Optimistic bias in the evaluation of statistical methods: illustrations and possible solutions

Christina Sauer (geb. Nießl)

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

eingereicht am 09.09.2025

Erstgutachterin: Prof. Dr. Anne-Laure Boulesteix
Zweitgutachterin: Prof. Dr. Sarah Friedrich-Welz
Drittgutachter: Prof. Dr. Mark Robinson
Tag der Disputation: 19.12.2025

Summary

Benchmark studies are an important tool for assessing the properties of statistical methods by evaluating and comparing them on simulated or real data. Conducting such studies requires researchers to make many choices, for example the specific methods to compare as well as the data and performance measures to use for the assessment. From applied research, which examines the models produced by methods rather than the methods themselves, it is well known that such flexibility, combined with the inherent non-neutrality of researchers, may lead to results biased in the direction of their expectations. This deviation can be referred to as optimistic bias and may, for example, manifest as false positive rejections in hypothesis testing. In light of this, there is concern that optimistic bias may also occur in benchmark studies. Such bias is particularly likely to arise in studies that accompany the proposal of a new method, where researchers are clearly not neutral, potentially causing false claims of superiority.

This thesis adds to existing work by broadening the discussion on how optimistic bias can arise in benchmark studies, while also addressing the possibility that performance differences between studies result from factors other than optimistic bias. Furthermore, it provides additional strategies to reduce optimistic bias. To this end, the cumulative thesis comprises four contributions.

The first contribution considers the often-overlooked role of preprocessing steps, such as variable selection or transformation, in the generation and evaluation of prediction models. By formalizing these choices as preprocessing hyperparameters, it highlights their impact and potential for misuse. While being the only contribution not situated in methodological but in applied research, the insights of this contribution are relevant to both contexts, as the evaluation procedures it discusses closely parallel those used in benchmark studies.

The second contribution extends an existing benchmark study to empirically illustrate how results can vary when different design and analysis decisions are made, and how this variability can be easily exploited to obtain favorable results. As the first contribution, it also examines important but rarely addressed choices, specifically the handling of missing performance values and the derivation of method rankings. It further proposes an approach for visualizing the results obtained from different benchmark variants.

The widely noted tendency for newly proposed methods to perform best in the benchmark studies accompanying their introduction is the focus of the third contribution. Through a cross-design validation experiment, where two methods are reevaluated using each other's original benchmark study setup, it explores the roles of optimistic bias, researcher expertise, and mismatches between original and subsequent study settings in explaining performance differences.

Finally, the fourth contribution focuses on the choice of data in benchmark studies, in particular the generation of data using parametric simulations. A common approach is

to base these simulations on real datasets, yet in practice only one or two datasets are typically used, and the rationale for their selection is often unclear. In addition to formalizing real-data-based parametric simulations, the fourth contribution promotes a more systematic procedure for selecting real datasets, clarifying the data settings to which the benchmark study's conclusions are intended to generalize and increasing their representativeness for that scope.

Zusammenfassung

Benchmarkstudien, in denen Methoden anhand simulierter oder realer Daten evaluiert und verglichen werden, sind ein wichtiges Instrument zur Beurteilung der Eigenschaften statistischer Methoden. Die Durchführung solcher Studien erfordert zahlreiche Entscheidungen, die von den Forschenden getroffen werden müssen. Dazu gehört beispielsweise die Auswahl der zu vergleichenden Methoden sowie der Datensätze und Performance-Maße, die für die Evaluation verwendet werden sollen. Aus der angewandten Forschung, welche nicht die Methoden selbst, sondern die von ihnen erzeugten Modelle untersucht, ist hinlänglich bekannt, dass eine solche Flexibilität in Verbindung mit fehlender Neutralität der Forschenden zu Ergebnissen führen kann, die in Richtung ihrer Erwartungen verzerrt sind. Diese Verzerrung kann als optimistischer Bias bezeichnet werden und sich beispielsweise in einer fälschlichen Ablehnung der Nullhypothese im Kontext von Hypothesentests äußern. Vor diesem Hintergrund besteht die Möglichkeit, dass ein solcher Bias auch in Benchmarkstudien zu finden ist. Besonders wahrscheinlich ist dies in Studien, welche die Vorstellung einer neuen Methode begleiten, da die Forschenden hier offenkundig nicht neutral sind, was zu einer Überschätzung der Methodenperformance führen kann.
Diese Dissertation erweitert die bestehende Literatur, indem sie die Diskussion darüber, wie ein optimistischer Bias in Benchmark-Studien entstehen kann, vertieft und zugleich die Möglichkeit berücksichtigt, dass Performance-Unterschiede zwischen Studien auch auf andere Faktoren zurückzuführen sind. Darüber hinaus werden zusätzliche Strategien zur Verringerung von optimistischem Bias vorgestellt. Diese Aspekte werden in vier Beiträgen untersucht.
Der erste Beitrag befasst sich mit der oft vernachlässigten Rolle von Preprocessing-Schritten, etwa der Selektion oder Transformation von Variablen, bei der Entwicklung und Evaluation von Prädiktionsmodellen. Durch die formale Einordnung dieser Entscheidungen als Preprocessing-Hyperparameter wird deren Bedeutung und das Risiko einer methodisch unsachgemäßen Handhabung deutlich gemacht. Obwohl dieser Beitrag als einziger nicht im Bereich der methodologischen, sondern der angewandten Forschung angesiedelt ist, sind die gewonnenen Erkenntnisse für beide Kontexte relevant, da die beschriebenen Evaluationsverfahren denen in Benchmarkstudien weitgehend entsprechen.
Der zweite Beitrag erweitert eine bestehende Benchmarkstudie, um empirisch zu zeigen, wie stark sich die Ergebnisse verändern können, wenn einzelne Komponenten der Studie variiert werden, und wie diese Variabilität ausgenutzt werden kann, um vorteilhafte Ergebnisse zu erzielen. Wie der erste Beitrag betrachtet auch dieser wichtige, aber selten untersuchte Entscheidungen, hier insbesondere den Umgang mit fehlenden Performance-Werten und die Erstellung von Methodenrankings. Darüber hinaus wird ein Ansatz zur Visualisierung der Ergebnisse verschiedener Varianten einer Benchmarkstudie vorgestellt.
Die weithin beobachtete Tendenz, dass neu entwickelte Methoden in Benchmarkstudien

zur Einführung der Methode die beste Performance aufweisen, steht im Fokus des dritten Beitrags. Anhand eines Cross-Design-Validation-Experiments, in dem zwei Methoden in dem ursprünglichen Benchmark-Setup der jeweils anderen Methode erneut evaluiert werden, wird untersucht, welche Rollen der optimistische Bias, die Expertise der Forschenden und Abweichungen zwischen den ursprünglichen und den späteren Studiensettings bei der Erklärung von Performance-Unterschieden spielen.
Der vierte Beitrag widmet sich schließlich der Datenauswahl in Benchmarkstudien, insbesondere der Erzeugung von Daten mittels parametrischer Simulationen. Eine gängige Vorgehensweise besteht darin, diese Simulationen auf realen Datensätzen zu basieren; in der Praxis werden jedoch meist nur ein oder zwei solcher Datensätze verwendet und die Kriterien für deren Auswahl sind oft unklar. Neben der formalen Behandlung realdatenbasierter parametrischer Simulationen wird in diesem Beitrag ein systematisches Verfahren zur Auswahl geeigneter Datensätze vorgeschlagen, das die Datensettings, auf welche die Schlussfolgerungen der Benchmarkstudie generalisiert werden sollen, transparenter macht und zugleich ihre Repräsentativität für diesen Zielbereich erhöht.

Acknowledgments

First of all, I would like to express my sincere gratitude to Anne-Laure for her invaluable supervision. Thank you for continuously encouraging me, offering steady support, and welcoming new ideas with openness. I truly could not have wished for better guidance during my PhD.

Furthermore, I would like to thank:

- *Prof. Dr. Sarah Friedrich-Welz and Prof. Dr. Mark Robinson for kindly taking on the role of reviewers of this thesis.*
- *Prof. Dr. Thomas Augustin and Prof. Dr. Bernd Bischl for their availability to be part of the examination panel.*
- *My colleagues at the IBE for creating such a supportive and enjoyable working environment and my co-authors, both from IBE and beyond, for the fruitful collaborations and the opportunity to learn from their expertise.*
- *Dr. Sabine Hoffmann and the StaBLab working group, who supported the completion of this thesis through their understanding and kindness.*
- *Milena Wünsch for carefully reading the thesis and providing valuable comments.*

Finally, I would like to thank my wonderful friends for always being there for me and for knowing exactly what I need. I am also deeply grateful to my family, especially my parents for their constant support and my sister who has supported me at every stage of my academic path, from refusing to let me quit during the first week of my bachelor's to providing advice during the writing of this thesis. Most importantly, I thank my husband. Listing all the reasons would fill another thesis, so I will keep it short: you are everything.

Contents

1 Introduction and motivation **1**

2 Terminology **3**
2.1 Benchmark study 3
2.2 Statistical method 4

3 Components of benchmark studies **6**
3.1 Aim 6
3.2 Design 8
3.2.1 Method implementations 8
3.2.2 Data-generating mechanisms 11
3.2.3 Performance assessment 15
3.3 Analysis 17
3.3.1 Handling of missing performance values 18
3.3.2 Derivation of method rankings 19

4 Optimistic bias in benchmark studies **22**
4.1 Structural risk factors 22
4.2 Mechanism and manifestations 24
4.3 Assessment 28
4.4 Possible solutions 32

5 Summary of the contributions **36**

6 Outlook **39**

References **42**

A Contribution 1: "Beyond algorithm hyperparameters: on preprocessing hyperparameters and associated pitfalls in machine learning applications" **55**

B Contribution 2: "Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results" **128**

C Contribution 3: "Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment" **158**

D Contribution 4: "Statistical parametric simulation studies based on real data" **196**

1 Introduction and motivation

Methodological research in statistics and related computational fields generally encompasses all activities aimed at advancing statistical methods. While the development of new methods is a central focus, methodological work also involves generating evidence about their properties. Such evidence can be obtained through theoretical analysis, but an equally important component is the collection of *empirical evidence* by evaluating and comparing methods on simulated or real data (Heinze et al., 2024). Work of this kind can be referred to as *benchmark studies* and may serve different purposes, such as guiding applied method users in making informed choices for a given task and assisting method developers in identifying opportunities for further improvement (Weber et al., 2019).

Given the considerable impact that benchmark studies can have on multiple research fields, it is particularly important that they are conducted with sufficient rigor to ensure valid conclusions. To identify potential threats that could undermine the conclusions drawn from benchmark studies, it is useful to first review considerations already discussed in applied research, an approach also taken by Boulesteix et al. (2017) and Hullman et al. (2022). In contrast to methodological research, applied research does not aim to evaluate methods, but employs the models they produce to address substantive questions. Over the past two decades, replication efforts in many applied research fields, including psychology and preclinical cancer biology, have frequently been unsuccessful, that is, attempts to repeat the original study's analysis with new data have failed to obtain the same results (see, e.g., Camerer et al., 2018; Errington et al., 2021; Open Science Collaboration, 2015). These outcomes have raised concerns about a possible *replication crisis* (see Wiggins and Christopherson, 2019 for a concise overview) and echo earlier warnings by Ioannidis (2005), who provocatively argued that "most published research findings are false."

As one possible reason for non-replicable results, practices commonly referred to as *p-hacking*, *fishing expeditions*, or *data dredging* have been identified (e.g., Davey Smith, 2002; Head et al., 2015; Munafò et al., 2017; Wagenmakers et al., 2012). In essence, these terms describe the (often unintentional) misuse of *researcher degrees of freedom* (Simmons et al., 2011), which refer to the set of choices open to researchers during data collection and analysis. If researchers make corresponding decisions in a way that supports their intended outcomes (often coinciding with what is publishable), this can lead to *optimistically biased* results, that is, results that systematically deviate from the truth in the direction of the researchers' hopes or expectations, with common manifestations including false rejections of the null hypothesis or inflated effect sizes (see, e.g., Ioannidis, 2008; Simmons et al., 2011).

