathbf{X},T) = \lambda_0(y|\mathbf{X})\exp(-z(\mathbf{X}|\boldsymbol{\beta})T), \tag{19}$$

where the baseline hazard $\lambda_0(y|\mathbf{X})$ does not depend on the treatment assignment and $z(\mathbf{X}|\boldsymbol{\beta})$ is the treatment contrast modeled as a linear function of the covariates. If the outcome variable Y is completely observed, the optimal treatment assignment rule is based on the treatment contrast function $z(\mathbf{X}|\boldsymbol{\beta})$ and can be estimated by using a weighted logistic model with the weights $Y/P(t|\mathbf{X})$. However, if Y is only partially observed because of (non-informative) censoring, as is most often the case, the optimal rule is expressed using the *same* contrast function $z(\mathbf{X}|\boldsymbol{\beta})$ as in (19). This function is now estimated via a weighted logistic model with the weights $\tilde{Y}/P(t|\mathbf{X})$, where $\tilde{Y}$ is the observed and possibly censored value of Y.

Considering binary outcomes, let n, n_1, and n_0 denote the total number of patients in the study and the number of patients randomized to treatment and control arms, respectively. Note that, when the outcome-weighted misclassification loss function $L(T,z(\mathbf{X}))$ is computed, the weights for the negative outcome $(Y=0)$ are all equal to zero and only patients with a positive outcome $(Y=1)$ can be used to estimate the treatment contrast (with the weights equal to $1/\pi(\mathbf{X}) = n/n_1$). This is clearly undesirable; however, one can easily incorporate data from the other patients to compute the treatment assignment rule. First, the rule can be estimated by using only patients with a negative outcome by simply switching the treatment labels in the loss function, that is, by setting $T^* = 1 - T$ and using $1/(1-\pi(\mathbf{X})) = n/n_0$ as the weights. Then, as argued in [89], because the minimizer of the two sets of losses also minimizes their linear combination, the losses can be combined as a linear combination with, say, equal weights. Therefore, a single loss function can be defined as

$$\begin{aligned}
L(\boldsymbol{\beta}) &= n^{-1}\sum\nolimits_{i=1}^{n} L[t_i^*, z(\mathbf{x}_i|\boldsymbol{\beta})]w_i,\\
t_i^* &= \begin{cases} t_i, & \text{if } y_i = 1,\\ 1-t_i, & \text{if } y_i = 0,\end{cases}\\
w_i &= \begin{cases} n/(2n_1), & \text{if } t_i = 1,\\ n/(2n_0), & \text{if } t_i = 0.\end{cases}
\end{aligned} \tag{20}$$

This approach leads to a simple method for personalized medicine that was discovered independently by many researchers and presented under different names. We will use this simple method in Case study 1 with a binary outcome variable in Section 10.4.

Xu *et al*. [27] defined two important measures, denoted by $d_+(\widehat{\boldsymbol{\beta}})$ and $d_-(\widehat{\boldsymbol{\beta}})$ to help quantify the performance of a treatment assignment rule and proposed inferential methods based on these quantities. These measures are conceptually similar to the measures of enhanced treatment effect in the identified subgroup $Q(\widehat{S})$ introduced in Section 6.4. The identified subgroup corresponds here to the group of patients who are allocated to the experimental treatment by the treatment assignment rule, that is,

$$\widehat{S} = \left\{\mathbf{x} : z\left(\mathbf{x}|\widehat{\boldsymbol{\beta}}\right) > 0\right\}.$$

The measures summarize the average treatment effect for the patients allocated to the experimental and alternative treatment arms:

$$\begin{aligned}
d_+(\widehat{\boldsymbol{\beta}}) &= E\left\{E(Y|z\left(\mathbf{X}|\widehat{\boldsymbol{\beta}}\right) > 0, T=1) - E(Y|z\left(\mathbf{X}|\widehat{\boldsymbol{\beta}}\right) > 0, T=0)\right\},\\
d_-(\widehat{\boldsymbol{\beta}}) &= E\left\{E(Y|z\left(\mathbf{X}|\widehat{\boldsymbol{\beta}}\right) < 0, T=0) - E(Y|z\left(\mathbf{X}|\widehat{\boldsymbol{\beta}}\right) < 0, T=1)\right\},
\end{aligned} \tag{21}$$

where the inner expectation is evaluated with respect to Y and the outer with respect to $\mathbf{X}$. An ideal treatment assignment rule maximizes both $d_+(\widehat{\boldsymbol{\beta}})$ and $d_-(\widehat{\boldsymbol{\beta}})$.

To construct confidence intervals for $d_+(\hat{\boldsymbol{\beta}})$ and $d_-(\hat{\boldsymbol{\beta}})$, Xu *et al.* [27] proposed to employ a resampling method. Specifically, the m-out-of-n bootstrap was utilized to account for the discontinuity in $d_+(\hat{\boldsymbol{\beta}})$ and $d_-(\hat{\boldsymbol{\beta}})$ because of a non-negligible probability of $z(\mathbf{x}|\hat{\boldsymbol{\beta}}) = 0$ (this occurs when there are multiple categorical covariates). First, the data are sampled with replacement $m < n$ times (m is chosen as n^α for some $\alpha < 1$, for example, $\alpha = 0.8$) and the estimate of $\hat{\boldsymbol{\beta}}$ obtained from the bth bootstrap sample is denoted by $\hat{\boldsymbol{\beta}}^{(b)}$. The sample estimates of the average treatment effects are then computed as follows:

$$\tilde{d}_+\left(\hat{\boldsymbol{\beta}}^{(b)}\right) = \frac{1}{\left|S_1^{(b)}\right|}\sum_{i\in S_1^{(b)}} y_i - \frac{1}{\left|S_0^{(b)}\right|}\sum_{i\in S_0^{(b)}} y_i,$$

$$\tilde{d}_-\left(\hat{\boldsymbol{\beta}}^{(b)}\right) = \frac{1}{\left|\bar{S}_1^{(b)}\right|}\sum_{i\in \bar{S}_1^{(b)}} y_i - \frac{1}{\left|\bar{S}_0^{(b)}\right|}\sum_{i\in \bar{S}_0^{(b)}} y_i,$$

where the summations are over the sets of indices (subgroups) defined as

$$S_0^{(b)} = \left\{i : z\left(\mathbf{x}_i|\hat{\boldsymbol{\beta}}^{(b)}\right) > 0, t_i = 0\right\},$$

$$S_1^{(b)} = \left\{i : z\left(\mathbf{x}_i|\hat{\boldsymbol{\beta}}^{(b)}\right) > 0, t_i = 1\right\},$$

$$\bar{S}_0^{(b)} = \left\{i : z\left(\mathbf{x}_i|\hat{\boldsymbol{\beta}}^{(b)}\right) \leqslant 0, t_i = 0\right\},$$

$$\bar{S}_1^{(b)} = \left\{i : z\left(\mathbf{x}_i|\hat{\boldsymbol{\beta}}^{(b)}\right) \leqslant 0, t_i = 1\right\}.$$

Each bootstrap sample gives rise to its own estimate of the treatment assignment rule, which is applied back to the original dataset. This approach helps reduce the overoptimism bias associated with the standard resubstitution estimates of $d_+(\hat{\boldsymbol{\beta}})$ and $d_-(\hat{\boldsymbol{\beta}})$ (similar to that of $Q(\hat{S})$ discussed in Section 6.4). Finally, the $(1-\alpha)100\%$ confidence intervals for $d_+(\hat{\boldsymbol{\beta}})$ and $d_-(\hat{\boldsymbol{\beta}})$ are formed using the $\alpha/2$ and $1-\alpha/2$ quantiles of the bootstrap distributions $\tilde{d}_+\left(\hat{\boldsymbol{\beta}}^{(b)}\right)$ and $\tilde{d}_-\left(\hat{\boldsymbol{\beta}}^{(b)}\right)$, $b = 1, \dots, B$, respectively.

9. Local modeling

Section 8 presented methods for estimating OTRs that arise within the second framework of personalized medicine defined in Section 4.2, and this section will focus on methods aiming at direct subgroup search (local modeling methods) that are typically developed within the first framework. Local modeling methods deal with direct identification of regions in the covariate space with desirable properties such as an improved treatment effect $z(\mathbf{x})$. Within this approach, the interest lies in studying specific subsets of the space, and there is no longer a need to estimate the outcome function $f(\mathbf{x}, t)$ over the entire covariate space.

Examples of local modeling approaches include the responder identification procedure aimed at discovering subgroups of treatment responders in clinical trials [90] and subgroup identification based on differential effect search (SIDES) method [91]. The latter procedure was constructed as an extension of the patient rule induction method (PRIM) proposed by Friedman and Fisher [92]. When developing PRIM, Friedman and Fisher [92] argued that the problem of deriving a general model for the outcome function $f(\mathbf{x}, t)$ is a challenging problem and forces the researcher to extend models over the subsets of the covariate space that are, in fact, 'uninteresting' for the research goal. It is often more sensible to shift the focus to the problem of *bump hunting*, that is, examining local features of the covariate space, known as bumps, such as regions with a strong treatment effect.

Indeed, as was shown in Section 6, global outcome modeling methods rely on increasingly more complex models that attempt to develop predictions for the individual treatment contrast $z(\mathbf{x})$ for any covariate vector $\mathbf{x}$. The requirement to achieve accurate predictions over the high-dimensional space may result in compromising the quality of predictions within regions that would be of most interest from a practical perspective, including covariate values that correspond to an enhanced treatment effect.

The main goal of bump hunting methods such as PRIM is to define sets of multivariate rectangular regions based on the candidate covariates $X_1, \dots, X_p$. For example, assuming for simplicity that all covariates are continuous, a rectangular region is defined as

$$S = \bigcap_{i=1}^{p} \{l_i \leqslant X_i \leqslant u_i\},$$

where l_i and u_i the lower and upper limits for the ith covariate. The limits are determined in a data-driven manner using a *peeling* technique. Specifically, extreme values of continuous/ordinal covariates or individual levels of nominal covariates are removed. The peeling algorithm is sequentially applied to a single covariate, and the order of the covariates is determined by the value of an appropriate objective function. The tuning parameter used in the peeling algorithm is the maximum proportion of observations to be removed at a time. This parameter can be calibrated via CV. It is instructive to compare bump hunting methods based on the peeling algorithm to recursive partitioning methods discussed, for example, in Section 7.1. With tree-based procedures, each split reduces the size of the current subgroup on average by 50% and bump hunting procedures generally proceed at a slower pace.

The peeling process is fairly unstable, and the performance of the peeling algorithm can be improved by combining this algorithm with a *pasting* algorithm. A pasting algorithm adds observations to selected regions to maximize the objective function used in the peeling algorithm. Friedman and Fisher [92] demonstrated that, within PRIM, the two algorithms can be applied iteratively. The resulting regions of interest are defined as unions of rectangular regions in the covariate space.

When Kehl and Ulm [90] introduced the responder identification procedure to extend PRIM to clinical trial settings, the key step within the new procedure was estimation of the outcome function in the control arm, that is, $f(\mathbf{x}, 0)$. An improved version of the responder identification procedure proposed in [93] bypassed this step by introducing a flexible objective function in the PRIM framework. The modified procedure supports direct search for patient subgroups with positive treatment effects while ensuring that the identified subgroups maintain a differential effect compared with the complementary subgroups (this is accomplished by imposing an *interaction effect constraint*).

As stated earlier, other important examples of local modeling methods are the SIDES method [91], and its extension known as the SIDEScreen procedure [94]. The SIDES method utilizes recursive partitioning to perform a direct search for subgroups of patients who experience a treatment benefit. The recursive partitioning algorithm is applied to each individual candidate biomarker and an optimal split is found by maximizing a pre-defined differential effect criterion. The SIDES method employs complexity control to reduce the size of the search space and multiplicity adjustments to account to selection bias inherent in subgroup search. The SIDEScreen procedure improves on the basic SIDES procedure by introducing a biomarker screen. A detailed description of the general SIDES method is presented in Section 9.1, and SIDEScreen is defined in Section 9.7. The SIDES method is illustrated in Sections 10.1 and 10.5 where it is applied to Case studies 1 and 2.