In light of these considerations, it is reasonable to reflect on whether optimistic bias may also be present in the empirical evaluation of methods, potentially contributing to a

"replication crisis in methodological research" (Boulesteix, Hoffmann, et al., 2020). Such concerns are particularly pronounced for benchmark studies that accompany the proposal of a new method, as in this case researchers have a strong interest in presenting their method as superior (e.g., Boulesteix et al., 2013; Norel et al., 2011). Indeed, multiple empirical studies illustrate how readily optimistic bias can be introduced in such studies (e.g., Jelizarow et al., 2010; Keogh & Kasetty, 2003; Macià et al., 2013; Pawel et al., 2024; Ullmann et al., 2023; Yousefi et al., 2010), typically focusing on researcher degrees of freedom related to the data component (i.e. the selection of real data sets or the generation of simulated data) or to the choice and implementation of the methods under comparison.
This thesis aims to complement this line of meta-methodological work in several ways. In addition to proposing a way to systematically report and analyze the results of different benchmark study variants, it broadens the scope of the discussion by highlighting that optimistic bias can also occur in benchmark studies not introducing a new method, and by examining a wider range of researcher degrees of freedom. In particular, it considers the flexibility in aggregating raw performance values into method rankings and in handling missing performance values, the latter being an aspect that has generally received little attention in methodological research until recently (Pawel et al., 2025; Wünsch et al., 2025). Furthermore, although for example Jelizarow et al. (2010) explicitly show how the selection of variables can be exploited to achieve favorable method performance, it remains insufficiently recognized in both methodological and applied research that the choices concerning such preprocessing steps essentially constitute hyperparameters of the full analysis pipeline. Correspondingly, their impact and potential for misuse have received little attention. By providing a formal treatment of this issue, this thesis also seeks to contribute to closing this gap.
In addition to understanding how optimistic bias can arise, an equally important question concerns its prevalence. A valuable investigation in this regard is provided by Buchka et al. (2021), who compare the relative performance of newly proposed methods to their performance in subsequent studies conducted by other researchers. Most methods perform worse in these later evaluations, which may indicate optimistic bias. However, such discrepancies can also result from factors other than optimistic bias, specifically differences in the expertise of those applying the methods or mismatches between the settings used in the subsequent studies and those to which the original study's conclusions were intended to generalize. While the relevance of researcher expertise (e.g., Boulesteix et al., 2017; Duin, 1996) and issues of unclear generalization with respect to data settings (e.g., Boulesteix, Hable, et al., 2015; Strobl & Leisch, 2024) have already been discussed in the meta-methodological literature, this thesis explicitly addresses these factors in the context of assessing optimistic bias. In addition, in the specific context of simulating data from real datasets, it proposes and formalizes a more structured approach to specifying data generation, which, among other benefits, helps clarify the data settings to which a

benchmark study's conclusions are intended to generalize.
The remainder of this cumulative thesis is organized as follows. Sections 2-4 provide the conceptual basis, drawing on the literature to give a general perspective on benchmark studies and the notion of optimistic bias in this context. Specifically, Section 2 clarifies key terminology, followed by a review of the main components of benchmark studies in Section 3. Section 4 then examines optimistic bias in benchmark studies, focusing on how it arises, the challenges in assessing it, and possible solutions. Section 5 summarizes the four contributions of the thesis, and Section 6 outlines possible directions for future research. The subsequent Sections A-D contain the four contributions as presented in the four articles: Contribution 1 (Sauer et al., 2024), Contribution 2 (Nießl et al., 2022), Contribution 3 (Nießl et al., 2024), and Contribution 4 (Sauer et al., 2025).

2 Terminology

To provide a clear basis for the discussions that follow, this section defines the terms benchmark study (Section 2.1) and statistical method (Section 2.2).

2.1 Benchmark study

In this thesis, the term *benchmark study* refers to studies that evaluate and compare methods using simulated or real data to assess their empirical properties. Depending on the research field and the specific types of methods and data considered, alternative terms may be encountered, including *comparison study* (e.g., Boulesteix et al., 2013), *method evaluation study* (e.g., Kreutz, 2016), *(empirical) methodological study* (e.g., Lange et al., 2025), or *comparative simulation study* (e.g., Pawel et al., 2024). Although the terminology in Contributions 1–4 also varies depending on the context of each study, this thesis consistently uses the term *benchmark study*, as the contributions primarily focus on method examples from machine learning and bioinformatics, where this term is commonly used (e.g., Brooks et al., 2024; Hothorn et al., 2005; Weber et al., 2019).
To further clarify how benchmark studies are understood in this thesis, it is worthwhile to briefly discuss a few study types that are regarded as included or excluded. First, although benchmark studies (and also the term *empirical*) are sometimes associated exclusively with real data (e.g., Boulesteix et al., 2017), the definition adopted here, as stated above, explicitly includes both real and simulated data. This is consistent with the usage in, for instance, Hothorn et al. (2005) and Weber et al. (2019). Second, while studies comparing multiple methods are the primary focus (which also aligns with the common understanding of benchmark studies), studies evaluating a single method are also included. Obviously, statements in the thesis that assume the presence of multiple methods do not apply in such cases. Third, only benchmark studies that are published

in some form (e.g., in a scientific journal or in conference proceedings) are considered. These are typically either studies that are part of methodological papers introducing a new method, or standalone studies that explicitly focus on the comparison of existing methods. In the former case, the benchmark component must be used to draw conclusions about the empirical properties of the new method. Purely illustrative elements, such as demonstrations of how to implement the method or interpret its output (Boulesteix, 2013), are not sufficient to qualify as a benchmark study in the sense intended here. Benchmark studies comparing only existing methods may, under certain conditions, also be referred to as *neutral benchmark studies* (Boulesteix et al., 2013), with the specific criteria for this term outlined in Section 4.1. Finally, although competitions or community challenges can also be regarded as benchmark studies, they will not be treated as such. These formats are coordinated externally rather than conducted by a single research team, with data and evaluation prespecified by the organizers and the methods implemented by various participants (Weber et al., 2019). While they represent valuable complements to the types of studies discussed here, they typically focus less on investigating the properties of individual methods and more on what can be achieved under a specific data and evaluation setup, while also raising distinct methodological and practical considerations that are beyond the scope of this thesis (Kodalci & Thas, 2024; Kreutz, 2016).

2.2 Statistical method

In describing benchmark studies as evaluating and comparing statistical methods, the term *statistical* is understood in a broad sense to include, for example, machine learning algorithms and computational methods commonly used in bioinformatics (for simplicity, the qualifier *statistical* will be omitted in the following). Moreover, while the term *method* might initially suggest a core analytical procedure (such as a statistical test or a machine learning algorithm), it is not necessarily limited to these. For example, it may refer only to a specific component or aspect of such a core procedure (e.g., a particular hyperparameter), or to different pre- and postprocessing steps that are part of the overall data analysis pipeline (Demšar, 2006; Siepe et al., 2024). More generally, in the context of benchmark studies, the term method can refer to any part of a data analysis pipeline that aims to recover a specific target (i.e. the ground truth of interest), as determined by the task at hand. Common targets include estimands (for estimation tasks), true outcome values (for prediction tasks), true hypotheses (for hypothesis testing tasks), or design characteristics such as the optimal sample size (for sample size calculation tasks); see Morris et al. (2019) and Siepe et al. (2024) for an overview.

To give concrete examples of methods that may be evaluated in a benchmark study, five tasks and associated methods, all considered in the empirical illustrations of the four contributions of this thesis, are outlined below.

Testing treatment differences for ordinal outcomes A simple task featured in Contribution 4 is the comparison of treatments in a two-arm randomized controlled trial with an ordinal outcome. Here, the target is the validity of the null hypothesis of no treatment difference. Well-known methods that may be used for this task include the Chi-square test and the Wilcoxon rank-sum test (e.g., Agresti, 2010).

Prediction of palliative care costs Contribution 1 examines the prediction of costs for patients in a palliative care setting based on multiple variables (e.g., cognitive or physical symptoms) describing the care situation. This is an example of a supervised machine learning task, where prediction models are trained on a dataset with observed outcome values, and the target is the true outcome values of new observations (here, the palliative care costs; Bischl et al., 2023). All methods applied in Contribution 1 are tree-based algorithms and include the Classification and Regression Tree algorithm (Breiman et al., 1984) and the Conditional Inference Tree algorithm (Hothorn et al., 2006). For an introduction to tree-based algorithms, see Strobl et al. (2009).

Prediction of survival outcomes using multiomic data Contribution 2 also considers a supervised machine learning task, but with a survival outcome (i.e. information on the time of death and whether the patient is deceased or censored) and different variables used for prediction. Specifically, these variables include both clinical data and multiple types of high-dimensional molecular data, such as genomic and proteomic data (referred to as multiomic data; Subramanian et al., 2020). Methods designed for this type of task include Block forests (Hornung & Wright, 2019) and Priority-Lasso (Klau et al., 2018). For an overview and comparison of these and other methods, see the benchmark study by Herrmann et al. (2021).

Cancer subtyping using multiomic data Another application that uses multiomic data is cancer subtyping, which is considered in Contribution 3. The aim is to identify biologically or clinically meaningful clusters (in this context referred to as subtypes), with the target being the true, but unknown, clusters (Duan et al., 2021). Unlike the supervised cases above, this is an unsupervised learning task, as no true outcome values are available (Hastie et al., 2009). Methods suitable for this task include NEMO (Rappoport & Shamir, 2019) and PINSPlus (Nguyen et al., 2019); see the review by Subramanian et al. (2020) for a broader overview.

Differential gene expression analysis Both Contribution 3 and Contribution 4 employ methods for differential gene expression analysis using RNA-Seq data, which is a specific type of omic data. In this task, the aim is to identify genes whose RNA-Seq expression levels differ across conditions (e.g., cancer vs. normal tissues), with the target being, for each gene, the validity of the null hypothesis that it is not differentially expressed

between conditions. Popular methods include edgeR (Robinson et al., 2009) and DESeq2 (Love et al., 2014). For an overview and comparison of these and other methods, see, for example, the benchmark study by Baik et al. (2020).

3 Components of benchmark studies

Although benchmark studies encountered in the literature can differ substantially in how they are conducted, they typically follow a similar overarching structure. This section distinguishes three main building blocks of a benchmark study: the aim (Section 3.1), the study design (Section 3.2), and the analysis of results (Section 3.3). Each will be discussed in detail in the following, along with the individual components they comprise and the key decisions researchers face in specifying them. An overview is provided in Table 1.

Table 1: Overview of benchmark study components.

Aim (Section 3.1)	**Operationalization**	**Section**	**Design vs. analysis**
Methods of interest	Method implementations $\mathcal{A}_1, \ldots, \mathcal{A}_M$	3.2.1	Design
Population of data-generating mechanisms	Data-generating mechanisms $\mathcal{G}_1, \ldots, \mathcal{G}_L$ (either specified directly or implicitly via $\mathcal{D}_1, \ldots, \mathcal{D}_L$)	3.2.2	
Evaluation perspective	Performance measures $\mathcal{P}_1, \ldots, \mathcal{P}_P$, generation of n_{rep} sampled datasets	3.2.3	
	Handling of missing performance values	3.3.1	Analysis
	Derivation of method rankings	3.3.2	

3.1 Aim

The most fundamental decision in constructing a benchmark study lies in defining its aim. At its core, this means formulating one or more *research questions* that the study is intended to address. For simplicity, a single research question is assumed throughout this thesis; in cases with multiple questions, the considerations apply to each one individually. The research question is typically shaped by the intended audience of the benchmark study (Weber et al., 2019). For instance, if directed at applied method users, the question might take the form: "Which method should be preferred in which data settings encountered in specific real-world applications?" In contrast, for method developers, a more relevant question would be: "In which data settings does a specific method still

require improvement?" The formulation of the research question is usually also guided by the development stage of the methods of interest, or more precisely, by the *phase of methodological research* the benchmark study aims to contribute to (a concept proposed by Heinze et al. 2024 in analogy to the phases of drug development). For example, when evaluating a newly proposed method, a corresponding question could be of the form: "Are there data settings where the new method offers clear advantages over existing alternatives?" or "Does the new method perform better than relevant competitors with respect to specific evaluation criteria in data settings characterized by particular challenges?" Conversely, when substantial empirical evidence already exists, a relevant question could be formulated as: "How does the method perform in complex, extreme, or otherwise atypical data settings relative to its initial field of application?"

As a related but separate dimension, the formulation of the research question also varies along the continuum between *exploratory* and *confirmatory* research (also referred to as *hypothesis-generating* and *hypothesis-testing* research), a distinction that has long been discussed in applied research and has recently gained attention in empirical methodological research (see Herrmann et al. 2024; Lange et al. 2025 for detailed discussions). This dimension can be interpreted as reflecting the strength of evidence the study aims to provide. Exploratory-style questions may take the form "In the given data settings, where could method A potentially perform better than method B?", while confirmatory-style questions typically involve a hypothesis that can be evaluated, such as "Does method A perform better than method B in a given data setting?"

Despite differences in emphasis, all of these (still broadly formulated) questions can be viewed as instances of a generic structure: "How do [methods of interest] perform according to [evaluation perspective] in a [population of data-generating mechanisms[1]]?" Here, the bracketed components serve as placeholders that can be instantiated in various ways and at different levels of detail (specific examples will be given throughout Sections 3.2 and 3.3). Importantly, the data component in this structure does not refer to datasets but to *data-generating mechanisms* (DGMs), also referred to as *data-generating processes*. A DGM can be viewed as the complete probabilistic description of how data is generated, including all distributions, structural relationships, and parameters, whether fully specified in simulations or implicitly assumed in real-data contexts (Hothorn et al., 2005; Morris et al., 2019; see Section 3.2.2 details on the different types of data). Referring instead to a population of datasets, as done for example by Boulesteix et al. (2017), Brombacher et al. (2025), and Lange et al. (2025), is also valid and consistent with the definition above, since datasets are realizations of DGMs. However, focusing on a population of

[1]In Contribution 4, the term *domain of interest* is used to refer to the population of DGMs the benchmark study aims to draw conclusions about, but specifically refers to populations of real-world DGMs. While this is often the case in practice, a benchmark study may also consider DGMs that do not occur in the real world. To avoid redefining the meaning of domain of interest as used in Contribution 4, this term will not be used here.

DGMs instead of datasets is arguably more appropriate in the context of methodological research, which (unlike applied research) is typically not concerned with specific datasets but with performance across a broader range of data-generating scenarios. This focus is especially clear in simulation studies, where it is standard practice to draw conclusions about DGMs rather than about individual datasets.

As will be discussed in more detail in Section 3.2.1, an important consideration when specifying the research question is that method performance will usually depend on the expertise of the researchers implementing the methods (Duin, 1996). Therefore, unless the benchmark study explicitly investigates robustness with respect to these factors (see Section 3.2.3), any research question (regardless of how it is formulated) implicitly includes the qualifier: "Given the methods are applied by a user with specific expertise."

Finally, note that the following Sections 3.2 and 3.3 proceed under the assumption of a formally specified research question, neglecting the fact that this is likely an idealized view. This issue, and the problems it can cause, will be addressed in more detail in Section 4.

3.2 Design

The design of a benchmark study, as defined in this thesis, encompasses all components that must be specified prior to executing the study in order to obtain raw performance results.[2] This includes the specification of method implementations (Section 3.2.1), data-generating mechanisms (Section 3.2.2), and both performance measures and the procedure for generating the sampled datasets (Section 3.2.3). Together with the analysis of performance results (Section 3.3), the design serves to operationalize the research question formulated in the aim.

3.2.1 Method implementations

The purpose of the method-related component of the study design is to translate the method specification as stated in the research question ("[methods of interest]" in Table 1 and Section 3.1) into a set of M fully defined analysis pipelines $\mathcal{A}_1, \ldots, \mathcal{A}_M$ that can be evaluated within the benchmark study. The flexibility available in this process depends on the level of detail with which the methods of interest are defined in the research question. The broadest (and at the same time minimal) specification consists in identifying methods solely by the target(s) they are intended to recover. This ensures that performance comparisons remain meaningful. Beyond this, corresponding method specifications are often vague. They may refer to broad classes of methods (e.g., "tree-based algorithms",

[2]The term "(study) design" may also be used in a broader sense to include aspects that are here considered separately as analysis. This usage is also adopted in Contribution 3, mainly for pragmatic reasons.

"regularized regression methods") or to criteria based on reputation or frequency of use (e.g., "popular methods", "state-of-the-art methods"). More explicit specifications are typically found in benchmark studies evaluating a newly proposed method, where it is at least inherently clear that the new method must be included. However, even in such cases, the number and type of competing methods are often only loosely defined, (e.g., "relevant competitors").

Accordingly, researchers must first define the M methods to be investigated, typically referring only to the specific part of the analysis pipeline that the study focuses on. For example, this could be {Classification and Regression Tree, Conditional Inference Tree, ...}; or, in studies considering preprocessing rather than core analytical procedures, {missing value imputation using the median, imputation using k-nearest neighbors, ...}. However, specifying the general methods to be considered is typically not sufficient to define the full analysis pipelines that will be executed. For each method, researchers must additionally specify (i) the configuration of the method's hyperparameters, (ii) the associated pre- and postprocessing steps (or, more generally, all remaining steps in the pipeline if the method is not the core analytical procedure), and (iii) the computational environment and constraints. In the remainder of this thesis, a fully specified analysis pipeline will be referred to as a *method implementation* $\mathcal{A}$, while the term *method* will, as introduced in Section 2.2, continue to denote the general form of a method under study (with a single method typically giving rise to multiple method implementations).

This section continues by discussing the decisions required to specify a method implementation, as well as the role of researcher expertise in this process.

Hyperparameters The *hyperparameters* of a method, also referred to as *method parameters* (Boulesteix et al., 2013), generally determine the specific configuration of a method and can substantially affect its performance (Bischl et al., 2023). For example, for a tree-based algorithm, a hyperparameter may be the minimum number of observations in each terminal node, or the choice of splitting criterion (with the latter illustrating that hyperparameters can also be categorical). Unless already specified in the research question, all existing hyperparameters must be set as part of defining the method implementation, for which one of three approaches may be chosen. First, if available, default values defined by the chosen software package (see below) can be used. Alternatively, researchers may choose to adjust the defaults to improve the method's performance. This can be done either data-independently (based, for instance, on findings from previous benchmark studies) or in a data-dependent way, commonly referred to as *tuning* or *(data-driven) hyperparameter optimization* (e.g., Bartz et al., 2023; Bischl et al., 2023; Probst et al., 2019). Tuning may be performed manually (often in an informal way) or, preferably, through an automated procedure. In the latter case, additional choices are required, such as how candidate hyperparameter values are generated and when the tuning process

is terminated. Further details on the purpose and selection of hyperparameters in the context of supervised machine learning are provided in Contribution 1.[3]
Note that instead of fixing hyperparameter values, researchers may also specify the tuning procedure itself as part of the method implementation, which is particularly common in machine learning. The resulting method implementation can then be considered "self-tuning" (Bischl et al., 2023). Importantly, this introduces stochasticity into the method implementation (although other sources of stochasticity may already exist, such as random feature selection in random forest algorithms; Bouthillier et al., 2021).

Pre- and postprocessing steps Although it may not receive particular attention when the benchmark study focuses on comparing core analytical procedures, researchers must still specify any pre- and postprocessing steps to ensure that the entire analysis pipeline is clearly defined. As the name suggests, *preprocessing* can be defined as any steps applied to a dataset in its rawest available form within the benchmark study, prior to the core analytical procedure. This includes, for example, the handling of missing values or the transformation and filtering of variables (Kapoor et al., 2024). In contrast, *postprocessing* steps comprise all operations performed after the core analytical procedure (Li et al., 2019). An example of this is the adjustment of p-values for multiple testing in differential gene expression analysis. In principle, whether pre- and postprocessing steps are applied in the analysis pipeline, and in what form, can also be treated as hyperparameters of the analysis pipeline (Binder & Pfisterer, 2024; Bischl et al., 2023), with all considerations discussed above applying accordingly. For example, the decision of whether to transform a variable, and the choice of transformation (e.g., logarithmic or square root), can be regarded as two preprocessing hyperparameters (where the latter is conditional on the former, as it is only relevant if the transformation is performed; Feurer and Hutter, 2019). Although this perspective is not yet widely established, it helps emphasize the importance of pre- and postprocessing steps by linking them to the well-recognized influence of hyperparameters on method performance. In Contribution 1, this is formalized and discussed in detail for preprocessing hyperparameters in the context of supervised machine learning. The framing of pre- and postprocessing steps as hyperparameters will also be adopted in the remainder of this thesis; unless stated otherwise, references to a method's hyperparameters will refer to the full set of hyperparameters of the analysis pipeline in which the method is embedded.

Computational environment and constraints To fully specify a method implementation, researchers must also define the computational environment in which it is

[3]Although Contribution 1 adopts the perspective of applied method users (concerned with evaluating the final prediction model produced by $\mathcal{A}$) rather than the methodological focus of benchmark studies (concerned with evaluating $\mathcal{A}$ itself), the evaluation procedures employed may, in fact, be the same. Accordingly, all explanations in Contribution 1 that do not concern a "final prediction model" (which is not relevant in benchmark studies and often does not exist) remain applicable.

executed, as this affects both the feasibility and performance of the implementation. This includes aspects such as the operating system, programming language, software packages and their versions, as well as hardware specifications such as the number of CPU cores and the available memory (Hodges et al., 2022; Pawel et al., 2025; Weber et al., 2019). In addition, it is often necessary to define computational constraints that limit the use of these resources, for example, by setting a maximum total runtime or restricting the time available for hyperparameter tuning (Lucic et al., 2018; Wünsch et al., 2025).

Researcher expertise An important factor in method implementation is the *expertise* of the researchers involved. In this context, expertise refers to both theoretical knowledge of a method and the practical experience gained through its repeated use. Importantly, it is not an additional choice to be made, but rather a latent factor that influences how the decisions described above are made and that cannot be directly controlled (at most, researchers with high expertise may deliberately mimic non-experts if this reflects the target audience, but not the other way round). Expertise is particularly relevant when setting hyperparameters. As stated above, three general strategies exist: using default values, relying on data-independent decisions, or applying data-dependent tuning. Setting aside the option of tuning, researchers with limited expertise typically rely on defaults, while experienced researchers may be more willing to deviate from default values and make informed decisions that are more likely to result in improved performance (Duin, 1996). Even when tuning is performed, expertise remains influential. For example, it guides decisions about which hyperparameters are worth tuning (i.e. "tunability"; Probst et al., 2019) and how to define suitable search spaces (Bischl et al., 2023). Expertise can also affect how the computational environment is chosen and managed during execution. For instance, experienced researchers may know which software packages are more efficient or stable, and how to resolve issues that arise while running a method (e.g., when it fails to execute as expected).

3.2.2 Data-generating mechanisms

To operationalize the population of DGMs specified in the research question ("[population of data-generating mechanisms]" in Table 1 and Section 3.1) in terms of a concrete study design, researchers must specify a set of L DGMs, $\mathcal{G}_1, \ldots, \mathcal{G}_L$, for which the performance of the M method implementations (Section 3.2.1) is to be assessed. However, before defining a concrete set of DGMs, a more general decision must be made regarding the type of data to be used. When discussing both the choice of data type and the specification of a concrete set of DGMs in the following, it will be assumed that only one type of data is employed in the benchmark study, although in practice, multiple types may be combined (Friedrich & Friede, 2024).

Data type Two general options can be distinguished for the type of data used in benchmark studies (see also Contribution 4 for a detailed discussion): *parametric simulation* and the use of *real data*, with the corresponding benchmark studies referred to as *parametric simulation studies* and *real-data studies*, respectively. They differ primarily in how the DGMs are defined and how the datasets for method evaluation can be obtained. In a parametric simulation, each DGM $\mathcal{G}_l$ is a parametric stochastic model that can be represented in closed form and is fully researcher-specified (Morris et al., 2019; Schreck et al., 2024). To assess method performance on a given $\mathcal{G}_l$, n_{rep} datasets are independently drawn from $\mathcal{G}_l$ (with n_{rep} set by the researchers) and used as inputs for the method implementations. Each such dataset will be referred to as a *sampled dataset* and is considered a realization of the corresponding DGM (as the generation of sampled datasets pertains to performance assessment rather than DGM specification, it is considered in more detail in Section 3.2.3).

As an alternative to parametric simulation, researchers may choose not to define the DGMs under study explicitly, but instead select L real datasets, $\mathcal{D}_1, \ldots, \mathcal{D}_L$, each assumed to have been generated by a (largely) unknown DGM, $\mathcal{G}_1, \ldots, \mathcal{G}_L$ (Hothorn et al., 2005). As in parametric simulation, the goal remains to assess the performance of each method implementation on $\mathcal{G}_1, \ldots, \mathcal{G}_L$; however, the specification of these DGMs is determined only implicitly through the selection of the corresponding real datasets.[4] Importantly, for each $\mathcal{G}_l$, only a single realization is available: the real dataset $\mathcal{D}_l$ (Friedrich & Friede, 2024). Therefore, to estimate performance, researchers may either apply the method implementations directly to $\mathcal{D}_l$ (implying $n_{\text{rep}} = 1$), or use a resampling scheme to draw datasets from $\mathcal{D}_l$ (again, see Section 3.2.3). This second strategy corresponds to defining an emulated DGM, $\widehat{\mathcal{G}}_l$, that is intended to approximate the unknown DGM $\mathcal{G}_l$ in the sense that performance estimates obtained from $\widehat{\mathcal{G}}_l$ are similar to those that would have been obtained if repeated sampling from the true DGM $\mathcal{G}_l$ were possible (Hothorn et al., 2005). Regardless of whether the method implementations are applied directly to the real datasets or to datasets drawn from an emulated DGM, the corresponding datasets will, as in parametric simulation, be referred to as *sampled datasets*, while the original datasets will continue to be referred to as real datasets $\mathcal{D}_1, \ldots, \mathcal{D}_L$ (noting that for the first strategy, the real and sampled datasets coincide).