Bayesian approaches within the local modeling framework were developed in [25, 95]. These approaches treat candidate patient subgroups as 'submodels' and perform inferences by applying Bayesian model averaging algorithms. Berger *et al.* [25] proposed to define each submodel as a combination of possible predictive and prognostic (or baseline) effects using 10 different basic subgroup structures. Within each such structure, the predictive and prognostic components are modeled locally by subsetting on a small number of dichotomous biomarkers (up to one variable in the prognostic or/and predictive components). A continuous outcome variable within a 'local' submodel M may be driven by a prognostic effect defined by the variable X_l and a predictive effect defined by the variable X_m, $l, m = 1, \dots, k$, that is,

$$y_i | M = \mu + \alpha I(x_{il} = 0) + t_i \beta I(x_{im} = 0) + \varepsilon_i, \; \varepsilon_i \sim N(0, \sigma^2).$$

Berger *et al.* [25] showed how to elicit priors for each submodel in the model space for all possible combinations of prognostic and predictive effects and indicated that assigning priors 'accounts for multiplicity and yet allows for (pre-experimental) preference to specific subgroups'. Various posteriors summaries can be computed at the individual patient level by model averaging. For example, prediction of the treatment effect for a patient with a specific biomarker profile is based on averaging predictions for that patient across all models weighted by their respective posterior probabilities.

9.1. General SIDES method

This section introduces the general SIDES method and its extension (SIDEScreen procedure) for biomarker exploration and identification of subgroups of patients who derive substantial benefit from the experimental treatment. The most important features of the SIDES method include the following:

- Complexity control, which is applied to reduce the size of the search space and produce easy-to-interpret results, and lessen the multiplicity burden.
- Multiplicity control, which is employed to preserve the probability of incorrect subgroup identification.
- Biomarker screen, which is introduced to filter out non-informative biomarkers.

The original SIDES procedure [91], which will be referred to at the SIDESbase procedure, is introduced first, followed by the SIDEScreen procedure [94]. SIDESbase effectively deals only with relatively small sets of candidate biomarkers, and its performance tends to deteriorate in 'massive' biomarker analysis problems with hundreds of baseline covariates. Performance loss is especially pronounced in settings where most of the candidate biomarkers are non-informative, which is quite common in real-life applications. SIDEScreen was specifically developed for the more challenging settings with large sets of pre-specified biomarkers. For application of the SIDES method to drug development programs, see [30, 96].

The SIDESbase subgroup search algorithm is outlined in Table VIII. To generate a large collection of *promising subgroups*, the SIDESbase algorithm starts with the overall population, which serves as the first parent group. The algorithm optimally splits the parent group into two complementary *child subgroups* for each candidate biomarker and selects the best *child group* based on a pre-specified *splitting criterion*. The procedure is then applied recursively to each child group, which is treated as a parent group.

Focusing first on continuous or ordinal biomarkers, an optimal cutoff c_i is chosen by examining all possible values of the biomarker that result in non-trivial biomarker-low and biomarker-high subgroups:

$$L_i(c_i) = \{X_i \leqslant c_i\} \text{ and } H_i(c_i) = \{X_i > c_i\},\ i = 1, \dots, p,$$

If X_i is a nominal biomarker with k levels, the optimal split of the parent group is found by optimizing the splitting criterion over all possible partitions of the k categories into two sets, which results in $2^{k-1} - 1$ non-trivial splits. For simplicity, it will be assumed from this point on that all biomarkers in the candidate set are continuous.

The individual steps of the SIDESbase subgroup search algorithm defined in Table VIII are described in detail in the succeeding discussion.

Table VIII. SIDESbase subgroup search algorithm.

Step 1. *Initialize*

A single 0-stage parent group includes all observations in the dataset. Initialize the set of promising subgroups as an empty set, $\mathcal{P} = \emptyset$.

Step 2. *Iterate* (splitting the current l-stage parent group, $0 \leqslant l \leqslant L$)

If $l = L$, the current parent group becomes terminal and is not considered for further splitting, otherwise:

1. Arrange the p candidate biomarkers from the 'best' to 'worst' in terms of the optimal value of the adjusted *splitting criterion*.
2. For each of the top M covariates, select two child subgroups based on the biomarker's 'best split' among all allowable splits. Let S_i denote the subgroup with the larger positive treatment effect based on the biomarker X_i.
3. Evaluate the *complexity criterion* on S_i and, if passed, include it in the set of *promising subgroups* $\mathcal{P}$.
4. For each promising subgroup S_i, set S_i as the current parent group, let $l = l + 1$ and go to Step 2.
5. If no biomarker has allowable splits resulting in a promising subgroup, the current parent group becomes terminal and is not considered for further splitting.

Step 3. *Finalize*

Include a promising subgroup from $\mathcal{P}$ in the final set if the unadjusted treatment effect p-value or the *multiplicity-adjusted* p-value is less than $p_{\max}$.

9.2. Step 2.1: splitting criterion

The splitting criterion is a function evaluated for each candidate biomarker at each allowable cutoff value. The most commonly used splitting criterion is the *differential splitting criterion*, which is given by

$$D(c) = 2\left[1 - \Phi\left(\frac{|T_H(c) - T_L(c)|}{\sqrt{2}}\right)\right], \tag{22}$$

where T_H and T_L are the appropriate test statistics for evaluating the treatment effect in the biomarker-high and biomarker-low child subgroups, respectively. A larger value of the test statistic indicates clinical improvement within a patient subgroup. Further, $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Note that the splitting criterion is measured on a 'p-value' scale and a smaller value of D indicates a stronger differential effect between the two child subgroups. The criterion (22) is derived as a one-sided p-value for the test statistic

$$t = \frac{|T_H(c) - T_L(c)|}{\sqrt{2}},$$

based on the half-normal reference distribution, $F(t) = 2\Phi(t) - 1, t \geqslant 0$.

One potential limitation of the differential splitting criterion is that it is computed based on the absolute treatment difference, and thus, it fails to discriminate between the splits associated with a large positive treatment effect in one of the subgroups and those associated with a large negative effect. Other types of splitting criteria are defined in [32]. For example, the *directional splitting criterion* helps address this limitation of the differential criterion by reducing the contribution of the child subgroup with a large negative treatment effect.

To arrange the candidate biomarkers, the splitting criterion $D(X, c)$ is evaluated exhaustively for all cutoffs c associated with the biomarker X. The best split is defined as follows:

$$c_i^* = \underset{c \in \mathcal{C}_i}{\operatorname{argmin}}\, D(X_i, c),$$

where $\mathcal{C}_i$ defines the *allowable set* of cutoffs for the biomarker X_i. This set includes all unique values of X_i, or a pre-specified grid of values can be used to facilitate the search. The set of allowable splits is further reduced by imposing appropriate sample size constrains, for example, the smallest acceptable number of patients in both child subgroups $\left(n_{\min}^{\text{both}}\right)$ and the smallest acceptable number of patients in the subgroup with the larger treatment effect $\left(n_{\min}^{\text{best}}\right)$.

The optimal value of the splitting criterion for X_i is denoted by

$$d_i = D\left(X_i, c_i^*\right).$$

As explained in Section 7, the optimal value of any splitting criterion found by an exhaustive search over the set of possible cutoffs is known to be biased in favor of biomarkers with a larger set of unique values or levels. As a result, an adjustment needs to be applied to support a 'fair' comparison of the candidate biomarkers. This 'local' multiplicity adjustment is based on the modified Šidák test (Appendix B of [32]). The adjusted values of the splitting criterion are denoted by

$$\tilde{d}_1, \dots, \tilde{d}_p.$$

The candidate biomarkers are arranged by the adjusted criterion.

9.3. Step 2.2: selection of the top subgroups

Let L_i^* and H_i^* denote the child subgroups obtained by an optimal splitting of the current parent group on the biomarker X_i. Let S_i denote the subgroup with the larger positive treatment effect, that is, $S_i = L_i^*$ if $T(L_j^*) > T(H_j^*)$ and $S_i = H_i^*$, otherwise. The subgroup S_i known as the *promising subgroup*.

To simplify the notation, assume that $\tilde{d}_1 > \cdots > \tilde{d}_p$, and thus, the biomarker X_1 is associated with the largest value of the adjusted splitting criterion. To streamline the subgroup search, the SIDESbase algorithm retains only the top M promising subgroups, that is, $S_1, \dots, S_M$, where M is a pre-specified algorithm parameter. As explained in [32], the main reason behind retaining multiple child subgroups for each parent group is that it ultimately improves the performance of the subgroup search. It is recognized in the machine-learning literature that a greedy approach that always chooses the best option out of multiple available options is likely to lead to unstable model selection. It will be shown in the succeeding discussion that the SIDEScreen procedure takes advantage of this feature of the SIDESbase algorithm by computing VI scores over the collection of subgroups that are used for identifying important biomarkers.

9.4. Step 2.3: complexity criterion

An important component of the SIDESbase subgroup search algorithm is complexity control. A complexity criterion is introduced to explicitly control the size of the search space (total number of subgroups examined by the algorithm). A promising subgroup is explored further only if the treatment effect in this subgroup is appreciably large compared with the effect in the parent group.

To apply the complexity criterion, let $p_1, \dots, p_M$ denote the one-sided treatment effect p-values in the promising subgroups $S_1, \dots, S_M$. The one-sided treatment effect p-value in the corresponding parent group is denoted by p_0. The complexity criterion is met in the subgroup S_i if $p_i \leqslant \gamma p_0$, where γ is a predefined complexity parameter with $0 < \gamma \leqslant 1$. If a promising subgroup meets the complexity criterion, the subgroup is added to the list of parent groups.

To understand the impact of the complexity parameter γ, note that, with a larger value of γ, the total number of subgroups generated by the SIDESbase algorithm is not controlled and is determined by parameters L and M (Step 2.4). By contrast, if γ is set to a small value, very few promising subgroups will be identified. An optimal value of the complexity parameter can be determined by CV [91].

9.5. Step 2.4: recursion

The SIDESbase subgroup search algorithm proceeds to the next stage and is applied recursively to the resulting list of parent groups defined at the end of Stage 1 until the algorithm reaches the maximum number of levels. This number is denoted by L and represents the number of times the algorithm can be recursively applied to parent groups starting with the overall population at Stage 0. The largest number of promising subgroups that can be generated by the algorithm is equal to $M + M^2 + \dots + M^L$.

9.6. Step 3: multiplicity adjustment

A multiplicity adjustment needs to be applied to each subgroup in the final set to remove selection bias and carry out reliable treatment effect tests. Unadjusted inferences within the promising subgroups are known to be highly unreliable. Appropriate adjustments of the treatment effect p-values help protect the probability of incorrectly discovering a patient subgroup with a large treatment difference under the assumption no treatment effect across all subsets of the overall patient population.

The SIDESbase procedure includes an important option to adjust the treatment effect p-values based on a permutation procedure [97]. This multiplicity adjustment relies on generating a large number of null datasets. These null datasets are constructed by permuting the treatment labels in the original dataset, and thus, the treatment effect tends to zero over the entire patient population.

First, the regular treatment effect p-values are computed in all subgroups identified by the SIDESbase procedure. These p-values are expected to be highly significant because the search algorithm pursued the subgroups with a strong treatment benefit. Let p_j^* denote the p-value in the subgroup S_j, where $j = 1, \dots, m$ and m is the total number of subgroups in the final set. In addition, the treatment effect p-value is computed in the best subgroup selected within each null dataset. This p-value is denoted by q_k, $k = 1, \dots, K$, and K is the number of null datasets. A multiplicity-adjusted p-value for the subgroup S_j is defined as the proportion of null datasets where the treatment difference in the best subgroup is more significant than the treatment difference within S_j. In other words, the adjusted treatment effect p-value in S_j is given by

$$\tilde{p}_j = \frac{1}{K} \sum_{k=1}^{K} I \left\{ q_k \leqslant p_j^* \right\}.$$

The resulting adjusted p-values provide the basis for reliable inferences within patient subgroups identified by SIDESbase and are generally much greater than the original treatment effect p-values.

It is shown in [32] that the complexity parameter γ helps control the multiplicity burden for SIDESbase. In particular, selecting a smaller value of γ reduces the size of the search space, and as a consequence, it reduces the degree of multiplicity adjustment.

9.7. SIDEScreen procedure

It was pointed out earlier in this section that the SIDESbase procedure defined in Table VIII tends to perform best in biomarker analysis problems with a relatively small number of candidate biomarkers. The SIDEScreen procedure serves as an extension of SIDESbase, which is designed to efficiently handle much larger sets of candidate biomarkers.

The SIDEScreen procedure is set up as a two-stage procedure, which applies the SIDESbase algorithm at the first stage without complexity control to generate a large collection of promising subgroups. A biomarker screen is introduced at the end of the first stage to filter out the biomarkers that are poor predictors of treatment response. The biomarker screen helps reduce the level of 'background noise' associated with the non-informative covariates. In the second stage, the SIDESbase algorithm is applied to the selected biomarkers with stronger predictive properties to arrive at the final set of patient subgroups.