The distinction outlined above can be further refined through the following remarks. First, using an emulated DGM where datasets are generated via resampling from a real dataset can also be viewed as a form of non-parametric simulation (e.g., Morris et al., 2019), since the sampled datasets are not real datasets themselves (even if they consist

[4]This DGM-based perspective may be unfamiliar to researchers focused solely on benchmark studies using real datasets, where performance is often assessed in terms of the datasets themselves rather than their underlying DGMs. However, as already argued in Section 3.1, in methodological research, the interest usually lies not in the dataset per se, but in what it represents: a DGM, which in turn reflects a broader population of DGMs.

of real observations). However, particularly in prediction contexts, it is more common to categorize such resampling-based evaluations as real-data studies (e.g., Boulesteix, Hable, et al., 2015). Second, real datasets may also be incorporated into parametric simulations, for example by estimating parameters of a given DGM $\mathcal{G}_l$ from one or more real datasets. This is discussed in detail in Contribution 4. Nevertheless, unlike in real-data studies, the DGM of interest in such real-data-based parametric simulation is still $\mathcal{G}_l$, not the (unknown) DGM(s) underlying the real dataset(s) used to construct it.[5] Third, it is also possible to combine elements from both parametric simulation and resampling from real datasets. This approach can be referred to as *semi-parametric simulation*, or, for specific implementations, as *statistical Plasmode simulation* (Franklin et al., 2014; Schreck et al., 2024). In the remainder of this thesis, however, only parametric simulation studies and real-data studies are discussed explicitly. Still, statements about these two types of studies can generally be understood to apply to the corresponding elements of semi-parametric simulation studies.

The choice of data type typically depends on how performance is to be assessed and on the population of DGMs the study aims to reflect. With respect to performance assessment, parametric simulation studies have clear advantages: each DGM is fully specified and known to the researchers, so the target(s) of interest are either analytically available or can be reasonably approximated (Boulesteix, Groenwold, et al., 2020; Friedrich & Friede, 2024). In addition, performance can be estimated with high precision for each DGM, as, in principle, any desired number of independent datasets can be generated (Boulesteix, Groenwold, et al., 2020). In real-data studies, by contrast, only one realization per DGM is available. As described above, researchers must either use this dataset directly or apply resampling, both of which pose challenges for performance estimation (see Section 3.3.2). Moreover, since the DGM underlying a given real dataset is unknown, the target(s) of interest are often also unknown. This limits the set of usable performance measures (see Section 3.2.3), particularly in tasks such as estimation or hypothesis testing. Exceptions include prediction tasks with labeled data, where the target (i.e. the true outcome values) is known, and datasets from controlled experiments, such as spike-in studies, where known amounts of specific biomolecules are added to real samples (Brooks et al., 2024; Weber et al., 2019).

When it comes to representing the population of DGMs that the benchmark study aims to reflect, both data types present limitations. Parametric simulations can, in principle, be designed to reflect any DGM that is encompassed by the population of interest. In practice, however, this population is typically broadly defined, and only a limited number of DGMs can be implemented due to computational constraints (Friedrich & Friede, 2024). Consequently, researchers must specifiy a subset of DGMs that is sufficiently representa-

[5] This distinction is why this thesis uses $\mathcal{G}$ (DGM of interest) and $\widehat{\mathcal{G}}$ (emulated DGM), while Contribution 4 uses $\mathcal{G}$ (DGM of interest) and $\mathcal{G}^*$ (DGM underlying a real dataset used to construct $\mathcal{G}$).

tive of the population, which is a non-trivial task. Additional challenges arise when the goal is to represent a population of real-world DGMs. In such cases, it is often difficult to specify DGMs that plausibly reflect the data structures and complexity encountered in practice (see, e.g., the reviews by Bono et al. 2017; Fernández-Castilla et al. 2020; Guevara Morel et al. 2022; Langan et al. 2017; Pénichoux et al. 2015; Welvaert and Rosseel 2014). While this issue can partly be mitigated by the above-discussed option of constructing DGMs based on real datasets, this raises the problem of identifying appropriate datasets (see below).

In contrast to parametric simulations, evaluations using real datasets are not suitable for studying extreme or otherwise specific DGMs, as researchers cannot control the DGM underlying a given dataset. Although employing real data has clear advantages when the goal is to reflect a population of real-world DGMs, identifying suitable datasets remains difficult, as each dataset's underlying DGM is largely unknown and its representativeness therefore uncertain. Combined with the generally limited access to real datasets, there are concerns that the set of datasets (and the corresponding set of DGMs) selected for real-data studies often constitutes a convenience rather than a representative sample (Boulesteix, Hable, et al., 2015; Friedrich & Friede, 2024; Strobl & Leisch, 2024; Van Mechelen et al., 2023).

Set of DGMs Once the data type has been specified, a concrete set of L DGMs must be defined for which performance will be evaluated. As described above, this is done either directly in parametric simulation studies or implicitly through the selection of real datasets in real-data studies. As noted in the discussion on data types, the DGM population of interest is typically defined in broad terms, with only a few characteristics precisely constrained. For instance, it could be specified as all DGMs that generate data from clinical trial settings with two treatment groups, a continuous outcome, and a violation of a specific assumption (see Contribution 4 for this and further examples). Since the full population usually cannot be covered, a representative subset of DGMs must be selected. Practical constraints, as previously mentioned, mainly arise from limited computational resources in parametric simulation studies and from restricted dataset availability in real-data studies. In the latter case, researchers may rely on datasets already available to them or obtain datasets from public repositories, where selection must consider not only representativeness with respect to the DGM population but also quality-related aspects such as sufficient documentation. A concrete example of a public repository, also used in Contribution 2, Contribution 3, and Contribution 4, is The Cancer Genome Atlas (TCGA, https://www.cancer.gov/tcga), which offers access to various types of omics data.

3.2.3 Performance assessment

In addition to the methods of interest and the population of DGMs, the research question also involves an *evaluation perspective* ("[evaluation perspective]" in Table 1 and Section 3.1), which can be interpreted as encompassing all aspects that define what constitutes a relevant evaluation of the considered methods. This includes one or more *evaluation criteria*, describing the facets of method behavior the study intends to assess. Other aspects of the evaluation perspective that relate to how performance results are analyzed will be discussed separately in Section 3.3.[6] To operationalize the evaluation criteria in the study design, researchers must define one or more corresponding *performance measures* per criterion, resulting in a total of $\mathcal{P}_1, \ldots, \mathcal{P}_P$ performance measures. In addition, they must specify the procedure used to generate the sampled datasets, which depends not only on the selected performance measures but also on the method implementations and DGMs (Sections 3.2.1 and 3.2.2). Both components of performance assessment are reviewed in the remainder of this section.

Performance measures Performance measures quantify how well a method recovers a target of interest, or describe characteristics of the method that are relevant to this recovery. In this thesis, a performance measure is interpreted as a function $\mathcal{P}(\mathcal{G}, \mathcal{A})$ of a DGM $\mathcal{G}$ and a method implementation $\mathcal{A}$, implying that there is a true performance value for each combination of method implementation and DGM, as also suggested by Morris et al. (2019).[7]

In the following, some examples of evaluation criteria and corresponding performance measures are given, based on benchmarking guidelines and related literature by Bokulich et al. (2020), Mandl et al. (2025), Van Mechelen et al. (2023), Morris et al. (2019), Weber et al. (2019), and Xie et al. (2021). A commonly used evaluation criterion is *accuracy*, with corresponding performance measures including bias (for estimation tasks), root mean squared error (for estimation or prediction tasks), or the Adjusted Rand Index (for clustering tasks). Importantly, these performance measures, like many others, require knowledge of the target. In contrast, for evaluation criteria such as *stability*, corresponding performance measures such as the variance of an estimator (in the context of estimation) can be assessed without access to the target. Another example is *agreement with reference*

[6] In Contribution 3, the term evaluation criteria is used broadly to include both the choice of performance measures and aspects related to the analysis of results. In this thesis, a more nuanced terminology is adopted: the term evaluation perspective refers to both components, whereas evaluation criteria is used exclusively for the aspects of method performance that are to be quantified by performance measures.

[7] In Hothorn et al. (2005), performance measures are defined conditional on a sampled dataset drawn from a DGM $\mathcal{G}$, making them random variables whose distribution is determined by $\mathcal{G}$, and for which different distributional characteristics (e.g., expectation, variance) can be assessed (a similar view is taken in Bischl et al., 2023). This definition remains compatible with the one used in this thesis, where the true performance value is identified directly with such a characteristic. However, some performance measures listed in Table 6 in Morris et al. (2019) would, under the view of Hothorn et al. (2005), correspond to different characteristics of the same random variable rather than distinct performance measures.

methods, where the method's output is compared to that of a predefined reference method rather than to a known truth (e.g., the Jaccard index can be used as a performance measure in this context). Performance measures for efficiency-oriented evaluation criteria, such as total runtime (for *computational efficiency*) or the number of model parameters (for *model simplicity*), not only require no information about the target, but are also typically independent of the target of interest. Finally, while most performance measures are quantitative, certain evaluation criteria, such as *biological plausibility*, are often assessed using qualitative performance measures, such as expert judgment of a method's consistency with prior findings in the literature.

As defined above, performance measures in this thesis refer to assessments made for a single DGM and method implementation. However, some evaluation criteria require the use of higher-level performance measures, obtained by summarizing individual assessments across varying DGMs or method implementations. For example, the criterion *scalability* (Bokulich et al., 2020; Weber et al., 2019) involves repeatedly assessing performance (e.g., runtime) across DGMs of increasing complexity, with the results summarized into a higher-level performance measure (e.g., the slope of runtime increase over DGM complexity). Similarly, the criterion *stability across hyperparameter configurations* requires performance to be evaluated across different hyperparameter configurations, each corresponding to a distinct method implementation (Van Mechelen et al., 2023). In principle, this can be seen as a special case of the broader criterion *stability across expertise levels*, which would involve repeated performance assessment across different implementations of the same method by different researchers, but is rarely applied in practice, as it is challenging to realize within a single benchmark study (Boulesteix et al., 2017; Duin, 1996). However, it may be approximated using qualitative performance measures such as user-friendliness or documentation quality (Weber et al., 2019), which serve as proxies for the potential impact of expertise on performance.

In the remainder of this section and in Section 3.3, the focus is on quantitative performance measures. However, statements that do not assume numerical values also apply to qualitative performance measures. Furthermore, it will be assumed that no higher-level performance measures are considered in the study design, although the discussed concepts naturally extend to this setting.

Generation of sampled datasets As already described in Section 3.2.2, to enable performance estimation for a given DGM $\mathcal{G}$ and method implementation $\mathcal{A}$, one or more *sampled datasets* are required, which serve as inputs to the method implementation. Ideally, a large number of such datasets are drawn independently from $\mathcal{G}$, which, however, is only feasible in parametric simulation studies. In real-data studies, where only a single realization from the DGM is available, performance estimation requires either analyzing that dataset directly ($n_{\text{rep}} = 1$) or applying a resampling scheme to define an emulated

DGM, $\widehat{\mathcal{G}}$, from which sampled datasets can be drawn (Hothorn et al., 2005). Although this technically will yield an estimate under $\widehat{\mathcal{G}}$, it is intended to approximate the true performance value $\mathcal{P}(\mathcal{G}, \mathcal{A})$ for the original DGM $\mathcal{G}$.

In parametric simulation studies, formulas are available to derive n_{rep} to achieve a desired standard error (also referred to as *Monte Carlo standard error*) for common performance measures (Morris et al., 2019; Siepe et al., 2024). However, the choice of n_{rep} is typically constrained by available computational resources. In real-data studies, when the dataset $\mathcal{D}$ is not analyzed directly (i.e. $n_{\text{rep}} > 1$), both n_{rep} and the resampling scheme must be specified. While various schemes are available, it is often unclear which scheme leads to an emulated DGM $\widehat{\mathcal{G}}$ that best approximates the true DGM $\mathcal{G}$ (Schreck et al., 2024; Stolte et al., 2024).

Generating sampled datasets based on a single real dataset $\mathcal{D}$ poses particular challenges in the context of prediction, where each sampled dataset must itself comprise two distinct parts: a *training dataset* for fitting the method implementation and a *test dataset* for evaluating its performance. Otherwise, performance estimates may be optimistically biased due to *overfitting* (see, e.g., Efron, 1986; Hastie et al., 2009; Kuhn & Johnson, 2013). Here, a commonly used approach is k-fold cross-validation (see, e.g., Hastie et al., 2009), where $\mathcal{D}$ is partitioned into k folds, each used once as a test dataset while the remaining folds serve as the training dataset. Importantly, even when a clear separation between training and test datasets is maintained, performance estimates may still be overly optimistic if observations from the test dataset were already used to tune the hyperparameters of $\mathcal{A}$. This can be avoided by not evaluating $\mathcal{A}$ as a method implementation with fixed hyperparameters, but instead specifying it as a self-tuning method implementation (see Section 3.2.1), implying that tuning is performed exclusively on the corresponding training dataset. However, ensuring this separation typically requires a *nested resampling* scheme, which is computationally expensive. While the issue of tuning-induced overoptimism can arise in other tasks as well, it is most commonly discussed in the context of prediction (e.g., Bischl et al., 2023). For a detailed discussion of resampling and hyperparameter tuning for prediction tasks, see Contribution 1.

3.3 Analysis

After the study has been executed, the result is a collection of raw performance values, with one value intended for each combination of performance measure, method implementation, DGM, and sampled dataset (i.e. $P \times M \times L \times n_{\text{rep}}$ values, assuming n_{rep} is the same for each DGM). The subsequent analysis concerns how these results are further processed and examined. Specifically, this includes how missing performance values are handled (Section 3.3.1) and how method rankings, which ultimately form the basis for performance conclusions, are derived (Section 3.3.2).

3.3.1 Handling of missing performance values

Once the raw performance values are obtained, the first step is to check whether all intended values are present. If not, this is referred to as *missing* or *undefined performance values*. The latter term is suggested by Wünsch et al. (2025) to emphasize that such values often do not exist, rather than merely being unobserved. However, the term missing remains common (e.g., Pawel et al., 2025) and is used here, consistent with Contribution 2. Pawel et al. (2025) distinguish three types of missingness. The first is *DGM missingness*, which occurs when one or more sampled datasets are invalid (e.g., datasets containing only one class when predicting class labels). The second is *performance missingness*, where the method produces a valid output but the performance value is undefined (e.g., a calibration slope cannot be computed due to constant predictions). The third and most complex is *method missingness*, where a method returns an invalid output despite being applied to a valid dataset. The causes of method missingness are often unclear, which is a natural consequence given that the considered methods are not fully understood and are therefore investigated in the benchmark study. According to Wünsch et al. (2025), common manifestations of method missingness (rather than explicit causes) include computational errors (e.g., non-convergence), memory issues, or runtime failures.

In addition to identifying reasons for missingness, researchers must decide how to proceed with the analysis. This depends not only on the suspected cause of missingness and the already chosen study design but, as with all decisions discussed in Sections 3.2 and 3.3, also on the specified research question, in particular the evaluation perspective. In the context of missing performance values, relevant considerations implied by the evaluation perspective include whether these cases should be handled in a way that reflects how applied users would typically proceed, or in a way that prioritizes aspects of interest to method developers.

Provided the missingness is not attributable to a trivial implementation error (e.g., a typo in the code), two general approaches can be distinguished, as discussed by Pawel et al. (2025) and Wünsch et al. (2025). In the first approach, the results matrix is modified without rerunning the study. This may involve deleting individual values, entire DGMs, methods, or metrics, or imputing missing values (e.g., using the worst possible value, see also Contribution 2). The second approach involves adapting the study design and re-executing parts of the study. In the case of method missingness, researchers may slightly alter the method implementation (e.g., by changing hyperparameters, using different software, or allocating more resources) or replace the method entirely (e.g., with a baseline). This adaptation may be applied to the entire study or only in instances of missingness (in which case it is termed a *fallback strategy*), and can be specified prior to the initial execution or decided post hoc after analyzing the missingness. Overall, the handling of

missing values is a delicate matter, as it may complicate the interpretation of results; for detailed considerations, see Pawel et al. (2025) and Wünsch et al. (2025).

3.3.2 Derivation of method rankings

Once missing performance values have been addressed, the next step is to determine how the raw performance values should be analyzed and summarized. As with the components of the benchmark study discussed in Sections 3.2.3 and 3.3.1, this step can be understood as part of the operationalization of the evaluation perspective specified in the research question, with different evaluation perspectives implying different types of analyses. A first distinction relates to whether the focus lies on absolute or relative performance. Most benchmark studies are concerned with the latter, comparing methods against each other and deriving some form of ranking (formally a strict or weak order; see Mersmann et al., 2015). This comparative perspective will be assumed in the following.

Another distinction, also motivated by the evaluation perspective, is whether the benchmark study aims to draw *unconditional* or *conditional* conclusions (Van Mechelen et al., 2023). In the former case, the aim is to derive general-purpose recommendations, such as identifying a "default method" that performs well in the absence of problem-specific knowledge (Mersmann et al., 2015), which requires analysis strategies that yield a single overall ranking of methods across all DGMs and performance measures. In contrast, aiming for conditional conclusions reflects the intention to capture more nuanced aspects of method performance, for example to identify decision rules that can guide method selection (Hand, 2006; Van Mechelen et al., 2023). Strobl and Leisch (2024) even argue that conditional conclusions are generally preferable, particularly when the population of DGMs is very heterogeneous. Corresponding analysis strategies yield separate method rankings within subgroups defined by DGMs or performance measures and may also investigate how the rankings vary across subgroups.

In the following, possible analysis strategies to obtain both types of conclusions are reviewed. While the strategies presented below follow a formal structure, they are in practice often applied in a more informal manner, without explicit reference to specific procedures or terms. Note that the form in which the results of the analysis are presented (i.e. figures, tables, and textual summaries) also plays an important role but is not covered here, as it is highly context dependent.

Unconditional analysis If only a single performance measure is considered, a possible approach to obtain an overall ranking is to first compute a point estimate for each DGM and method implementation, aggregate the results across DGMs (e.g., using the mean or median), and rank the method implementations accordingly. However, this is only suitable for performance measures that are comparable across DGMs (e.g., this applies to dimensionless measures like AUC), and even then, rankings based on aggregated

point estimates ignore the relative ordering within each DGM and may be distorted by skewed performance distributions across DGMs (Hornik & Meyer, 2007; Rofin et al., 2023; Wainer, 2023). A common alternative is to still use the point estimates but first assign ranks within each DGM, and then aggregate these ranks across DGMs, for example using the mean rank or counting wins (Dehghani et al., 2021; Demšar, 2006; see Contribution 2 for an application). Such procedures can be viewed as simple examples of *consensus ranking* (Hornik & Meyer, 2007), which describes the process of deriving an overall ranking from individual rankings (or more generally, a set of pairwise relations), with roots in disciplines such as social choice theory (Rofin et al., 2023). The fact that some consensus ranking procedures require only pairwise relations rather than full rankings allows the derivation of an overall ranking based on statistical tests performed for each DGM (rather than the point estimates per DGM). By stating a relation such as $\mathcal{A}_1 > \mathcal{A}_2$ only when $\mathcal{A}_1$ is significantly better than $\mathcal{A}_2$, these procedures allow to incorporate the uncertainty associated with estimating performance on a given DGM (Eugster et al., 2012; Mersmann et al., 2015). Challenges in this respect include the fact that statistical significance does not imply practical relevance (Hothorn et al., 2005; Van Mechelen et al., 2023; Wagstaff, 2012; Wainer, 2023), additional variability in case of stochastic method implementations (Bouthillier et al., 2021) and, for real-data studies, that the sampled datasets are not independent (in the context of prediction see, e.g., Dietterich, 1998 for a test that addresses this issue and Schulz-Kümpel et al., 2025 for a general discussion). When multiple performance measures are considered, an overall ranking can be obtained by first applying consensus ranking within each DGM across performance measures (optionally incorporating weights), and then across DGMs (Eugster et al., 2012).

While applying pairwise statistical tests per DGM can address estimation uncertainty within DGMs (*first-level uncertainty*), all approaches discussed so far neglect the fact that the DGMs themselves represent only a sample from a larger population of DGMs on which the benchmark study aims to draw conclusions (*second-level uncertainty*; see also Boulesteix, Hable, et al., 2015; Wainer, 2023). For a single performance measure, second-level uncertainty can be addressed by performing statistical tests where DGMs are treated as sampling units (e.g., the Friedman test followed by a post-hoc Nemenyi test, as proposed by Demšar, 2006). However, these tests rely on point estimates (only valid for comparable performance measures) or ranks derived from them, and therefore do not account for first-level uncertainty (Dietterich, 1998). To incorporate both levels of uncertainty, a possible approach is to derive consensus rankings based on statistical tests as discussed above, but specifically apply a probabilistic consensus procedure such as the Bradley–Terry model (Bradley & Terry, 1952). In this model, the estimated ability parameters for each method implementation are accompanied by uncertainty estimates that reflect variation across sampled DGMs, corresponding to second-level uncertainty. A Bayesian extension of the Bradley–Terry model has been proposed by Wainer (2023),

which addresses limitations of frequentist significance testing, e.g., by allowing for the specification of regions of practical equivalence. For comparable performance measures, another approach is to model the performance values on each sampled dataset for each DGM and method implementation as observations using linear mixed models, as proposed by Eugster et al. (2012). Here, the overall method performance is modeled as a fixed effect, while various sources of variation are captured through random effects, including one for DGMs to reflect that they are sampled from a larger population of DGMs. Similar to the previous approaches, the model output allows for statistical inference on the method-specific performance estimates.
The inference results from the presented approaches that account for second-level uncertainty can be used in two ways: either to supplement a consensus ranking derived without considering this uncertainty, or to directly construct a ranking based on the pairwise comparisons obtained from these results. The latter is discussed by Eugster et al. (2012), as it enables subsequent aggregation across multiple performance measures.

Conditional analysis As outlined earlier, conducting a conditional analysis implies deriving method rankings within subgroups defined by performance measures or DGMs. In principle, any of the ranking procedures described above can be applied separately within each specified subgroup. For DGMs, however, finding meaningful subgroups is often nontrivial. These are typically constructed based on DGM characteristics that are expected to influence method performance, so that rankings obtained within a subgroup are informative for all DGMs in the population sharing those characteristics. In simulation studies, suitable DGM characteristics are often directly available, as they correspond to components of the DGM that are systematically varied (e.g., parameters such as effect size; Eugster et al., 2014). In real-data studies, relevant characteristics are more difficult to define, both because there are many possible options, and because many DGM characteristics can only be estimated from the real dataset available (see also Contribution 4 for a discussion on this differentiation). Still, commonly used characteristics include aspects of data dimensionality (e.g., number of variables or observations), variable types, and basic summary statistics (Brombacher et al., 2025; Eugster et al., 2014; Mersmann et al., 2015; Oreski et al., 2017). To investigate which characteristics are associated with differences in method rankings, Eugster et al. (2014) and Oreski et al. (2017) have proposed using decision trees with DGM characteristics as splitting variables (with these characteristics derived from real datasets in the context of their studies). Finally, if no such characteristics can or should be extracted, an alternative approach proposed by Kandanaarachchi and Smith-Miles (2023) for real-data studies is to use an inverted item response theory model, which originates from educational psychometrics. In this approach, datasets are treated as participants and method implementations as test items, allowing derivation of both dataset difficulty and interpretable method properties such as a method's difficulty

limit, which describes how difficult a dataset can be while the method is still expected to perform well. Although this approach does not involve explicit subgrouping, it can still be interpreted as a form of conditional analysis, as it analyzes differences in method performance in a structured way rather than aggregating results into an overall ranking.

4 Optimistic bias in benchmark studies

This section provides a structured discussion of optimistic bias in benchmark studies, covering the structural risk factors that enable it (Section 4.1), its core mechanism and manifestations (Section 4.2), challenges in assessing its presence along with related conceptual considerations (Section 4.3), and possible solutions (Section 4.4).

4.1 Structural risk factors

To understand how optimistic bias can arise in benchmark studies, a first step is to examine two structural risk factors: the non-neutrality of researchers and the available researcher degrees of freedom.

(Non-)neutrality As described in Section 2.1, the benchmark studies considered in this thesis are either (i) studies that are part of methodological papers introducing a new method, or (ii) standalone studies that explicitly focus on the comparison of existing methods.

For studies of type (i), the researchers are clearly not neutral towards their newly proposed method, as they typically hope to show that it is, in some way, superior to existing methods. Developing a new method usually involves substantial time and effort, which makes researchers reluctant to publish results that might be seen as a "failure" and potentially harm their scientific reputation. Moreover, demonstrating superiority is still often an (implicit) requirement for publication in journals and at conferences (Boulesteix, Stierle, & Hapfelmeier, 2015; Ferrari Dacrema et al., 2021; Hullman et al., 2022). An acknowledgment of this issue is rare, but Makino et al. (2023) explicitly address it in a comment on their preceding publication introducing MBCdeg (a method for differential expression analysis, originally presented by Osabe et al., 2021 and investigated in Contribution 3). They state: "Like other method's papers, we claimed the potential high performance of MBCdeg. This is simply because a new method needs to have some merits in order to be accepted for publishing in most cases" (p. 3 of the preprint; this statement was removed in the final published version, see Makino et al., 2024).