The biomarker screen is applied by computing the VI score for each candidate biomarker. A biomarker's VI score is defined as the average value of the splitting criterion over all subgroups included in the final set. This approach takes advantage of the fact that only a small number of the candidate biomarkers demonstrate predictive ability and can be used for identifying treatment responders. More formally, consider the patient subgroups in the final set and denote them by $S_1, \dots, S_m$. The VI score for the biomarker X_i is defined as

$$\mathrm{VI}(X_i) = \frac{1}{m} \sum_{j=1}^{m} \lambda_{ij},\ i = 1, \dots, p.$$

Here, λ_{ij} quantifies the predictive ability of the biomarker X_i within the final subgroup S_j. Specifically, λ_{ij} is equal to the value of the adjusted splitting criterion for the best split in this subgroup (on the negative log scale), that is, $-\log(\tilde{d}_j)$, if X_i contributes to this subgroup and 0 otherwise.

Biomarker screens help considerably improve the performance of SIDEScreen compared to SIDESbase because VI scores help distinguish strong predictors of treatment response associated with higher values of the splitting criterion from noise covariates. Another important feature of VI scores is that they provide a comprehensive characterization of the predictive properties of a given biomarker. As indicated above, VI scores are computed by averaging contributions of a biomarker over the entire set of final subgroups. Even though the biomarker may not exhibit predictive properties at top levels of the subgroup search algorithm, it may turn out to be a stronger predictor in subgroups identified at deeper levels. This information will be taken into account when its VI score is computed.

Lipkovich and Dmitrienko [32] defined several biomarker screens that can be utilized within a two-stage subgroup search procedure. The more efficient approach to filtering out irrelevant covariates, termed the *adaptive biomarker screen*, relies on defining a data-driven threshold for the VI scores computed at the end of the first stage. The threshold is derived from the null distribution of the maximum VI score. Using a large number of null datasets, the maximum VI score ($\mathrm{VI}_{\max}$) is found over all candidate biomarkers within each null dataset. Let $\widehat{E}_0(\mathrm{VI}_{\max})$ and $\widehat{V}_0(\mathrm{VI}_{\max})$ denote the sample mean and variance of $\mathrm{VI}_{\max}$ under the null distribution. The adaptive biomarker screen retains the most important biomarkers that satisfy the following condition:

$$\mathrm{VI}(X) \geqslant \widehat{E}_0(\mathrm{VI}_{\max}) + c\sqrt{\widehat{V}_0(\mathrm{VI}_{\max})},$$

where c is a pre-defined constant. It is common to set c to 1 and the resulting biomarker screen rules is conceptually similar to the 'minCV+1SE' rule used in tree-based and penalized regression models (see Sections 5.3 and 6.2).

Finally, when performing a multiplicity adjustment to control the probability of incorrect subgroup discovery within the SIDEScreen procedure, it is critical to account for both stages of the algorithm used in this procedure. Multiplicity control is performed within the SIDEScreen approach by accounting for the subgroup assessment and biomarker screen in the first stage of the algorithm as well as the subgroup identification in the second stage. A permutation-based method used in Step 3 of SIDESbase is now applied to the entire two-stage subgroup search algorithm to compute the multiplicity-adjusted treatment effect p-values in the final set of subgroups selected after the second stage.

10. Application of subgroup discovery methods

This section illustrates the subgroup discovery methods defined in Sections 6 through 9 by applying them to the two case studies introduced in Section 3. It is important to note that the current implementation of these methods may be limited to a particular type of outcome variables. For example, the VT method is currently applicable only to continuous and binary outcomes. By contrast, the general SIDES method supports subgroup identification in trials with continuous, categorical, count-type, and time-to-event endpoints. For this reason, the subgroup discovery methods were applied to the two case studies as follows:

- Case study 1 (binary outcome): SIDES method (Sections 9.1 and 9.7), penalized regression methods (Section 6.2), VT method (Section 6.4), and OWL method for OTRs (Section 8.1).
- Case study 2 (survival outcome): SIDES method (Sections 9.1 and 9.7) and IT method (Section 7.1).

The subgroup discovery methods were implemented using open-source software, including R packages available on the Comprehensive R Archive Network (CRAN) web site:

```
http://cran.r-project.org
```

In addition, R programs kindly provided by the developers of the individual methods were used in this section. The code can be found on the Biopharmaceutical Network web site at

```
http://biopharmnet.com/subgroup-analysis/
```

10.1. Application of the SIDES method to Case study 1

The general SIDES method (SIDESbase and SIDEScreen procedures) defined in Sections 9.1 and 9.7 was applied to examine the dataset in Case study 1 to identify patient subgroups with enhanced efficacy. The SIDESbase and SIDEScreen procedures were implemented using the R package *RSIDES*.

The SIDESbase procedure was applied with the following parameters:

- Splitting criterion: differential criterion defined in Equation (22).
- Maximum number of promising child subgroups retained for each parent group (width): $M = 5$.
- Maximum number of levels (depth): $L = 2$.
- Maximum unadjusted treatment effect p-value (one-sided): $p_{\max} = 0.1$.
- Complexity parameter: $\gamma = 0.5$.
- Constraints on the subgroup sample size: smallest sample size in the any of the two child subgroups formed by a split: $n_{\min}^{\text{both}} = 30$; smallest sample size in the best of the two child subgroups: $n_{\min}^{\text{best}} = 60$.
- Number of permutations to compute multiplicity-adjusted p-values: $K = 10,000$.

The values of the algorithm parameters listed previously can be considered standard choices. For example, the depth parameter L was set to 2 to avoid hard-to-interpret subgroups. The constraints on the subgroup sample size were introduced to prevent erratic behavior of the test statistics, which is expected when smaller datasets are examined. Because $p_{\max} = 0.1$, the treatment effect within the subgroups was expected to be significant at a one-sided level of 0.1. Note that this restriction was applied to the final set of subgroups rather than to the promising subgroups at the intermediate stages of the SIDESbase algorithm. In general, the effect of this constraint on SIDESbase is rather cosmetic. However, $p_{\max}$ plays an important role in SIDEScreen because it affects the list of subgroups used for computing VI scores (subgroups with a non-significant treatment effect are excluded from the computation).

Like many methods based on recursive partitioning, SIDES can handle missing values in the dataset, in contrast with parametric regression approaches where all cases with at least one missing covariate should be deleted or missing values imputed. However, a rather simplistic approach was used in this particular case.

- For a nominal covariate X with missing values, an additional category 'missing' was created that captured all patients with missing values of X.
- For an ordinal/continuous covariate X with missing values, the splitting criterion was evaluated for all allowable splits on the set of patients with a non-missing X. Once an optimal split c was found, the child subgroups were formed as $L(c) = \{X \leqslant c \text{ and } X \text{ is not missing}\}$ and $H(c) = \{X > c \text{ and } X \text{ is not missing}\}$. Therefore, patients with missing values of X were not included in subgroups defined using this biomarker.

Table IX lists the four subgroups identified by SIDESbase and their characteristics, including the original and multiplicity-adjusted one-sided p-values. This table shows that the top subgroup identified by SIDESbase was defined using the biomarkers X_1 (patient's age) and X_2 (time from the first organ failure). The unadjusted treatment effect p-value in this subgroup was 0.0020; however, the treatment difference was no longer significant after an adjustment for selection bias was applied and the multiplicity-adjusted p-value equalled 0.3762. In other words, when the subgroup search algorithm was applied to 10,000 null datasets without a treatment effect, the p-value in the best subgroup was less than or equal to 0.0020 about 37.6% of the time. The non-significant multiplicity-adjusted p-value suggested that the apparent treatment effect in the top subgroup was most likely due to selection bias.

It is instructive to compare the basic subgroup search procedure (SIDESbase) to the more advanced procedure (SIDEScreen). SIDEScreen is a two-stage procedure which first evaluates the predictive ability of the candidate biomarkers (by computing the VI score for each biomarker) and then performs subgroup search based on the most relevant biomarkers that are associated with the highest variable (VI) importance scores. When computing the VI score in the first stage of the procedure, it is important to examine a broader search space. This is accomplished by disabling complexity control to generate the largest possible number of subgroups.

The SIDEScreen procedure was applied to the dataset in Case study 1 with the same options that were used in SIDESbase. However, the following options were modified:

- Maximum number of levels (depth): $L = 3$.
- Maximum unadjusted treatment effect p-value (one-sided): $p_{\max} = 1$.
- Complexity parameter: $\gamma = \infty$.

As pointed out earlier, complexity was not controlled in the first stage of the SIDEScreen procedure because γ was set to ∞.

A total of 64 subgroups was generated in the first stage of SIDEScreen based on the SIDESbase algorithm and the VI score were computed for each of the 11 candidate biomarkers. To define an efficient (adaptive) biomarker screen, the reference distribution of the VI scores was computed by running SIDESbase on 1000 null datasets. Figure 7 displays the VI scores along with the benchmarks derived from the reference distribution (the biomarkers are arranged by the VI score). The benchmark values were

Table IX. Overall population and patient subgroups identified using the SIDESbase procedure in Case study 1.

			Treatment effect p-value (one-sided)	
Subgroup	Size	Test statistic	Unadjusted p-value	Adjusted p-value
Overall population	470	−0.96	0.8301	NA
$\{X_2 \leqslant 31$ and $X_1 > 60\}$	123	2.89	0.0020	0.3762
$\{X_1 > 60$ and $X_2 > 2.2\}$	70	2.88	0.0020	0.3794
$\{X_2 \leqslant 31$ and $X_{10} > 0\}$	88	2.66	0.0039	0.4621
$\{X_3 > 244\}$	92	1.70	0.0453	0.6224

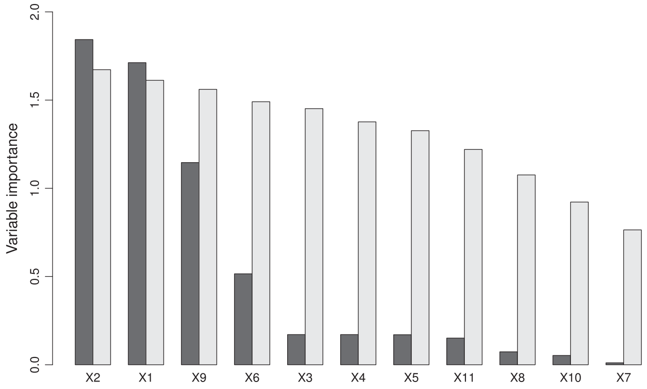

Figure 7. Variable importance scores (dark gray bars) and associated benchmarks (light gray bars) in the first stage of the SIDEScreen procedure in Case study 1.

Table X. Patient subgroup identified using the SIDEScreen procedure in Case study 1.

			Treatment effect p-value (one-sided)	
Subgroup	Size	Test statistic	Unadjusted p-value	Adjusted p-value
$\{X_2 \leqslant 31$ and $X_1 > 60\}$	123	2.89	0.0020	0.0444

computed using the following algorithm which follows algorithms used in resampling-based step-down multiple testing procedures (Westfall and Young [97]):

- Beginning with the top predictor of treatment response shown in Figure 7 (X_2), the benchmark value was given by
$$\widehat{E}_0(\mathrm{VI}_{\max}) + c\sqrt{\widehat{V}_0(\mathrm{VI}_{\max})},$$
where $c = 1$ and $\mathrm{VI}_{\max}$ was the maximum VI score over all 11 candidate covariates within each null set. Further, E_0 and V_0 were the sample mean and variance of $\mathrm{VI}_{\max}$ under the null distribution, respectively.
- For the next predictor (X_1), the benchmark was computed similarly, and the only exception was that the maximum VI score was found over the remaining 10 biomarkers (in other words, X_2 was excluded).
- For the third predictor (X_9), the benchmark was derived using the maximum VI score computed over the remaining nine biomarkers, and so on.

It follows from Figure 7 that the VI score was greater than the benchmark values for only two biomarkers (X_1 and X_2). This means that the adaptive biomarker screen retained X_1 and X_2. The other candidate biomarkers were dropped at the end of the first stage of SIDEScreen as non-informative.

The SIDESbase algorithm was applied to the biomarkers X_1 and X_2 in the second stage of SIDEScreen and identified a single subgroup, namely, $\{X_2 \leqslant 31$ and $X_1 > 60\}$. This subgroup is identical to the top subgroup listed in Table X. In order to obtain the multiplicity-adjusted p-value for the treatment effect test in this subgroup, additional 10,000 null datasets were generated from the original date set by randomly permuting the treatment labels and the two-stage procedure was applied to each dataset. Specifically, SIDESbase was applied with the same parameters that were used in the analysis of the original data, and the biomarkers were screened using the benchmarks displayed in Figure 7. The resulting multiplicity-adjusted p-value reflected the proportion of the null datasets where the SIDEScreen procedure selected a patient subgroup with a treatment effect p-value, which was as significant or more significant than the p-value in $\{X_2 \leqslant 31$ and $X_1 > 60\}$, that is, 0.0020. The adjusted p-value was equal to 0.0444 and is presented in Table X.

It is clear that the adjusted p-value in the subgroup $\{X_2 \leqslant 31$ and $X_1 > 60\}$ produced by the SIDEScreen procedure (Table X) is considerably smaller compared with the adjusted p-value based on the SIDES-base procedure (Table IX). This should come as no surprise since SIDEScreen is considerably more efficient (less greedy) than SIDESbase. The former utilizes a powerful biomarker screen that effectively shrinks the search space and reduces the multiplicity burden and, subsequently, results in a more efficient multiplicity adjustment.

10.2. Application of penalized regression methods to Case study 1

Penalized logistic regression models introduced in Section 6.2 were applied to Case study 1 to identify covariates that are predictive of treatment response and define subsets of the overall patient population with a positive treatment effect. The predictive properties of the candidate biomarkers were examined using the following two procedures:

- LASSO procedure: logistic regression modeling with the lasso penalty (implemented using the *glmnet* package); and
- FINDIT procedure: Logistic regression modeling based on the FindIt method (implemented using the *FindIt* package).

The binary outcome variable (28-day survival status) was re-coded as 0 (death) and 1 (survival) when the *glmnet* package was used and as −1 (death) and +1 (survival) when the *FindIt* package was used.

The feature space for regression modeling was defined using the same approach which was utilized in the artificial example presented in Section 6.2. The feature space included the treatment indicator, 11 candidate biomarkers, two-way biomarker interactions as well as interactions of the individual biomarkers and their two-way interactions with the treatment indicator. This resulted in a total of 155 variables in the model (note that the intercept was left 'unpenalized').

One significant limitation of penalized regression methods (and regression methods in general) is that no missing values are allowed in a dataset. As a result, the missing biomarker values needed to be imputed prior to applying these methods. A more principled approach would involve 'embedding' a subgroup identification procedure based on penalized regression within a multiple imputation framework. To achieve consistency across the global outcome modeling methods presented in this section (penalized regression and VT), missing covariate values were imputed in this case study using the same random forest-based method, which will be later applied in Section 10.3. This method was implemented using the native *rfImpute* function from the *randomForest* package.

Another important difference between tree-based subgroup identification procedures such as SIDES and regression-based procedures is that a biomarker with a highly skewed distribution and/or outlying values may exert undue influence on a regression model, which results in biased estimates of the model parameters. To ameliorate this problem, it is recommended to transform biomarkers with irregular distributions, for example, to apply a log transformation. By contrast, as emphasized in Section 5.3, tree-based approaches are invariant to monotone covariate transformations. In this particular case, a log transformation was applied to X_2 (time from the first organ failure to the start of study drug administration) before it was added to the feature space.

Beginning with the LASSO procedure, a 10-fold cross-validated binomial negative log-likelihood was used to select optimal values of the penalty parameter. The optimal values ($\lambda_{\min}$ and $\lambda_{\min 1se}$) were computed as in Section 6.2. The selection process is illustrated in Figure 8.

With a conservative penalty based on $\lambda_{\min 1se}$, none of the predictive variables, that is, treatment-by-biomarker interactions, were included in the final model. When the penalty parameter was set to $\lambda_{\min}$, several treatment-by-biomarker interaction terms were selected. Figure 9 shows the variables in the feature

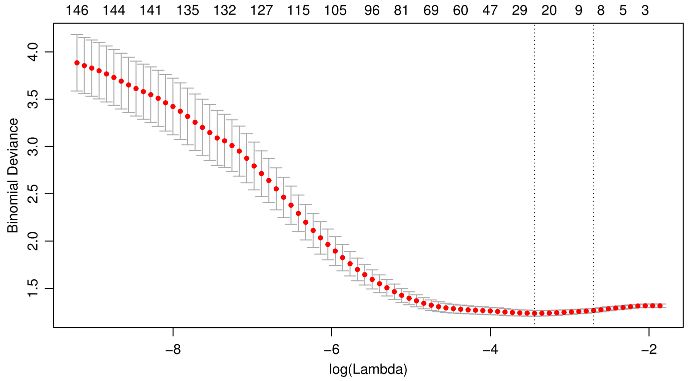

Figure 8. Ten-fold cross-validated negative log-likelihood as a function of the log-transformed penalty parameter λ in Case study 1. The vertical lines correspond to $\lambda_{\min}$ (left line) and $\lambda_{\min 1se}$ (right line). The error bars represent the standard errors. The values shown in the upper horizontal axis are the numbers of non-zero coefficients resulting when the penalty ($\log(\lambda)$) is set at values shown in the lower horizontal axis.

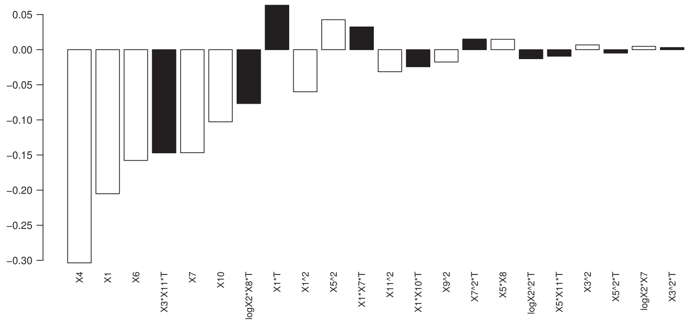

Figure 9. Coefficients estimated by the LASSO procedure (penalized logistic regression model with the penalty parameter set to $\lambda_{\min}$) in Case study 1 (T denotes the treatment variable).

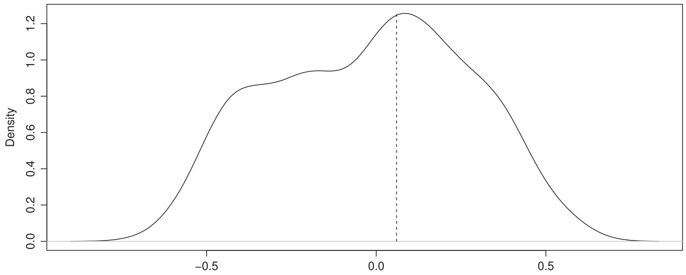

Figure 10. Distribution of the individual treatment contrasts computed using the random forest method in Case study 1.

space selected by the LASSO procedure when the penalty parameter was set to $\lambda_{\min}$. The predictive effects are represented by the black bars. We can see that the strongest predictors of treatment effect were the interaction between X_3 (baseline platelets) and X_{11} (baseline bilirubin), the interaction between $\log X_2$ (time from the first organ failure) and X_8 (GLASGOW score) and X_1 (patient's age). Recall that the SIDES-based procedures also identified X_1 and X_2 as the most promising predictors of a differential treatment effect in Section 10.1. Although, the interaction of the treatment indicator with $\log X_2$ was also selected by the LASSO procedure, it was mixed with another biomarker.

An application of the FINDIT procedure to Case study 1 revealed that no treatment-biomarker interactions were strong enough to be included in the final model. Interestingly, when the non-transformed version of X_2 was used, several interaction effects were selected and the results were very similar to those for the LASSO procedure with $\lambda_{\min}$ shown in Figure 9. This example underscores the sensitivity of regression-based subgroup identification methods to data anomalies.

10.3. Application of the VT method to Case study 1

This section presents an application of the VT method introduced in Section 6.4 to the problem of subgroup selection in Case study 1. The analysis dataset contained the binary outcome variable Y with values 1 (survival) and 0 (death). The set of covariates or inputs to be 'fed' into the random forest algorithm consisted of 11 continuous biomarkers $X_1, \ldots, X_{11}$, treatment indicator T with values 1 (treatment) and 0 (control) and two sets of treatment-by-biomarker interactions X_jT and $X_j(1-T), j = 1, \ldots, 11$. The latter were added based on the recommendation in [43] who found that precomputing treatment-by-covariate interactions appeared to facilitate the random forest algorithm. Because the current implementation of the random forest method in the R package *randomForest* assumes that all candidate biomarkers contain no missing values, the first step in the application of the VT method was to create a complete dataset. The imputation of the missing values was performed using the procedure utilized earlier in Section 10.2. This procedure relies on the Random Forest method to impute values in a way that is consistent with the 'model' fitted by the same method. After the missing biomarker values were imputed, the VT method was applied using the R code provided by one of the authors of this method (Dr. Jared Foster). The code is publicly available on the biopharmaceutical network web site.

Random forest was applied to the dataset in Case study 1 to compute the individual treatment contrast $z(\mathbf{x})$ for each patient. The contrast was defined as the difference between the predicted 28-day survival rates when the patient is assigned to the treatment and control arms. It is instructive to examine the distribution of the individual treatment contrasts, for example, to check whether any obvious clustering can be detected (e.g., clusters of super responders). In addition, it is helpful to plot the contrasts against important biomarkers to gain insight into their predictive properties. For illustration, Figure 10 displays the distribution of the individual treatment contrast in the severe sepsis dataset. The vertical line defines the minimal clinically important difference on an absolute scale, which was set to $\delta = 0.06$. This value is commonly used in severe sepsis trials (this absolute difference corresponds to a relative risk reduction of 20% if the 28-day survival rate in the placebo arm is 30%).

To perform a quick univariate assessment of the predictive abilities of the candidate biomarkers, Figure 11 displays plots of the individual treatment contrasts for two important biomarkers, X_1 (patient's age) and X_2 (time from the first organ failure). Note that, as in Section 10.2, a log transformation was applied to X_2 because there were multiple outliers in the dataset. It is generally helpful to study the relationship between the contrast and selected biomarker to determine if there is a simple way to select a cutoff to define a subgroup of treatment responders. A visual inspection of Figure 11 suggests that a

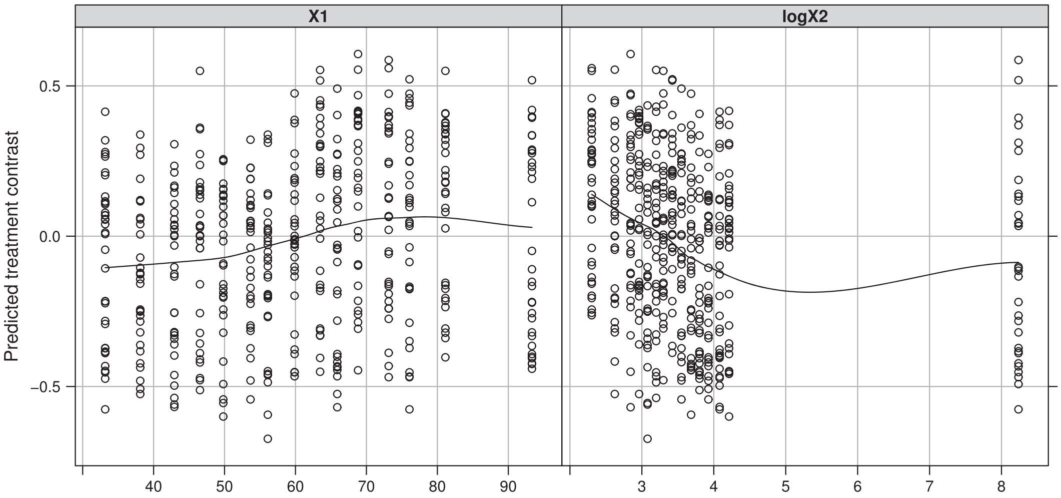

Figure 11. Individual treatment contrasts as functions of the biomarkers X_1 and $\log X_2$ in Case study 1.

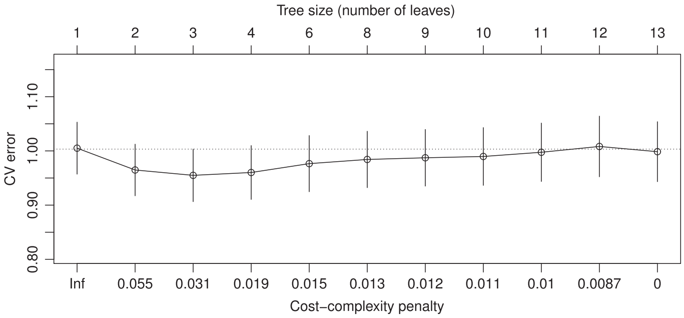

Figure 12. Cross-validation profile for the virtual twins regression trees fitted to the individual treatment contrasts in Case study 1.