In contrast to studies of type (i), for those of type (ii), generally a higher degree of neutrality can be expected. However, it is not uncommon for researchers conducting such benchmark studies to have been involved in the development of some of the evaluated methods, in which case they cannot be considered fully neutral either. Boulesteix et al. (2013)

addressed this issue by introducing the concept of a *neutral comparison study* (referred to here as a *neutral benchmark study*). According to their definition, neutrality requires not only that the study focuses explicitly on method comparison but also that the researchers are "reasonably neutral" (p. 8; the definition also mentions that the study design should be chosen in a rational way, although this can typically be expected if the authors are reasonably neutral). While Boulesteix et al. (2013) do not comprehensively define when this requirement is fulfilled, they explicitly exclude cases in which the researchers have contributed to the development of any of the evaluated methods. Nonetheless, even in the absence of such direct involvement, researchers may still hold implicit preferences or prior beliefs, for example, be particularly convinced of (or skeptical about) a specific method or class of methods. More generally, one can argue that different degrees of neutrality exist, but complete neutrality is difficult, if not impossible, to achieve. As a consequence, a benchmark study is rarely conducted without some expectation or hope regarding its conclusions.

Researcher degrees of freedom Originally introduced in the context of applied research (Simmons et al., 2011), researcher degrees of freedom (RDFs) in benchmark studies can be broadly defined as the set of choices open to researchers in the design and analysis of the study given a specific research question, where the decisions made for these choices typically affect the results of the study.

This definition, in the first place, covers all choices related to the components discussed in Sections 3.2 and 3.3. As emphasized there, these choices should, in addition to practical considerations, be made such that they constitute a representative operationalization of the research question formulated in the study's aim. However, in addition to requiring that researchers carefully reflect on the research question and its implied generalization (which may not always be the case in practice), this also presupposes that the research question can be clearly and unambiguously specified. This is already difficult because the elements constituting the research question, namely [methods of interest], [evaluation perspective], and [population of DGMs] (see Table 1 and Section 3.1), are hard to define precisely. For the population of DGMs, this challenge has been discussed frequently in the literature (e.g., Boulesteix, Hable, et al., 2015; Hullman et al., 2022; Strobl & Leisch, 2024), but it has also been addressed to some extent for the methods of interest (e.g., Hand, 2006). In addition, the evaluation perspective, as it is understood in this thesis as a broad concept encompassing diverse considerations related to method assessment (see Sections 3.2.3 and 3.3), is also difficult to specify comprehensively. Given these conceptual difficulties, researchers have substantial freedom in making the required design and analysis decisions, often with many options available for each RDF. In addition, there is often no requirement to justify their decisions, and no consensus in the field on what constitutes an appropriate choice. For example, in real-data studies, the selection of datasets

typically involves a vast number of possible options, and random sampling, which could help ensure representativeness with respect to the population of DGMs, is difficult to implement in practice (Boulesteix, Hable, et al., 2015). Moreover, the criteria guiding this selection are often not clearly reported (Boulesteix et al., 2017; Ferrari Dacrema et al., 2021; Macià et al., 2013), and while it is generally agreed that using only a single dataset is insufficient (e.g., Kreutz, 2019), there is no consensus on what constitutes an adequate sample size (although specific proposals exist, such as the framework by Boulesteix, Hable, et al., 2015 for supervised machine learning).

When discussing RDFs associated with benchmark studies, another important yet often overlooked choice is the random seed used to initialize the pseudo-random number generator (Gundersen et al., 2023). Many components of benchmark studies involve stochastic elements, such as the generation of sampled datasets or the use of stochastic method implementations (see Section 3.2.1), which implies that executing the same benchmark code multiple times with different seeds may yield different results (see, e.g., Henderson et al., 2018; Picard, 2023 for empirical illustrations). Accordingly, although the choice of seed is not linked to the study's research question and may, contrary to good practice, not even be explicitly specified by the researchers (in which case it is set by the system), it still constitutes a distinct and potentially impactful RDF.

Finally, although RDFs are typically understood in line with the initial definition above as the set of choices conditional on a given research question, the research question itself can also be regarded as a special type of RDF. Setting aside the challenges associated with precisely specifying the research question, its formulation corresponds to a fundamental choice: While changing the research question does not alter the numerical results of the benchmark study (assuming all other components are fixed), it does affect how the results are interpreted and, consequently, the conclusions drawn (similar to hypothesis testing in applied research, where different formulations of the null and alternative hypotheses may leave the observed test results unchanged yet still lead to different interpretations). In this broader sense, the definition of RDFs can be extended to include the specification of the research question itself.

4.2 Mechanism and manifestations

Having considered the structural risk factors that create the potential for optimistic bias, the next step is to examine how it can be introduced. This includes the underlying core mechanism as well as its manifestations across different RDFs.

Definition of optimistic bias If researchers conducting a benchmark study use the available RDFs in ways that favor their desired outcome, this can introduce *optimistic bias* (or equivalently, lead to *over-optimistic conclusions*). Although the literature discusses optimistic bias in benchmark studies (see below), there is no unified definition of the term.

In this thesis, optimistic bias is defined as a systematic deviation of conclusions from what (hypothetical) neutral researchers would have obtained, given a defined research question, in the direction of the original researchers' hopes or expectations. Under this definition, optimistic bias undermines both methodological research, by distorting evidence about method performance, and applied research, by promoting potentially misleading recommendations for method use.

Mechanism of optimistic bias While there are various ways in which optimistic bias can be introduced in benchmark studies, its core mechanism involves two key elements beyond the presence of RDFs: *data leakage* and *selective reporting*.

The term data leakage originates from applied prediction modeling, where it refers to information from the test data improperly influencing model development (Kapoor & Narayanan, 2023). A textbook example of this, implicitly addressed in Section 3.2.3, is using the same dataset for both training and testing. This prevents the evaluation from detecting overfitting and leads to an optimistically biased estimate of prediction error (see Contribution 1 for a detailed discussion of this and other forms of data leakage in the prediction context). In benchmark studies, where the focus is on comparing methods rather than evaluating a specific model, data leakage occurs when information about the study's results or conclusions improperly influences how decisions are made. Specifically, it arises when researchers have some form of knowledge about which options to select for the available RDFs that favor their desired outcome (cf. Gundersen et al., 2023; Lange et al., 2025). This creates results that are unlikely to hold if alternative reasonable decisions were made, analogous to evaluating a prediction model on a different dataset. A typical example is post-hoc modification of study components after inspecting preliminary results, which allows researchers to evaluate different options for each RDF and learn which selections yield favorable outcomes. However, even without explicit evaluations, data leakage can occur subtly when researchers select specific options for the available RDFs based on expectations about which will be most favorable.

For this improper advantage to result in optimistic bias, the influence of data leakage must be concealed or ignored, which occurs through selective reporting. In the literature, selective reporting is commonly understood as presenting only a subset of the generated results (e.g., Buchka et al., 2021; Lohmann et al., 2022; Pawel et al., 2024; a similar understanding is also partly adopted in the contributions of this thesis). This corresponds to the concealment of data leakage through post-hoc modifications. However, as described above, data leakage may also occur without explicit evaluations. To describe optimistic bias coherently as arising through data leakage and selective reporting, the term selective reporting must therefore be extended to also include practices that obscure the presence of a priori leakage. While selective reporting in the narrower sense is essentially binary (results are either included or not), the concealment of a priori leakage may occur in

more nuanced ways, reflected in how difficult it is for readers to recognize its presence. In practice, this may take the form of presenting decisions as purely methodological or practical, without acknowledging that they were guided by expectations about favorable outcomes, or of omitting a description of the decisions altogether, thereby implying that there was nothing to report.

In conclusion, when combined with flexibility in RDFs, data leakage and selective reporting may lead to conclusions aligned with researchers' hopes and expectations, systematically deviating from what a neutral study would have produced. In applied research, corresponding practices are considered forms of *questionable research practices* (John et al., 2012).

Manifestations across RDFs In the literature, various forms of the previously described core mechanism have been identified and illustrated for specific RDFs. In the following, these examples are reviewed with a focus on how data leakage occurs and is exploited; the selective reporting step can then be understood as the subsequent concealment of this influence.

Regarding method implementation, typical forms of data leakage through post-hoc modification include changing hyperparameter values or tuning strategies after seeing the results, or even adding or removing entire methods from the comparison. These practices have been empirically illustrated by Jelizarow et al. (2010), Pawel et al. (2024), and Ullmann et al. (2023), who specifically focus on hypothetical benchmark studies of newly proposed methods. In this context, researchers have even more flexibility, as they may also adjust hidden hyperparameters, i.e. aspects of the method for which different variants are considered during initial development but which are later intended to remain fixed for method users (Ullmann et al., 2023). Another form of data leakage in method implementation arises when researchers decide a priori to apply more extensive hyperparameter tuning to the favored method(s) while leaving competing methods at their default values (Ferrari Dacrema et al., 2021; Weber et al., 2019). Data leakage here is introduced because the researchers can expect the tuned method(s) to outperform its competitors.[8] Another example of this more subtle form of data leakage occurs in parametric simulation studies, when only the hyperparameters of the favored method(s) are specified using information about the DGM (Kreutz et al., 2020; Ullmann et al., 2023).

Regarding the selection of the set of considered DGMs, either directly in parametric simulation studies or implicitly through the selection of real datasets in real-data studies, post-hoc modification has been frequently discussed in the literature. Empirical illustrations of this practice have been provided by Pawel et al. (2024) and Ullmann et al. (2023)

[8]Importantly, the central point in this context is not the unequal treatment of methods as such, which may also reflect differences in researchers' expertise across methods, but the unequal effort that researchers invest in implementing the methods given their expertise (see Section 4.3 for a discussion of the interplay between expertise and optimistic bias).

for the simulation case, and by Jelizarow et al. (2010), Keogh and Kasetty (2003), Macià et al. (2013), Ullmann et al. (2023), Yousefi et al. (2010), as well as in Contribution 2, for the real-data case. In parametric simulation studies, there are also well-known examples of data leakage concerning the a priori specification of DGMs in a way that creates advantageous conditions for the favored method(s). This includes cases where the DGM structure is closely aligned with the assumptions of the favored method(s), or where the favored method(s) is directly used to generate the sampled datasets (Brooks et al., 2024; Smith et al., 2022).

For the remaining design and analysis RDFs, the literature has given comparatively less attention to specific forms of data leakage. One of the more frequently addressed cases is the post-hoc selection of performance measures (Jelizarow et al., 2010; Norel et al., 2011; Ullmann et al., 2023), which is empirically illustrated in Contribution 2. Contribution 2 also investigates the post-hoc selection of the imputation method for missing values and the method of performance aggregation.

As discussed in Section 4.1, in addition to RDFs related to design and analysis, there are also RDFs concerning the random seed and the formulation of the research question. For random seeds, it is generally not possible to know a priori which seed will yield favorable results. However, by modifying the seed post-hoc, researchers may re-execute the code with different seeds until a "lucky seed" is found; see Picard (2023) and Ullmann et al. (2023) for empirical illustrations. The research question itself can also be (re-)formulated post-hoc based on the results obtained (Lange et al., 2025; Pawel et al., 2024). For example, if a favored method performs well only for a specific subset of DGMs, researchers might not only discard the less favorable DGMs (as discussed above), but also adjust the research question by redefining the target DGM population so that the remaining DGMs appear representative of it. This practice is related to what has been termed HARKing ("hypothesizing after results are known") in applied research (Kerr, 1998).

Psychology behind optimistic bias While the empirical literature illustrating optimistic bias typically involves deliberate and systematic introduction of optimistic bias, in practice, researchers are more likely to engage in corresponding practices unconsciously. This can be explained by the human tendency toward self-deception (Nuzzo, 2015), which leads individuals to interpret ambiguity in ways that support their own hopes or expectations, often without realizing it (Simmons et al., 2011). In the context of benchmark studies, this pitfall is particularly likely to occur when post-hoc decisions arise naturally as part of the research process. For example, journal constraints may require researchers to report only a subset of results, or unexpected failures of certain methods may necessitate implementation changes. Although such adjustments may have legitimate motivations, they also offer convenient justifications for post-hoc modifications that make the results appear more favorable from the researchers' perspective (Boulesteix et al., 2017; Lohmann

et al., 2022; Pawel et al., 2024). A particularly challenging scenario arises in the development of new methods. Prior to systematic evaluation in a formal benchmark study, it is natural for researchers to conduct informal, undocumented testing and refine their method accordingly. However, when components used during method development are carried over into the subsequent benchmark study, corresponding method refinements effectively constitute post-hoc modifications of the method implementation (Boulesteix et al., 2013; Jelizarow et al., 2010; Lange et al., 2025).

Optimistic bias vs. methodological unsoundness When examining how optimistic bias arises, it is important to recognize that it is not the only way in which misleading conclusions can be obtained. Specifically, even if decisions on RDFs are not made to benefit or disadvantage specific methods, they may still be inappropriate in other respects, making the study methodologically unsound without introducing optimistic bias. For example, in parametric simulation studies, researchers may choose overly simplistic DGMs, despite aiming to represent a population of real-world DGMs (Brooks et al., 2024; Weber et al., 2019). In the derivation of method rankings (see Section 3.3.1), common issues include relying solely on point estimates without quantifying uncertainty, or applying statistical tests despite clear violations of their assumptions (Hullman et al., 2022; Pineau et al., 2021; Van Mechelen et al., 2023). While such decisions may lead to misleading conclusions, they do not constitute optimistic bias unless they systematically benefit one or more methods. However, they may increase the likelihood of optimistic bias. For instance, ignoring uncertainty makes it easier to present a method as superior than if it were integrated (see Boulesteix, Stierle, and Hapfelmeier, 2015 for a related discussion and Section 4.4).