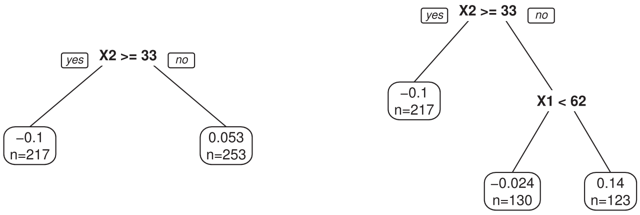

Figure 13. Virtual twins regression trees fitted to the individual treatment contrasts, pruned down to two terminal nodes (left-hand panel) and three terminal nodes (right-hand panel) in Case study 1. Patients who meet the splitting condition form the left branch, and those who do not form the right branch. The top value within each terminal node is the treatment difference in the 28-day survival rate, and the lower value is the subgroup size.

cutoff between 60 and 65 years for X_1 and a cutoff at about 3.5 for the log-transformed X_2 (corresponding to a cutoff at 33 h on the original scale) may be reasonable as they corresponded to the treatment contrast of 0. For example, patients who were older than 60–65 years of age tended to experience a beneficial treatment effect whereas patients in the complementary subgroup did not.

Continuing to the second stage of the VT method, CART models were fitted to the individual treatment contrasts. The CV profile in Figure 12 suggested that the optimal number of leaves in the VT tree was either 2 or 3, which corresponded to the cost-complexity penalty of 0.055 and 0.031, respectively.

Figure 13 displays the VT tree pruned down to two and three leaves. The figure presents the average value of the outcome variable, that is, the treatment difference in the 28-day survival rate predicted by random forest, and the number of patients for each subgroup. For example, within the subgroup

Table XI. Expected improvement in the 28-day survival rate in the subgroup $\{X_1 \geqslant 62$ and $X_2 < 33\}$ over the overall patient population computed using the VT method in Case study 1.

	Estimate of the treatment benefit		
General estimation approach	Re-substitution	Bootstrap bias-corrected	0.632 estimate
Model-based approach	0.154	0.134	0.141
Data-based approach	0.321	0.273	0.197

$\{X_1 \geqslant 62$ and $X_2 < 33\}$, the predicted treatment difference in the survival rate was 14%. Applying the pre-specified criterion with the clinically relevant difference $\delta = 0.06$ to the tree with two leaves returns no subgroup and for the tree with three leaves results in a patient subgroup based on the terminal node defined by splits on X_1 and X_2:

$$\widehat{S} = \{X_1 \geqslant 62 \text{ and } X_2 < 33\}.$$

This subgroup evaluated from the tree with three terminal nodes (leaves) is practically identical to the top subgroup identified by the SIDES method (Tables IX and X).

After the most promising subgroup with a clinically important beneficial effect was identified, the next step was to produce a reliable estimate of the treatment difference within this subgroup using the methodology presented in Section 6.4. The treatment benefits in $\widehat{S}$ were computed using the model-based (i.e., random forest-based) and data-based approaches. The corresponding estimates of $Q(\widehat{S})$ were first found using the naive re-substitution methods, which were then adjusted by the bootstrap bias-corrected procedures. Note that the VT method should be applied to each bootstrap sample exactly the same way as it is applied to the observed data. In particular, the selection of a subgroup based on a regression tree should proceed with no 'human intervention' based on a pre-defined rule. For example, a rule can be defined as follows: 'prune the tree using the value of the cost-complexity parameter (c) that minimizes the CV error and select terminal nodes with the treatment effect exceeding the pre-defined δ, if any'. This rule should be applied across all bootstrap samples. However, performing CV within each sample may be computationally prohibitive, and the method's authors suggested using a fixed pre-specified value of $c = 0.02$ that appeared to work well in simulations. With $c = 0.02$, a tree with four terminal nodes was selected in this example, and the same subgroup was identified as the subgroup based on the c suggested by CV. Therefore, we applied bootstrap correction with default value of $c = 0.02$.

The resulting estimates of the treatment benefit in the subgroup $\widehat{S}$ are shown in Table XI. The table lists the expected improvement in the 28-day survival rate in the identified subgroup over the overall patient population. As expected, Table XI shows that the re-substitution estimate of the treatment benefit was more conservative when the model-based approach to estimating of $Q(\widehat{S})$ was considered, compared with the data-based approach (15.4% versus 32.1%). The value of 32.1% is computed simply by subtracting the overall treatment difference in survival rates (−4.5%) from the treatment difference observed in the identified subgroup (27.6%). Note that this observed difference turned out to be much larger than the model-based difference of 14% estimated by the random forest algorithm (shown in Figure 13). Interestingly, while the data-based re-substitution estimate shrunk considerably when the basic bootstrap bias-correction procedure was applied (from 32.1% to 27.3%), the model-based estimate did not change much. When the bootstrap procedure based on the 0.632 estimator, which balances the re-substitution and 'out-of-bag' estimators, was applied, the model-based and data-based estimates appeared more consistent (14.1% and 19.7%, respectively).

10.4. Application of the outcome-weighted learning method for OTR to Case study 1

This section applies a simple OTR model for clinical trials with binary outcomes defined in Section 8.1 (Equation (20)) to Case study 1. Using the terminology introduced in [89], this approach corresponds to 'case-control weighting' because both cases and controls are utilized for estimating the optimal treatment assignment rule.

To estimate the coefficients of the penalized logistic model with the lasso penalty and patient-specific weights defined in (20), the *glmnet* package was utilized. The set of candidate predictors included the original 11 biomarkers plus 56 second-order covariate interactions and 11 squared covariates, resulting

Table XII. Penalized logistic regression models for optimal treatment regimes with the lasso penalty in Case study 1.

	Estimated regression coefficients based on $\lambda = \lambda_{\min}$	
Selected model term	Model 1	Model 2
Intercept	−0.1107	−0.0890
X_3X_{11}	−0.1627	0
X_1	0.1140	0.0690
X_1X_{10}	−0.0771	0
X_1X_7	0.0650	0
$\log X_2$	−0.0551	−0.0370
$X_{11}X_5$	−0.0517	0
X_9X_{11}	−0.0436	0
$X_{10} \log X_2$	−0.0340	0
$X_8 \log X_2$	−0.0236	0
X_7X_3	0.0078	0

in the total of 77 terms. As in other applications of penalized regression (Sections 6.2 and 10.2), 10-fold CV was used to choose an optimal penalty. The following models were considered:

- Model 1 included the full feature space described above as initial set of candidate predictors and estimated the coefficients with the lasso penalty parameter set to $\lambda_{\min}$ (minimum of cross-validated negative log-likelihood) resulting in 10 non-zero coefficients.
- Model 2 was a simpler model that excluded the interaction and quadratic terms from the feature space and the penalty parameter was also set to $\lambda_{\min}$ resulting in two non-zero coefficients besides the intercept.

The regression coefficients in Models 1 and 2 are listed in Table XII. The selected model terms are ordered by the absolute value of the regression coefficient in Model 1 (note that the coefficients correspond to the standardized values of the variables).

Based on the models defined in Table XII, patients should be allocated to the treatment ($t = 1$) if the linear combination based on the estimated regression coefficients is positive and to the control otherwise. Focusing on Model 1, patients of older age (based on X_1) with lower values of $\log X_2$ (log-transformed time from the first organ failure) should benefit from the experimental treatment. Interestingly, there were some important interaction effects of X_1 (age) with X_{10} (activity of daily living score) and X_7 (acute physiology and chronic health evaluation II score), which were negative. Given that higher values of age were associated with a beneficial treatment effect, we can conclude that the treatment effect may be further enhanced by considering patients with lower values of X_7 and X_{10}. The interpretation of the interaction effects is complicated by the fact that the lasso model does not obey the *hierarchy principle* which requires interactions must be considered only if the associated main effects are included in the model. For example, since the interaction term X_3X_{11} is included with a negative coefficient and none of the main effects is in the model, one can only conclude that patients would benefit from the experimental treatment if they score larger than average on X_3 (baseline platelets) and less than average on X_{11} (baseline bilirubin) or vice versa, which sounds a bit confusing. Forcing the regression coefficients for the main effects into the model would facilitate the interpretation of the results and could potentially result in eliminating some of the interaction terms. To facilitate the interpretability, one may consider recent modifications of the lasso method that would enforce hierarchy, see, for example, [98, 99]. These methods are implemented in the R packages *hierNet* and *glinternet*, respectively.

It is instructive to compare the two models presented in Table XII with the results of global outcome modeling based on the lasso method reported in Figure 9. The results are generally comparable in that the most predictive effects in Figure 9, that is, the interaction terms that include the treatment indicator and covariates X_1, $\log(X_2)$, X_{11}, X_3 also appear in Table XII with the same signs and a comparable order of magnitude.

Considering Model 2 in Table XII, the same main effects that were selected in Model 1 were also included in this model, that is, X_1 and $\log X_2$. The same two biomarkers were identified as strong predictors of a beneficial treatment effect using the SIDES and VT methods in Sections 10.1 and 10.3. The signs of the regression coefficients for X_1 and $\log X_2$ indicate that older patients with a shorter time from the first organ failure to the start of drug administration should allocated to the treatment. Recall that

Table XIII. Bootstrap-based confidence intervals for the expected treatment effects based on the optimal treatment assignment rules for Models 1 and 2 in Case study 1.

Model	95% CI for $d_-(\widehat{\boldsymbol{\beta}})$	95% CI for $d_+(\widehat{\boldsymbol{\beta}})$
Model 1	[−0.001, 0.558]	[0.037,1.00]
Model 2	[−0.004, 0.355]	[0.025, 0.73]

the subgroups of patients who are likely to experience an improvement suggested by the SIDES and VT methods were

$$\{X_2 \leqslant 31 \text{ and } X_1 > 60\} \text{ and } \{X_2 < 33 \text{ and } X_1 \geqslant 62\},$$

respectively. By comparison, after converting X_1 and $\log X_2$ to the original units, the following simple treatment assignment rule was derived based on Model 2:

- Assign a patient to the experimental treatment if $X_1 - 6.6 \log X_2 - 56.7 \geqslant 0$.
- Assign a patient to the control if $X_1 - 6.6 \log X_2 - 56.7 < 0$.

Further, to assess whether the treatment assignment rule based on Model 1 provides a substantial improvement over to a simpler rule based on Model 2, 95% bootstrap confidence intervals were computed for the average treatment effects $d_+(\widehat{\boldsymbol{\beta}})$ and $d_-(\widehat{\boldsymbol{\beta}})$ that were defined in Equation (21). Given that the outcome variable is binary in this case study, these quantities were measured on a probability scale. In particular, $d_+(\widehat{\boldsymbol{\beta}})$ was the treatment contrast in the subgroup of patients assigned to the treatment arm, that is, patients with $z(\mathbf{x}|\widehat{\boldsymbol{\beta}}) \geqslant 0$, with larger values indicating a beneficial effect of the experimental treatment. Similarly, $d_-(\widehat{\boldsymbol{\beta}})$ was the treatment contrast in the control arm, that is, $z(\mathbf{x}|\widehat{\boldsymbol{\beta}}) < 0$, with larger values indicating a beneficial effect of the control treatment.

The confidence intervals for the average treatment effects are shown in Table XIII. It follows from this table that the treatment assignment rules based on Models 1 and 2 appeared beneficial for the patients assigned to the experimental treatment because the 95% confidence intervals for $d_+(\widehat{\boldsymbol{\beta}})$ excluded zero. On the other hand, the benefits of assigning patients to the control arm were not obvious because the lower limits of the 95% confidence intervals for $d_-(\widehat{\boldsymbol{\beta}})$ were negative. This finding, therefore, may be especially useful for the trial's sponsor. The second observation is that because the conclusions from the two models are qualitatively similar, the advantage of a more complex treatment assignment rule based on Model 1 does not appear strong enough to justify including the interaction terms in the model.

10.5. Application of the SIDES method to Case study 2

This section and Section 10.6 provide illustrations of subgroup identification methods in the context of Case study 2 with a time-to-event outcome. An important feature of Case study 2 is that the treatment effect in the overall population was positive and marginally significant with the hazard ratio of 0.85 (a lower value of the hazard ratio indicates a beneficial effect). The associated one-sided log-rank p-value was 0.0367.

We will begin with the SIDES method and apply the two-stage SIDEScreen procedure with a biomarker screen to the problem of exploring subgroups of patients with enhanced treatment effect based on the 14 biomarkers listed in Table II. The first stage of the procedure was based on SIDESbase with the parameters that were identical to the parameters used in Section 10.1.

Figure 14 plots the VI scores of the candidate biomarkers. As in Figure 7, the biomarker-specific benchmark values estimated from the null distribution helped select the covariates with strong predictive properties. The VI scores exceeded the benchmarks for X_3 (cytogenetic category) and X_{13} (IPSS-R). These covariates passed the biomarker screen and were selected for the second stage of the SIDEScreen procedure.

Because the overall treatment effect was quite large, it was sensible to focus on subgroups with the treatment effect that was greater than that in the overall population. In order to carry over the overall treatment effect to the reference (null) sets, the sets were constructed by permuting the covariate columns rather than the treatment labels. Recall that permuting the treatment labels creates a homogeneous treatment effect across all subsets of the overall population, which is not relevant in this setting given that all subgroups would then inherit the overall treatment effect. Multiplicity-adjusted p-values in the subgroups

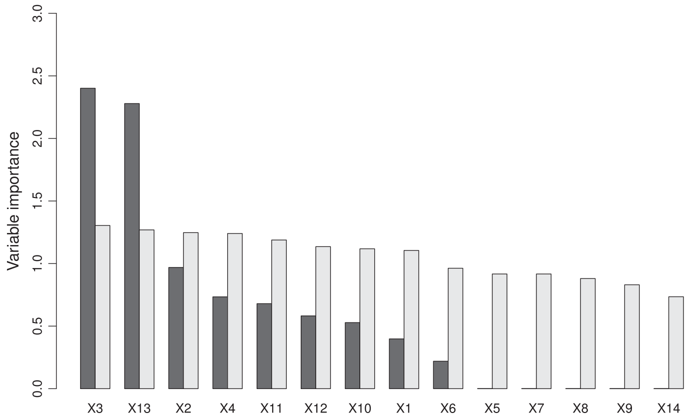

Figure 14. Variable importance scores (dark gray bars) and associated benchmarks (light gray bars) in the first stage of the SIDEScreen procedure in Case study 2.

Table XIV. Overall population and patient subgroups identified using the SIDEScreen procedure in Case study 2.

		Test	Hazard	Treatment effect p-value (one-sided)	
Subgroup	Size	statistic	ratio	Unadjusted p-value	Adjusted p-value
Overall population	599	1.79	0.849	0.0367	NA
$\{X_{13} > 3$ and $X_3 \leqslant 4\}$	220	4.46	0.513	0.000004	0.001
$\{X_{13} > 3\}$	329	4.31	0.596	0.000008	0.004
$\{X_3 > 2\}$	312	4.04	0.614	0.00003	0.010
$\{X_3 > 2$ and $X_{13} > 3\}$	272	3.11	0.666	0.0009	0.072

of interest were computed using $K = 1000$ null datasets where all covariate columns were randomly permuted. As a result, both prognostic and predictive covariate effects were removed; however, the overall treatment effect and correlations among the covariates were retained in each null set.

The results are summarized in Table XIV. The top subgroup identified by the SIDEScreen procedure was $\{X_{13} > 3$ and $X_3 \geqslant 4\}$, which corresponded to a very high IPSS-R score and poor or better cytogenetic category (Table II). It is important to note that another subgroup was also based on these two variables, namely, $\{X_3 > 2$ and $X_{13} > 3\}$. It may at first appear counterintuitive that patients with a larger value of X_3 experienced a beneficial treatment effect. However, the subpopulation defined by these cutoffs was driven by a large treatment effect in patients with $X_3 = 3$ (intermediate) and $X_3 = 4$ (poor). The overlap between the above groups, as measured by the Jaccard similarity index (intersection-to-union ratio) was about 50%.

10.6. Application of the IT method to Case study 2

To illustrate the IT procedure, the three-step algorithm defined in Section 7.1 was applied to Case study 2. Note that the IT procedure currently does not allow missing values in the candidate covariates, and therefore, eight records with unknown levels of X_{13} had to be deleted from the dataset.

The initial tree was grown using the *grow.INT* function from the suite of R functions kindly provided by Dr. Xiaogang Su and available on the biopharmaceutical network web site. The function was called with the following parameters:

- `min.ndsz`=20 (minimum number of observations for claiming a terminal node);
- `n0`=10 (minimum number of observations in each treatment arm within either of the two child groups when splitting a parent node); and
- `max.depth`=15 (the maximum depth in the subgroup identification algorithm). Note that this parameter corresponds to the tree height in [71].

Setting `max.depth` to 15 may appear as an overkill; however, note that the idea here is just to make sure that the only constraining factor for growing the initial large tree $\mathcal{T}_0$ will be the minimal size of the terminal node. Recall that the final tree will be selected by pruning the initial tree using a principled statistical procedure, which takes model complexity into account, and there is no need to impose any arbitrary pruning on the initial tree by limiting its depth. Also, the actual size of $\mathcal{T}_0$ will be limited by `min.ndsz` and `n0` and might never reach the maximum depth. As shown below, none of the trees grown on multiple bootstrap samples had more than 13 leaves.

The best-sized tree was selected from the pruned sequence by maximizing the criterion (13) with $\alpha_c = \log(n)$, and the bias-corrected estimates of $G(\mathcal{T}_i)$, $i = 0, \ldots, M$, were obtained using $B = 30$ bootstrap samples. The optimal tree had four leaves. For illustration, Figure 15 presents a family of bias-adjusted criteria corresponding to $\alpha_c = 2, 3, 4, \log(n)$ applied to each tree in the sequence.

The optimal tree shown in Figure 16 suggested two promising subgroups with enhanced treatment effect:

- The first subgroup was associated with the terminal node $\{X_{13} \geqslant 3.5$ and $X_3 < 4.5\}$. Note that, when a tree is formed by splitting on *ordinal* biomarkers, the cutoffs may be selected between the actual levels of a biomarker, which may lead to confusion. A more natural definition of this subgroup would be $\{X_{13} > 3$ and $X_3 \leqslant 4\}$. This is a large subgroup with $n = 220$ patients and a strong beneficial effect (hazard ratio of 0.51). This subgroup was identical to the top subgroup identified by the SIDEScreen procedure (Table XIV).
- The second promising subgroup was based on three biomarkers. It was comprised of male patients who had poor values of both Cytogenetic and IPSS-R scores: $\{X_{13} > 3$ and $X_3 > 4$ and $X_1 = 1\}$. This is a rather small subgroup with $n = 59$ patients and its usefulness and generalizability may be questionable.

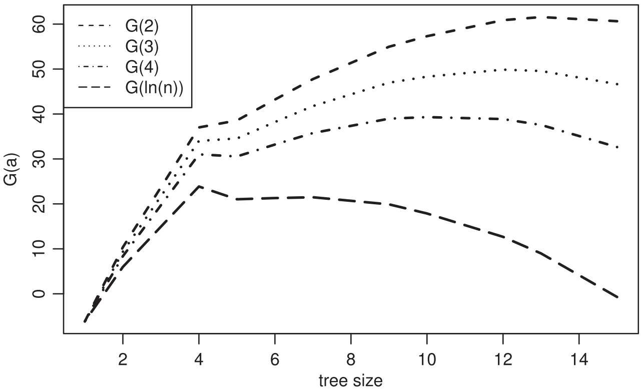

Figure 15. Selection of the best-sized tree using several bias-adjusted interaction-complexity criteria by the interaction trees procedure in Case study 2.

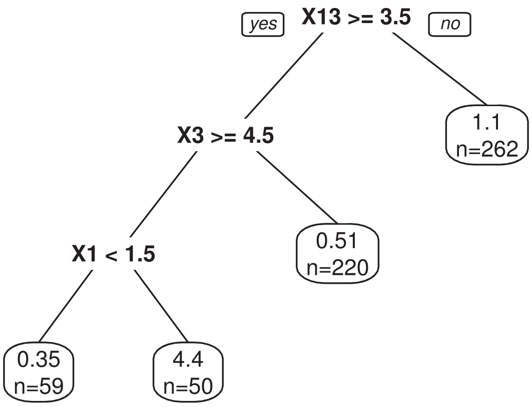

Figure 16. The best-sized tree selected by the interaction trees procedure in Case study 2. The hazard ratio (HR) for the experimental treatment versus control (HR< 1 indicates a beneficial treatment effect) and sample sizes are displayed within each terminal node.

11. Summary

We conclude the review of biomarker evaluation and subgroup identification methods in late-stage clinical trials with some general thoughts.

First of all, it is important to realize that popular approaches to subgroup identification and analysis come from such diverse fields of research as machine/statistical learning, multiple testing, and causal inference, and are necessarily linked with specific background and terminology that are common in these areas. Sometimes seemingly different methods developed by different groups of authors turn out to be almost equivalent to each other.

11.1. General comparison of subgroup identification methods

To provide a general comparison of different classes of subgroup identification methods, given the amount of uncertainty and lack of knowledge about subpopulations of patients who may experience enhanced treatment effect, non-parametric methods (e.g., methods based on recursive partitioning) appear more flexible and efficient compared with parametric approaches in that they support subgroup exploration within a very broad 'model space'. Furthermore, unlike standard recursive partitioning methods (e.g., CART) that aim at identifying subgroups with heterogeneous outcome values, partitioning methods for personalized medicine rely on a variety of splitting criteria that are modified appropriately to focus on subgroups with a differential treatment effect. This is typically achieved by incorporating treatment-by-splitting covariate interaction effects.

11.2. Multiplicity adjustment and complexity control

There is increasing demand for multiplicity adjustments in subgroup identification strategies in clinical trials, which in the past was seen as a rather unusual or unnecessary feature in data-mining/machine-learning applications. While strict multiplicity control may be virtually impossible to achieve (in the sense of strong familywise error rate control [22]) owing to the complexity of the model space and difficulty of enumerating all null hypotheses, weak control of the probability of incorrect subgroup selection associated with a subgroup identification strategy can be implemented based on resampling methods.

Complexity control is another and perhaps even more important principle that applies to subgroup identification as much as to any exercise of model selection (recall that we consider biomarker/subgroup selection as a special case of model selection). The fundamental idea behind complexity control in applications of machine learning is achieving a reasonable trade-off between bias and variance, and the same principle applies to subgroup identification. Complexity control is also related to the reproducibility principle, because lack of constraint on the search space typically results in selecting patient subgroups based on noise/non-informative biomarkers and, as a result, these subgroups are unlikely to be replicated with the future data. As part of complexity control, it is important to account for the fact that some biomarkers are more likely to be chosen by chance because the sets of unique values vary across the candidate biomarkers, which may lead to selection bias. Therefore applying a penalty that 'equalizes' the probability of selection by chance across the biomarkers should be a part of complexity control strategy as well as multiplicity control.

More generally, multiplicity adjustments ought to be used in combination with controlling the complexity of the subgroup selection process. Performing an unconstrained search for subgroups followed by a multiplicity adjustment may be an inefficient strategy because

- It may result in identifying patient subgroups that have a low chance of being replicated in an independent dataset.
- The resulting multiplicity adjustment may be too conservative, which will lead to very large multiplicity-adjusted treatment effect p-values within the selected subgroups.

As an example, performing a greedy search for subgroups by brute force, that is, by a complete enumeration of all possible subgroups that can be formed by, say, up to three biomarkers, is likely to generate spurious subgroups with highly significant treatment effect p-values. However, the probability of observing a similar significant treatment effect within these subgroups in another study will be low. Replicating the entire strategy on the reference (null) data is likely to also generate subgroups with highly significant p-values. Therefore, resampling-based multiplicity-adjusted p-values (i.e., the proportion of null sets with p-values as small as or smaller than the observed p-value) would be relatively large. The approach described previously needs to be contrasted with less greedy strategies that put an appropriate 'constraint jacket' on the model space by balancing bias and variance (e.g., methods of penalized regression) resulting in less complex subgroups based on fewer biomarkers. The reduced model space results in a lower multiplicity burden and therefore a smaller multiplicity penalty when computing multiplicity-adjusted p-values. As a consequence, a modestly significant observed p-value associated with a subgroup based on constrained subgroup search is likely to translate into a much smaller adjusted p-value compared with that obtained after unconstrained search.

Several approaches have been proposed recently to avoid 'greediness' and overfitting in subgroup search:

- frequentist methods employing complexity penalties, typically determined by resampling-based methods, for example, methods based on penalized regression [27, 38, 79] and tree-based methods [71, 75];
- ensemble learning methods that average over a large number of 'learners' to shrink the contribution of noise covariates to zero [43, 94, 100];
- shrinkage and model averaging via Bayesian methods [25, 47, 77]; and
- methods that use 'indirect' or less direct criteria for variable/subgroup selection that avoid exhaustive search for subgroups with desired features [72].