4.3 Assessment

After examining how optimistic bias can be introduced into benchmark studies, a key concern is how often it occurs in practice. In the following, challenges in assessing its presence are discussed, and how such assessments relate to reproducibility and replicability.

Challenges in assessment In general, it is not straightforward to assess the presence of optimistic bias. While there may occasionally be hints of an unjustified a priori advantage for specific methods (e.g., when hyperparameters are tuned only for one method), post-hoc modifications are particularly difficult to detect. An alternative strategy is to examine the extent to which the results of benchmark studies align with what researchers are likely to prefer. As discussed in Section 4.1, in benchmark studies published alongside the proposal of a new method, such preferences are typically clear: the new method is expected to outperform existing ones. Indeed, several empirical investigations have found that, in a substantial proportion of cases, the newly proposed method was reported to

be superior to its competitors (Boulesteix et al., 2013; Buchka et al., 2021; Norel et al., 2011; Smith et al., 2022). However, these patterns could also, in principle, reflect genuine scientific progress. Therefore, to determine whether a specific benchmark study is affected by optimistic bias, the definition introduced in Section 4.2 must be used as the basis for assessment. That is, the study's conclusions must be compared to those drawn by neutral researchers, given the same research question. In practice, this would require repeating the study multiple times by different neutral researchers, adopting only the research question formulated by the original study, while reasonably varying all other components. The presence of optimistic bias would then be indicated by a systematic deviation of the original study's conclusions from those of the neutral repetitions.

Along these lines, Buchka et al. (2021) empirically investigate how pairwise comparisons of methods derived from introducing papers hold in later studies (which they frequently do not find confirmed). A different yet small-scale approach is pursued in Contribution 3, where four benchmark studies introducing a new method from two analysis tasks are considered. Each of the four methods is reevaluated using the design and analysis choices of a different benchmark study that proposes another method (and vice versa, which is why this setup is referred to as a cross-design validation experiment). The adoption of all components from another study (as far as possible), combined with the fact that the authors of Contribution 3 were not involved in the development of any of the investigated methods, can be seen as an approximation of neutral researchers. Although three of the four new methods perform worse in this reevaluation, a single instance is clearly not sufficient to provide evidence for a systematic deviation. In practice, each study would need to be reevaluated multiple times to draw reliable conclusions about the presence of optimistic bias.

Even if a sufficient number of (approximately) neutral researchers could be recruited for such an assessment, two fundamental problems remain (see also the discussion in Contribution 3, which specifically considers these problems for newly proposed methods). First, as discussed in Section 4.1, the research question in benchmark studies is difficult to specify precisely. While this already creates challenges for researchers in making design and analysis decisions, it poses an even greater problem for readers, who must rely solely on what is reported, without access to the researchers' implicit knowledge or intentions: not only is it unclear what the benchmark study provides evidence for, but in the specific context of assessing optimistic bias, it is also difficult to judge whether changes in design or analysis components still constitute a valid operationalization of the same research question or instead reflect a shift away from it. For example, given that the method is evaluated in the original study as the specific implementation $\mathcal{A}$, do the conclusions also apply to all other values of each hyperparameter, or only to similar ones, or only to variations in a subset of hyperparameters, such as preprocessing hyperparameters? Similarly, in parametric simulation studies, to what extent can DGM parameters be varied with-

out departing from the intended population of DGMs? As a result, differing conclusions cannot be clearly attributed to optimistic bias or to an implicit change in the research question.

The second fundamental problem in assessing optimistic bias originates from the fact that researcher expertise generally affects method performance (see Section 3.2.1) and, unless robustness to this factor is explicitly evaluated, becomes implicitly tied to the research question by the qualifier "given the methods are applied by a user with specific expertise" (see Section 3.1). Accordingly, even if the first problem of imprecise research question specification were resolved, accurately considering the exact same research question when assessing optimistic bias would still require identifying researchers who are not only neutral toward all methods but also match the original study's expertise profile across all considered methods. Otherwise, differences in method performance between the original and new study might simply reflect differences in expertise rather than bias. In practice, however, identifying suitable researchers for such an assessment is hardly feasible. In addition to being a latent factor that cannot be measured, expertise is often positively correlated with neutrality: the more expertise a researcher has with a method, the more likely they are to view it favorably, and vice versa.[9]

Taken together, these problems make it difficult to definitively determine whether optimistic bias is present in a given benchmark study.

Relation to reproducibility and replicability When discussing the assessment of optimistic bias, it is worthwhile to briefly address the concepts of *reproducibility* and *replicability*, as they are frequently invoked in this context, often implicitly associating bias with failures to reproduce or replicate studies (e.g., Boulesteix, Hoffmann, et al., 2020; Gundersen et al., 2023; Hullman et al., 2022). Given the inconsistent use of the terms reproducibility and replicability in both methodological and applied research (Bouthillier et al., 2019; Plesser, 2018), the following discussion focuses on the aspects most relevant for the scope of this thesis, rather than on terminological distinctions.

Similar to the assessment of optimistic bias, reproducibility and replicability broadly concern the variation of components of an original study, followed by a comparison of the results and conclusions (with a focus on assessments performed by independent researchers, i.e. researchers not involved in the original study). A first central distinction relates to the purpose of such assessments: either to determine how closely the original results can be reproduced based on the provided documentation, or to examine whether

[9]This correlation may help explain why the two are often not treated as separate dimensions. For instance, Boulesteix et al. (2017, p. 8) interpret the "reasonably neutral" requirement for neutral comparison studies by Boulesteix et al. (2013, p. 8) discussed in Section 4.1 as implying that researchers should be "approximately equally experienced" with all methods. While this is a legitimate definition that is also used in Contribution 2, one might argue that unequal expertise alone does not necessarily lead to unfair comparisons, as the study may still be informative, or even more informative, for readers with similarly unbalanced expertise.

deliberate modifications to specific components still lead to the same results or general conclusions. Albertoni et al. (2023, p. 4) describe this distinction as one between "validating the repeatability of the experiment" and "corroborating the scientific hypothesis and theory the experiment aims to support". While in the experiment repeatability setting, researchers must decide which forms of documentation to rely on for reproducing the original results (e.g., textual descriptions only, or a combination of text and code), in the corroboration setting, they must determine which components of the original study to keep fixed and which to vary. Corresponding variations in applied research typically involve using different data while keeping the analysis fixed, using a different analysis while keeping the data fixed, or modifying both (The Turing Way Community, 2025). However, transferring this framework to benchmark studies is not straightforward, since the distinction between data and analysis is not clearly defined in this context. For example, data might refer to the raw performance values, which serve as the input for the analysis of results described in Section 3.3, with different data arising from changes in any component of the study design (Section 3.2). Alternatively, data might be understood more narrowly as the sampled datasets, where different data could result from drawing new samples with a different seed, or even from selecting a different set of DGMs.

Given these and other conceptual difficulties, terminology remains challenging. For orientation, the following provides a brief overview of how terms are typically used in the literature (see Albertoni et al., 2023 for a review that also provides references in which the respective terms are employed). Assessments of experiment repeatability that use all available information are typically described as evaluations of *(computational) reproducibility*. If only written documentation is considered, the terminology becomes less consistent: both *(result) reproducibility* and *(direct) replicability* are commonly employed to describe this case. By contrast, assessments aligned with the corroboration setting, where specific components of the study are intentionally varied, are often referred to as evaluations of *(conceptual) replicability*, *generalizability*, or *robustness*. Here, choice of term generally depends on which components are held fixed or modified, but the terminology remains inconsistent (and it is debatable whether distinct labels are needed for every possible combination).

Reconsidering optimistic bias in this context, the empirical assessment discussed earlier, that is, repeating the benchmark study with the same research question conducted independently by neutral researchers with comparable expertise to the original researchers, can be understood as a corroboration experiment in which only the research question is fixed, while specific requirements regarding neutrality and expertise are imposed (noting that such researcher-related factors are often neglected in discussions of reproducibility, replicability, and related concepts).

4.4 Possible solutions

While optimistic bias can likely never be avoided entirely, it can and should be reduced as much as possible. This section discusses possible solutions by revisiting the main contributing factors outlined earlier: (non-)neutrality, RDFs, data leakage, and selective reporting, with the strategies addressing these factors intended to complement each other. In addition, special consideration is given to benchmark studies that introduce a new method, as well as to the role of other actors in addressing optimistic bias.
Importantly, the discussion assumes no malicious intent. If bias is introduced deliberately, the strategies discussed here are irrelevant, as they are unlikely to be implemented sincerely. Moreover, while the proposed strategies align with general recommendations for good practice in empirical methodological research, they should not be understood as a comprehensive guide to conducting high-quality benchmark studies.

(Non-)neutrality While researchers cannot just set aside their non-neutral positions toward specific methods, it is important to acknowledge and transparently disclose them (Boulesteix et al., 2013; Van Mechelen et al., 2023). Moreover, although neutrality may not be realistically achieved on an individual level, some balance may be attained at the team level by including researchers with differing preferences and perspectives (Siepe et al., 2024).

Researcher degrees of freedom To mitigate optimistic bias at the level of RDFs, efforts should focus on avoiding the often arbitrary and uncontrolled nature of how decisions are made, and on reducing the variability in results that can arise from individual RDFs.
As an initial step, researchers should carefully reflect on the research question and formulate it as precisely as possible, in order to have a well-defined basis for subsequent design and analysis choices. A possible strategy for these choices is to shift the decision from directly selecting a component to specifying one or more criteria for its selection, where each criterion may reflect either the research question or practical considerations. This shift not only makes the decision process more structured and systematic but, if the criteria are reported, also improves transparency for readers and provides an implicit justification for the decisions made. This strategy has been concretely proposed for selecting methods to be considered in the study (Boulesteix et al., 2013; Xie et al., 2021) and for selecting datasets in real-data studies (Boulesteix et al., 2017). A similar shift is made in the context of real-data-based parametric simulations, where parts of the considered DGMs are not specified directly, but researchers instead define a set of real datasets and a procedure to infer these parts (see Section 3.2.2). An extension of this idea, which has received little attention so far, is to select the real datasets themselves based on suitable criteria as well. This approach is proposed in Contribution 4.

To also prevent uncontrolled modifications of study components, researchers can prepare and potentially preregister a study protocol that documents all planned decisions prior to conducting the benchmark study, with any deviations explicitly reported and justified afterward. Where it is not appropriate to prespecify concrete decisions, result-dependent decision rules can be formulated. For example, with regard to handling missing performance values, researchers might define a rule stating that if a method fails in more than a specific proportion of sampled datasets, it will be excluded; otherwise, missing values will be imputed. At present, few templates exist for such protocols. One exception is the framework by Siepe et al. (2024), developed for simulation studies in the context of methodological research in psychology. A general list of essential items to be included in benchmark study protocols is provided by Lange et al. (2025).
As an additional complementary strategy, researchers can establish conditions that reduce the impact of individual RDFs by making benchmark results more robust to small changes in study components and thereby lowering the potential for their misuse. In general, this can be achieved by setting up the study in a more comprehensive way. For example, when drawing sampled datasets, the impact of the random seed on the resulting performance estimates is generally reduced when a large n_{rep} is used. The impact of dataset choice in real-data studies likewise decreases when many datasets are included (see, e.g., the empirical illustrations in Contribution 2). Also, the more performance measures are considered and the more comprehensively the results are analyzed, the more difficult it becomes to exploit these RDFs in a way that favors a desired outcome (as also reflected in the findings of Norel et al., 2011, who found that new methods were less frequently reported as best when more performance measures were used).

Data leakage As discussed in Section 4.2, data leakage can occur either when researchers modify components after inspecting results or when they make a priori decisions based on expected outcomes. While post-hoc leakage can in principle be prevented by prespecifying all decisions (see above), it may still be useful to implement additional protective measures. One such measure, inspired by clinical trials, is *blinding* (Boulesteix et al., 2017). In the context of benchmark studies, this can be achieved by concealing method identities (e.g., relabeling them as A, B, C) once raw performance values are obtained. This ensures that even if post-hoc modifications are made, they cannot be directed toward specific methods. Note that this procedure may require splitting responsibilities within the research team, as certain tasks, such as identifying the source of missing performance values, may still require knowledge of method identities.
To some extent, blinding can also help reduce a priori data leakage. Specifically, in parametric simulation studies, selecting hyperparameters based on DGM knowledge for some methods but not others can be avoided by assigning the specification of DGMs and hyperparameters to separate research teams, with the team responsible for hyperparameters

blinded to the DGM definitions; see Kreutz et al. (2020) for a concrete implementation. Regarding other forms of a priori leakage that may arise in method implementation, researchers should honestly reflect on whether there are any obvious unequal treatments of methods that cannot be justified by differences in expertise across methods, such as applying extensive hyperparameter tuning only to specific methods or allocating unequal computing time (Ferrari Dacrema et al., 2021; Van Mechelen et al., 2023).
Finally, concerning the specification of DGMs in parametric simulation studies, a priori advantages resulting from aligning DGMs with the assumptions of specific methods can be avoided by granting this advantage to all methods, for example by generating separate DGMs based on each method individually (Brooks et al., 2024; Smith et al., 2022).