11.3. Bias-corrected treatment effect estimates

One of the most challenging tasks in subgroup identification is obtaining unbiased and reliable estimates of treatment effects in the selected patient subgroups, known as 'honest' estimates. Note that the estimated effects are typically used in designing future conformation studies, often as part of seamless Phase II/Phase III development programs. Incorrectly estimated effect sizes may lead to wrong decisions resulting in wasted time and money and/or lost opportunities both for the sponsor and society in general. Obtaining such estimates normally requires additional independent (or test) data. When no test datasets are available, resampling methods (bootstrap or CV) can be applied. When resampling data have been used for tuning a method's complexity parameters, the same data cannot be re-used to compute 'honest' estimates of treatment effect. In such cases, researchers may resort to double bootstrap or double CV. As a general principle, when using resampling methods for computing bias-corrected subgroup effects, it is important that the *entire* search strategy (including estimation of any data-driven tuning parameters) be implemented *afresh* on each dataset. As is the case with any method of predictive learning, accurate predictions (here, individual and subgroup-specific treatment contrast), rather than 'enforcing' strict Type I error rate control is the key objective. Different measures of performance or 'expected benefit' can be defined for a given subgroup or predictive biomarker, based on the ultimate goals of subgroup identification:

- expected treatment effect in a specific subgroup and its excess over that in the overall population;
- utility function evaluated on a subgroup that takes into account the 'treatment burden' based on safety and/or extra costs that may also reflect the minimal clinically meaningful treatment effect in the subgroup;
- power or predictive power of a future trial where the identified subgroup will be used as part of a tailoring strategy, for example, the trial may utilize an enrichment design based on this patient subgroup; and
- value function of the optimal treatment assignment rule based on the identified biomarkers/subgroups compared with a rule that assigns all patients to the same treatment.

11.4. Missing data

Another important issue that arises in subgroup identification, as much as in any analysis of clinical data, is proper handling of missing data. Missing values can arise both in the set of covariates and outcomes (the latter typically due to loss to follow up). Most methods presented in this tutorial do not explicitly handle missing outcomes and typically assume the last observed value, which is prone to selection bias unless missingness is completely at random.

However, some general methods such as inverse probability (of censoring) weighting or multiple imputation can be used in conjunction with many of the proposed methods. This requires additional modeling steps to be completed prior to data analysis and brings additional challenges. For example, it is not clear how to integrate the results of subgroup analysis across multiply imputed datasets.

Handling missing covariates is a more challenging issue for subgroup identification methods given that they thrive on covariate information. Methods based on parametric modeling would have to dispense with the entire patient record as long as a single missing biomarker is present, unless imputation techniques are used to complete the biomarker profile. Some methods may have 'built-in' imputation strategies such as the VT method [43] that imputes missing data using a generic imputation method of random forest. Tree-based methods are less affected by missing covariates in that they do not require deletion of incomplete profiles. However, their performance may be severely affected by simply ignoring missing values when evaluating candidate splits. Some tree-based methods, including the methods based on the GUIDE platform [72]), use missingness as a distinct category (that the tree algorithm can split on), which may be a better alternative to ignoring missing observations.

More research is warranted to better understand the impact of different missingness mechanisms and develop principled methods for dealing with missing data in the context of subgroup identification. Sensitivity analysis that can be implemented using imputation procedures enhanced with sensitivity parameters quantifying possible departures from misssingness at random can be useful to evaluate the potential impact of missing data on the selected subgroups and treatment effect estimates within these subgroups.

11.5. Key features of subgroup identification methods

To help the reader navigate through the thick forest of emerging subgroup identification methods, we propose a check list of several important features that should be examined for any prospective method. These features are defined as follows:

1. modeling type: Freq (Frequentist), Bayes (Bayesian); P (parametric), SP (semiparametric), NP (nonparametric);
2. dimensionality of the covariate space: low, medium, high;
3. results produced by the method: B (selected biomarkers or biomarker ranking based on VI scores that can be used for tailoring), P (predictive scores for individual treatment effects), T (optimal treatment assignment), S (identified subgroups);
4. evaluation of the Type I error rate/false discovery rate for the entire subgroup search strategy: yes, no;
5. application of complexity control to prevent data overfitting: yes, no;
6. control (reduction) of selection bias when evaluating candidate subgroups: yes, no;
7. Availability of 'honest' estimates of treatment effects in identified subgroups: yes, no; and
8. availability of software implementation: C (R package available on the CRAN web site), B (R code available on the biopharmaceutical network web site), P (proprietary).

Note that most methods are associated with a combination of letters when the third category is considered, for example, 'B, P' means that the method selects predictive biomarkers and constructs predictive score. Also, the descriptor 'B, S' may appear redundant since the knowledge of subgroups (S) implies knowledge of predictive biomarkers (B); however, some methods may report a broader set of selected biomarkers including 'overlapping' variables that may be 'eliminated' at the subgroup construction stage.

Table XV provides information on these features for several commonly used subgroup identification methods, most of which were considered in this tutorial. The methods are broken into four classes based on the taxonomy of subgroup identification methods introduced in Section 4.3. This classification of available methods provides some insight as to the situations when different methods may be particularly applicable. For example, methods that evaluate optimal regimes are useful in large Phase III or IV trials that compare several active treatments in a diverse population. Methods that utilize penalized regression and ensemble learning can handle very large sets of candidate covariates. As a consequence, these methods can be used in settings where the sample size is rather small, including early-stage trials, and the main focus is on selecting biomarkers rather than specific patient subgroups that can be utilized in subsequent Phase III trials. Tree-based methods are useful when there are a few candidate biomarkers, for example, 15–20 biomarkers, in relatively large datasets (say, with 1000–2000 patients) and subgroups can be reliably estimated. Evaluation of biomarkers using Bayesian shrinkage regression models such as models studied in [77] is well suited to evaluating post-hoc hypotheses or meta-analysis with a relatively small number of subgroups defined by units where the exchangeability assumption is reasonable. Examples include studies that focus on the effect of multiple countries or demographic groups.

Although the methods presented in this table were developed mainly for clinical trial settings with random treatment assignment, some of these methods are easily extended to observational studies where

Table XV. Key features of commonly used subgroup identification methods.

Method	Modeling type (1)	Dimen-sionality (2)	Biomarker selection (3)	Control of false positive rate (4)	Complexity control (5)	Selection control (6)	Honest estimate of treatment effect (7)	Software implemen-tation (8)
		Global outcome modeling						
Virtual twins (Foster *et al*. [43])	Freq/NP	High	P, S	No	Yes	No	Yes	B
Cai *et al*. [39]	Freq/NP	Low	P	Yes	No	No	Yes	
FindIt (Imai and Ratkovic [38])	Freq/P	High		No	Yes	No	No	C
STIMA (Dusseldorp *et al*. [45]	Freq/NP	Medium	S	No	Yes	No	No	C
Bayesian approaches (Dixon and Simon [46]; Hodges *et al*. [47])	Bayes/P	Low	P	No	Yes	No	Yes	
		Global treatment effect modeling						
Interaction Trees (Su *et al*. [71]; Negassa *et al*. [69])	Freq/NP	High	S	No	Yes	No	No	B
Gi as part of GUIDE (Loh *et al*. [72])	Freq/NP	Medium	S	No	Yes	Yes	Yes	C
Modified covariate method (Tian *et al*. [76])	Freq/P	High	P	No	Yes	No	No	
QUINT (Dusseldorp and Mechelen [75])	Freq/NP	Medium	S	No	Yes	No	No	C
		Optimal treatment regimes						
Biomarker selector (Gunter *et al*. [21]	Freq/P	High	B	Yes	Yes	No	No	
Qian and Murphy [79]) Freq/P	High	P,T	No	Yes	No	No		
Zhao *et al*. [83], *Xu et al*. [27]	Freq/P	High	P, T	No	Yes	No	No	B
Zhang *et al*. [84]	Freq/SP	High	T	No	Yes	No	No	
		Local modeling						
Adaptation of PRIM (Chen *et al*. [93]; Kehl and Ulm [90])	Freq/NP	High	S	No	Yes	No	No	P
SIDES (Lipkovich *et al*. [91]) and SIDEScreen (Lipkovich *et al*. [94])	Freq/NP	Medium	B, S	Yes	Yes	Yes	Yes	B
Berger *et al*. [25], Sivaganesan *et al*. [95]	Bayes/NP	Medium	S	Yes	Yes	No	Yes	P

the treatment choice is determined by a prescribing physician and may be driven by patient-level characteristics. For example, within the outcome weighted learning framework, the probability of treatment assignment is explicitly incorporated in the outcome-based weights, and in the context of observational data, this probability can be estimated using propensity score modeling from the available data.

We hope that the feature list would prove useful and applicable not only for the methods considered in this tutorial but also to those that were not covered as well as new methods that will be developed in the future.

11.6. Operating characteristics of subgroup identification methods

While we did not attempt to evaluate operating characteristics of available subgroup identification and biomarker evaluation methods that should require a comprehensive simulation study, we feel it will be useful to list key criteria or operating characteristics that can be used for evaluating the performance of these methods. The ideas presented in the succeeding text are based on those found in the literature on subgroup identification (for example, [27,43,72,82,94]) and our own experience in this area. The criteria can be divided into several domains:

(1) *Biomarker/subgroup level*. How well are the predictive biomarkers and specific signatures, that is, biomarkers and associated cutoffs, identified? For example, how often are the true predictive biomarkers/signatures selected (power) and how often are irrelevant biomarkers incorrectly identified as predictive biomarkers (Type I error rate)? For the latter, it is important to evaluate the presence and impact of selection bias, that is, whether or not the probability of selecting noise variables with different sets of candidate splits is the same.
(2) *Subject level*. How closely are the patients included in the 'true' subgroup approximated with the identified subgroup? Relevant performance measures include sensitivity (percent of patients in the true subgroup among those included in the selected subgroup) or specificity (percent of patients not in the true subgroup among those not included in the selected subgroup).
(3) *Treatment effect level*. What is the excess of the treatment effect in the selected subgroup compared with the overall population effect (for example, the performance measures proposed in [94]).
(4) In the context of evaluating an *optimal treatment policy/regime*, what is the expected treatment effect resulting from the treatment assignment based on the estimated individual treatment regime [27]? The agreement between the true and estimated regimes can be evaluated as the percent of patients with the correct treatment decision if the treatment assignment is based on the estimated treatment regime.

When comparing characteristics of different subgroup identification and biomarker evaluation methods, it is critical to consider the general features (1–8) summarized in Table XV as well as the operating characteristics listed previously in a consistent way. A discussion of approaches to a comprehensive comparison of operating characteristics of several applicable methods using the same battery of real and simulated datasets is an important topic for future research.

Acknowledgements

We would like to thank Dr. Stephen Ruberg and Dr. Gregory Enas for numerous discussions and Dr. Jared Foster and Dr. Bohdana Ratitch for valuable comments. In addition, we would like to thank two anonymous reviewers, and we are grateful to Michael Dmitrienko for editorial assistance.

References

1. Food and Drug Administration. Guidance for industry: enrichment strategies for clinical trials to support approval of human drugs and biological products, 2012.
2. European Medicines Agency. Guideline on the investigation of subgroups in confirmatory clinical trials, 2014.
3. Dmitrienko A, Muysers C, Fritsch A, Lipkovich I. General guidance on exploratory and confirmatory subgroup analysis in late-stage clinical trials. *Journal of Biopharmaceutical Statistics* 2016; **26**:71–98.
4. Varadhan R, Segal J B, Boyd C M, Wu AW, Weiss CO. A framework for the analysis of heterogeneity of treatment effect in patient-centered outcomes research. *Journal of Clinical Epidemiology* 2013; **66**:818–825.
5. Mayer C, Lipkovich I, Dmitrienko A. Survey results on industry practices and challenges in subgroup analysis in clinical trials. *Statistics in Biopharmaceutical Research* 2015; **7**:272–282.
6. Freidlin B, Jiang W, Simon R. Adaptive signature design: the cross-validated adaptive signature design. *Clinical Cancer Research* 2010; **16**:691–698.

7. Simon N, Simon R. Adaptive enrichment designs for clinical trials. *Biostatistics* 2013; **14**:513–625.
8. Xu Y, Trippa L, Müller P, Ji Y. Subgroup-based adaptive. (SUBA) designs for multi-arm biomarker trials. *Statistics in Biosciences* 2016; **8**(1):159–180.
9. Gu X, Chen N, Wei C, Liu S, Papadimitrakopoulou VA, Herbst RS, Lee JJ. Bayesian two-stage biomarker-based adaptive design for targeted therapy development. *Statistics in Biosciences* 2014:1–30.
10. Zhao Y, Kosorok MR, Zeng D. Reinforcement learning design for cancer clinical trials. *Statistics in Medicine* 2009; **28**:3294–3315.
11. Schulte PJ, Tsiatis AA, Laber EB, Davidian M. Q- and A-learning methods for estimating optimal dynamic treatment regimes. *Statistical Science* 2014; **29**:640–661.
12. Brookes ST, Whitley E, Peters TJ, Mulheran PA, Egger M, Davey Smith G. Subgroup analyses in randomised controlled trials: quantifying the risks of false-positives and false-negatives. *Health Technology Assessment* 2001; **5**:1–56.
13. Rothwell PM. Subgroup analysis in randomized controlled trials: importance, indications, and interpretation. *Lancet* 2005; **365**:176–186.
14. Sun X, Briel M, Walter SD, Guyatt GH. Is a subgroup effect believable? Updating criteria to evaluate the credibility of subgroup analyses. *British Medical Journal* 2010; **340**:850–854.
15. Berry DA. Subgroup analysis. *Biometrics* 1990; **46**:1227–1230.
16. Meinshausen N, Meier L, Bühlmann P. *P*-values for high-dimensional regression. *Journal of the American Statistical Association* 2009; **104**:1671–1681.
17. Lockhart R, Taylor J, Tibshirani RJ, Tibshirani R. A significance test for the lasso. *The Annals of Statistics* 2014; **42**: 413–463.
18. Meinshausen N, Bühlmann P. Stability selection. *Journal of Royal Statistical Society. Series B* 2010; **72**:417–473.
19. Freidlin B, Simon R. Adaptive signature design: an adaptive clinical trial design for generating and prospectively testing a gene expression signature for sensitive patients. *Clinical Cancer Research* 2005; **11**:7872–7878.
20. Simon RM, Subramanian J, Li MC, Menezes S. Using cross validation to evaluate the predictive accuracy of survival risk classifiers based on high dimensional data. *Briefings in Bioinformatics* 2011:1–12.
21. Gunter L, Zhu J, Murphy S. Variable selection for qualitative interactions in personalized medicine while controlling the familywise error rate. *Journal of Biopharmaceutical Statistics* 2011; **21**:1063–1078.
22. Dmitrienko A, D'Agostino RB. Tutorial in biostatistics: traditional multiplicity adjustment methods in clinical trials. *Statistics in Medicine* 2013; **32**:5172–5218.
23. Loh WY, Shih YS. Split selection methods for classification trees. *Statistica Sinica* 1997; **7**:815–840.
24. Hothorn T, Hornik K, Zeileis A. Unbiased recursive partitioning: a conditional inference framework. *Journal of Computational and Graphical Statistics* 2006; **15**:651–674.
25. Berger J, Wang X, Shen L. A Bayesian approach to subgroup identification. *Journal of Biopharmaceutical Statistics* 2014; **24**:110–129.
26. Knaus WA, Draper EA, Wagner DP, Zimmerman JE. APACHE II: a severity of disease classification system. *Critical Care Medicine* 1985; **13**:818–829.
27. Xu Y, Yu M, Zhao YQ, Li Q, Wang S, Shao J. Regularized outcome weighted subgroup identification for differential treatment effects. *Biometrics* 2015; **71**:645–653.
28. Little RJ, Rubin DB. Causal effects in clinical and epidemiological studies via potential outcomes. *Annual Review of Public Health* 2000; **21**:121–45.
29. Ruberg SJ, Chen L, Wang Y. The mean does not mean as much anymore: finding sub-groups for tailored therapeutics. *Clinical Trials* 2010; **7**:574–583.
30. Dmitrienko A, Lipkovich I, Hopkins A, Li YP, Wang W. Biomarker Evaluation and subgroup identification in a Pneumonia Development Program using SIDES. In *Applied Statistics in Biomedicine and Clinical Trials Design*, Chen Z, Liu A, Qu Y, Tang L, Ting N, Tsong Y (eds). Springer: New York, 2015; 427–466.
31. Ruberg SJ, Shen L. Personalized medicine: four perspectives for clinical drug development. *Statistics in Biopharmaceutical Research* 2015; **7**:214–229.
32. Lipkovich I, Dmitrienko A. Biomarker identification in clinical trials. In *Clinical and Statistical Considerations in Personalized Medicine*, Carini C, Chang M (eds). Chapman and Hall/CRC Press: New York, 2014; 211–264.
33. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Wadsworth: Belmont, CA, 1984.
34. Breiman L. Statistical modeling: the two cultures. *Statistical science* 2001; **16**:199–231.
35. Royston P, Altman DG. Regression using factional polynomials of continuous covariates: parsimonious parametric modeling (with discussion). *Applied Statistics* 1994; **43**:429–467.
36. Royston P, Sauerbrei W. A new approach to modelling interaction between treatment and continuous covariates in clinical trials by using fractional polynomials. *Statistics in Medicine* 2004; **23**:2509–2525.
37. Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* 1996; **58**:267–288.
38. Imai K, Ratkovic M. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics* 2013; **7**:443–470.
39. Cai T, Tian L, Wong P, Wei LJ. Analysis of randomized comparative clinical trial data for personalized treatment selections. *Biostatistics* 2011; **12**:270–282.
40. Zhao L, Tian L, Cai T, Claggett B, Wei L. Effectively selecting a target population for a future comparative study. *Journal of the American Statistical Association* 2013; **108**:527–539.
41. Song X, Pepe MS. Evaluating markers for selecting a patient's treatment. *Biometrics* 2004; **60**:874–883.
42. Huang Y, Gilbert PB, Janes H. Assessing treatment-selection markers using a potential outcomes framework. *Biometrics* 2012; **68**:687–696.

43. Foster JC, Taylor JMC, Ruberg SJ. Subgroup identification from randomized clinical trial data. *Statistics in Medicine* 2011; **30**:2867–2880.
44. Breiman L. Random forests. *Machine Learning* 2001; **45**:5–32.
45. Dusseldorp E, Conversano C, Van Os BJ. Combining an additive and tree-based regression model simultaneously: STIMA. *Journal of Computational and Graphical Statistics* 2010; **19**:514–530.
46. Dixon DO, Simon R. Bayesian subset analysis. *Biometrics* 1991; **47**:871–882.
47. Hodges JS, Cui Y, Sargent DJ, Carlin BP. Smoothing balanced single-error-term analysis of variance. *Technometrics* 2007; **49**:12–25.
48. Gelman A, Hill J, Yajima M. Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness* 2012; **5**:189–211.
49. Gelman A, Jakulin A, Pittau MG, Su YS. A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics* 2008; **2**:1360–1383.
50. Park T, Casella G. The Bayesian lasso. *Journal of the American Statistical Association* 2008; **103**:681–686.
51. Gu X, Yin G, Lee JJ. Bayesian two-step lasso strategy for biomarker selection in personalized medicine development for time-to-event endpoints. *Contemporary Clinical Trials* 2013; **36**:642–650.
52. Chipman HA, George EI, McCulloch RE. Bayesian CART model search. *Journal of the American Statistical Association* 1997; **93**:935–960.
53. Shen J, He X. Inference for subgroup analysis with a structured logistic-normal mixture model. *Journal of the American Statistical Association* 2015; **110**:303–312.
54. Hoerl AE, Kennard R. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 1970; **12**:55–67.
55. Hastie T, Friedman J, Tibshirani R. *The Elements of Statistical Learning* 2nd edition. Springer-Verlag: New York, 2009.
56. Vapnik VN. *The Nature of Statistical Learning Theory*. Springer: New York, 1995.
57. Zou H, Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B* 2005; **67**:301–320.
58. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *The Annals of Statistics* 2004; **32**:407–499.
59. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B* 2004; **67**:91–108.
60. Zou H. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* 2006; **101**:1418–1429.
61. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society. Series B* 2007; **68**:49–67.
62. Hastie T, Rosset S, Tibshirani R, Zhu J. The entire regularization path for the support vector machine. *Journal of Machine Learning Research* 2004; **5**:1391–1415.
63. Rosset S, Zhu J. Piecewise linear regularized solution paths. *The Annals of Statistics* 2007; **35**:1012–1030.
64. Friedman JH. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* 2001; **29**:1189–1232.
65. Wahba G. *Spline Models for Observational Data*. SIAM: Philadelphia, 1990.
66. Friedman JH, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* 2000; **28**:337–407.
67. Chipman HA, George EI, McCulloch RE. BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 2010; **4**:266–298.
68. Efron B. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association* 1983; **78**:316–331.
69. Negassa A, Ciampi A, Abrahamowicz M, Shapiro S, Boivin JF. Tree-structured subgroup analysis for censored survival data: validation of computationally inexpensive model selection criteria. *Statistics and Computing* 2005; **15**:231–239.
70. Su X, Zhou T, Yan X, Fan J, Yang S. Interaction trees with censored survival data. *The International Journal of Biostatistics* 2008; **4**(Issue 1). Article 2.
71. Su X, Tsai CL, Wang H, Nickerson DM, Li B. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research* 2009; **10**:141–158.
72. Loh WY, He X, Man M. A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine* 2015; **34**:1818–1833.
73. Loh WY. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica* 2002; **12**:361–386.
74. Zeileis A, Hothorn T, Hornik K. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics* 2008; **17**:492–514.
75. Dusseldorp E, Van Mechelen I. Qualitative interaction trees: a tool to identify qualitative treatment-subgroup interactions. *Statistics in Medicine* 2014; **33**:219–237.
76. Tian L, Alizaden AA, Gentles AJ, Tibshirani R. A simple method for detecting interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association* 2014; **109**:1517–1532.
77. Jones HE, Ohlssen DI, Neuenschwander B, Racine A, Branson M. Bayesian models for subgroup analysis in clinical trials. *Clinical Trials* 2011; **8**:129–143.
78. Leblanc M, Crowley J. Survival trees by goodness of split. *Journal of the American Statistical Association* 1993; **88**: 457–467.
79. Qian M, Murphy SA. Performance guarantees for individualized treatment rules. *The Annals of Statistics* 2011; **39**: 1180–1210.
80. Zhao Y, Zeng D. Recent development on statistical methods for personalized medicine discovery. *Frontiers of Medicine* 2013; **7**:102–110.
81. Lu W, Zhang HH, Zeng D. Variable selection for optimal treatment decision. *Statistical Methods in Medical Research* 2011; **22**:493–504.
82. Foster JC, Taylor JMG, Kaciroti N, Nan B. Simple subgroup approximation to optimal treatment regimes from randomized clinical trial data. *Biostatistics* 2015; **16**:368–382.

83. Zhao Y, Zheng D, Rush AJ, Kosorok MR. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association* 2012; **107**:1106–1118.
84. Zhang B, Tsiatis AA, Davidian M, Zhang M, Laber EB. Estimating optimal treatment regimes from a classification perspective. *Statistics* 2012; **1**:103–114.
85. Zhang B, Tsiatis AA, Laber EB, Davidian M. A robust method for estimating optimal treatment regimes. *Biometrics* 2012; **68**:1010–1018.
86. Fu H, Zhou J, Faries DE. Estimating optimal treatment regimes via subgroup identification in randomized control trials and observational studies. *Statistics in Medicine* 2016; **35**(19):3285—3302.
87. Laber EB, Zhao YQ. Tree-based methods for individualized treatment regimes. *Biometrika* 2015; **102**:501–514.
88. Zhang Y, Laber EB, Tsiatis A, Davidian M. Using decision lists to construct interpretable and parsimonious treatment regimes. *Biometrics* 2015; **71**:895–904.
89. Huang Y, Fong Y. Identifying optimal biomarker combinations for treatment selection via a robust kernel method. *Biometrics* 2014; **70**:891–901.
90. Kehl V, Ulm K. Responder identification in clinical trials with censored data. *Computational Statistics and Data Analysis* 2006; **50**:1338–1355.
91. Lipkovich I, Dmitrienko A, Denne J, Enas G. Subgroup identification based on differential effect search (SIDES): a recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine* 2011; **30**:2601–2621.
92. Friedman JH, Fisher NI. Bump hunting in high-dimensional data. *Statistics and Computing* 1999; **9**:123–143.
93. Chen G, Zhong H, Belousov A, Viswanath D. PRIM approach to predictive-signature development for patient stratification. *Statistics in Medicine* 2015; **34**:317–342.
94. Lipkovich I, Dmitrienko A. Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using SIDES. *Journal of Biopharmaceutical Statistics* 2014; **24**:130–153.
95. Sivaganesan S, Laud PW, Mueller P. A Bayesian subgroup analysis with a zero-enriched Polya Urn scheme. *Statistics in Medicine* 2011; **30**:312–323.
96. Hardin DS, Rohwer RD, Curtis BH, Zagar A, Chen L, Boye KS, Jiang HH, Lipkovich IA. Understanding heterogeneity in response to antidiabetes treatment: a post hoc analysis using SIDES, a subgroup identification algorithm. *Journal of Diabetes Science and Technology* 2013; **7**:420–429.
97. Westfall PH, Young SS. *Resampling-based Multiple Testing: Examples and Methods for P-value Adjustment*. John Wiley: New York, 1993.
98. Bien J, Taylor J, Tibshirani R. A lasso for hierarchical interactions. *The Annals of Statistics* 2013; **41**:1111–1141.
99. Lim M, Hastie T. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics* 2014; **24**:627–654.
100. Kang C, Janes H, Huang Y. Combining biomarkers to optimize patient treatment recommendations. *Biometrics* 2014; **70**:695–720.