Selective reporting If, despite all efforts, data leakage may have influenced decisions on RDFs, this should be clearly disclosed to avoid selective reporting. For decisions made a priori but still expected to favor or disadvantage specific methods, researchers should explicitly state which methods may be affected and how. In the case of post-hoc modifications, all previous versions of the corresponding components and their impact on results should be reported. As proposed in Contribution 2, visualizations such as multidimensional unfolding (Borg & Groenen, 2005) can support this by systematically displaying how method rankings vary across different combinations of specific choices. Originally developed in psychometrics, this technique places objects (here, methods) and subjects (here, combinations of other study components) in a typically two-dimensional space, where shorter distances indicate better performance of a method under the corresponding study conditions; see Contribution 2 for details. Note that while such transparent reporting mitigates optimistic bias by preventing both readers and the researchers conducting the study from drawing misleading conclusions, it cannot undo the consequences of compromised RDF decisions. In particular, the ability to adequately address the research question is diminished, and this should be acknowledged as a relevant limitation.
Although ideally all known or suspected cases of RDF decisions affected by data leakage should be reported, some instances may go unnoticed due to the often unconscious nature of the process (see Section 4.2). For this reason, researchers should generally make the entire benchmark study as open and accessible as possible to others. This includes not only clear textual summaries of all design and analysis components, but also all information required to reproduce the results (e.g., code, random seeds, software and hardware specifications, external datasets, intermediate results), as well as, to the extent possible, a specification of the study's research question and the researchers' expertise with each method. Such transparency enables readers to detect possible instances of optimistic bias, conduct tentative empirical assessments of its presence, and reduce avoidable suspicions in this respect. Moreover, it is generally considered good scientific practice and is therefore widely recommended in benchmarking guidelines (e.g., Brooks et al., 2024; Kreutz,

2019; Morris et al., 2019; Van Mechelen et al., 2023; Weber et al., 2019), additionally facilitating study extensions, meta-analyses, and other research efforts that contribute to building cumulative evidence beyond individual studies.

Newly proposed methods When discussing strategies to reduce bias, benchmark studies accompanying newly proposed methods merit particular consideration, as their authors are clearly non-neutral (see Section 4.1) and many of the strategies described above are difficult to implement. Specifically, as noted in Section 4.2, post-hoc modifications of the new method are an inherent part of the development process and are difficult to document in sufficient detail. In light of these challenges, it has been argued in the literature (e.g., Boulesteix, 2013; Boulesteix et al., 2013; Norel et al., 2011; also echoed to some extent in Contribution 2) that benchmark-style evaluations in papers introducing a new method should be regarded as purely illustrative, without allowing conclusions about empirical properties of the method. However, framing them only as illustrations carries its own risks. It may implicitly legitimize lower methodological standards, while readers may still interpret the results as empirical evidence. If a method is presented as well-performing, albeit incorrectly, it can spread quickly, for example through its inclusion as a relevant competitor in the evaluation of other new methods. Such early reputations are often persistent and difficult to revise (Boulesteix, Stierle, & Hapfelmeier, 2015; Henseler et al., 2024). For this reason, even though reducing optimistic bias is particularly challenging in benchmark studies of newly proposed methods, researchers should still make every effort to pursue it.

A possible approach in this context is to build on the idea of the train–test split used in prediction modeling (see Section 3.2.3 and Contribution 1) by conducting an additional evaluation once initial method development is complete. In the benchmarking literature on real-data studies, this has already been proposed, primarily focusing on evaluating the new method on additional real datasets (e.g., Boulesteix, 2009; Jelizarow et al., 2010; Keogh & Kasetty, 2003; Norel et al., 2011). However, it may also be worthwhile to extend this approach to other data types and, more generally, to additional components of benchmark studies, especially competing methods and performance measures. As suggested in Contribution 3, this could be realized by drawing on existing benchmark studies conducted by other research teams, while remaining mindful of possible differences in research questions. Of course, this is only feasible if authors share their materials transparently (see above).

Role of other actors Although the focus of this section is on strategies available to researchers conducting benchmark studies, it is also worthwhile to briefly address the role of other actors, in particular journal editors and reviewers.

A first step is to reduce incentives that may drive researchers toward practices introducing optimistic bias, by avoiding implicit requirements that new methods be shown as superior,

and, more generally, that results be exciting or unexpected. Instead, greater emphasis should be placed on the methodological soundness of the benchmark study as a whole, independent of its outcome (Boulesteix, Stierle, & Hapfelmeier, 2015). The registered reports format (Chambers, 2013) represents a rigorous implementation of this principle. In this format, publication decisions are made before results are known, based on the importance of the research question and the soundness of the study. In a similar vein, the general appreciation and acceptance of standalone benchmark studies evaluating existing methods should be strengthened (Boulesteix et al., 2013; Boulesteix et al., 2018).
Second, editors and reviewers can play a role in detecting optimistic bias. While its presence is generally difficult to assess (see Section 4.3), they can check for potential warning signs, such as consistently superior performance of a new method or benchmark setups with a high risk of bias (e.g., very few DGMs or competitors). They may also require the use of some of the above-mentioned strategies to reduce optimistic bias, such as sharing code, which has already been implemented by some journals (e.g., *Biometrical Journal*; Hofner et al., 2015).
Finally, journals should encourage studies that assess the reproducibility and replicability of existing benchmark studies, while making the intended aim of such assessments explicit (see Section 4.3). In this context, inspiring initiatives include the RepliSims project for simulation studies (Luijken et al., 2024) and the journal *ReScience C* (https://rescience.github.io/), which publishes replication studies in computational science.

5 Summary of the contributions

This section provides a summary of the four contributions on which this thesis is based.

Contribution 1 The first contribution focuses on the often-overlooked role of preprocessing steps in generating and evaluating prediction models based on supervised machine learning. In practice, choices such as how to handle missing values or how to transform and select variables are ubiquitous, yet they are rarely discussed in the literature. The relevance of this issue became apparent to the authors during work on a real-world problem, the prediction of palliative care costs, which motivated Contribution 1.
As a consequence, the optimization of preprocessing decisions is often informal and may not be adequately accounted for when evaluating model performance, which can lead to optimistically biased prediction error estimates. To address this, Contribution 1 formalizes preprocessing choices as hyperparameters that are part of the complete set of hyperparameters of the analysis pipeline (see Section 3.2.1; in the article, the term *learning pipeline* is used in line with the machine learning context). It also explains in detail how optimistically biased prediction error estimates arise through the improper influence of test data on model development, which in this context is commonly referred to as

data leakage (see Section 4.2). Finally, using the palliative care example, the contribution empirically illustrates both appropriate and inappropriate strategies for model generation and evaluation, and argues for a more conscious and transparent handling of all types of hyperparameters.
Although Contribution 1 is the only contribution in this thesis directed at an applied rather than a methodological audience, it is still relevant for the latter. One reason is that the performance attributed to a model in the applied context essentially constitutes the performance of the method that generated it in the benchmark context, which makes the insights from Contribution 1 transferable to methodological research. In addition, by clarifying the notion of data leakage in its original applied sense, Contribution 1 provides a basis for extending this concept to benchmarking.

Contribution 2 The second contribution extends the benchmark study by Herrmann et al. (2021), which compares 13 methods for predicting survival outcomes from multiomic data. The contribution empirically illustrates how results can vary depending on design and analysis choices, specifically the selection of real datasets, performance measures, handling of missing performance values, and the aggregation approach used to derive the method ranking. The last two represent RDFs that have received little attention in the literature (see Section 4.2). Considering all possible combinations of these choices (288 in total), the study shows that almost any method can achieve nearly any rank, a scenario that could easily be exploited to obtain a favorable result. Even when not all combinations are considered but only one choice is optimized at a time, which is more reflective of realistic researcher behavior, most methods still achieve a favorable rank (with optimization directed toward lower ranks). To complement existing strategies for reducing optimistic bias (see Section 4.4), Contribution 2 further proposes the use of multidimensional unfolding, originally developed in psychometrics, as a way to assess the impact of individual decisions and to systematically visualize how method rankings vary. In this context, the technique places methods and study conditions in a two-dimensional space, where shorter distances indicate better performance of a method under the corresponding conditions.

Contribution 3 In contrast to Contribution 2, which primarily considers standalone benchmark studies focusing on the comparison of existing methods, Contribution 3 examines benchmark studies that accompany the proposal of a new method. It investigates the well-known tendency for newly introduced methods to perform best in the benchmark studies presented in their introductory papers but worse in subsequent studies conducted by other researchers. To this end, the contribution introduces a cross-design validation experiment: for a given data analysis task, two methods developed for that task are reevaluated using each other's original benchmark study setup, emulating a subsequent study performed by more neutral researchers. The considered tasks are cancer subtyping

using multiomic data and differential gene expression analysis using RNA-Seq data, each with two methods.
While, similar to Contribution 2, the experiment illustrates the variability of benchmark results and is consistent with the observation of deteriorating performance in subsequent studies (three of the four methods indeed perform worse in the other method's design), even more importantly, it provides insights into the reasons for these performance discrepancies. The first reason, often seen as the most obvious, is the presence of optimistic bias in the original benchmark study, with its mechanism discussed in Section 4.2. In Contribution 3, this is framed as two specific variants ("overfitting of study design to method" and "overfitting of method to study design"), reflecting its focus on newly proposed methods. In addition, Contribution 3 identifies two further, less recognized reasons for performance discrepancies, which are also discussed in Section 4.3 and can generally be characterized as differences in researcher expertise and differences in the research question between the original and subsequent study. In Contribution 3, which specifically considers the new-method context, the latter is described as the possibility that a subsequent study does not align with the *field of application* investigated in the original study (where finding the appropriate field of application of a new method can be regarded as a specific research question, which is tied directly to that method).

Contribution 4 The fourth contribution focuses on parametric simulation studies that use real datasets as a basis for constructing the DGMs under which the methods are evaluated. While this approach is widely used to make DGMs more realistic, it relies on the choice of real datasets, which constitutes an impactful RDF; in practice, however, only one or two datasets are often used, and the rationale for their selection is frequently unclear. As a result, the constructed DGMs are unlikely to be representative of the population of DGMs to which the study intends to generalize, and the dataset choice itself, due to its strong impact and often unsystematic nature, is easily exploitable for obtaining favorable results.
Contribution 4 addresses this issue by formally discussing real-data-based parametric simulations and proposing a more systematic dataset selection procedure using a database and clearly specified eligibility criteria. While the idea of a more systematic dataset selection already exists in the literature, especially in the context of real-data studies (e.g., Boulesteix et al., 2017), this contribution provides a more detailed treatment, including a distinction between different types of eligibility criteria. It also illustrates the proposed approach with two empirical examples: ordinal outcomes in randomized controlled trials and differential gene expression analysis. For comparison, DGMs are additionally constructed either without a real dataset or from a single real dataset, and the results are contrasted with those obtained with a systematic selection of real datasets.

6 Outlook

While this thesis has advanced the discussion on optimistic bias in benchmark studies, several aspects remain open for further investigation.

Addressing further researcher degrees of freedom Both Contribution 1 and Contribution 2 highlight RDFs that have so far received little attention. Nevertheless, a range of other RDFs remain to be explored. For instance, while Contribution 2 considers the choice of descriptive summary statistics to aggregate performance results across DGMs into method rankings, future research could also investigate the choice of statistical tests for this purpose and explore the variability in results. In addition, while Contribution 1 formalizes preprocessing steps as hyperparameters, it could be equally valuable to explicitly formalize postprocessing steps in the same way. Moreover, the empirical illustration of preprocessing hyperparameters in Contribution 1 is primarily directed at applied researchers. Although these insights are also relevant for methodological researchers, a dedicated illustration explicitly adopting the perspective of a benchmark study, similar in style to Contribution 2, would likely raise awareness more directly for this audience.

Improving clarity on researcher expertise As discussed in Sections 3 and 4 and in Contribution 3, researcher expertise is a key factor in interpreting the results of benchmark studies and, consequently, in assessing optimistic bias. While Contribution 3 and earlier work (e.g., Boulesteix et al., 2017; Duin, 1996) highlight the importance of expertise and illustrate how it can affect method performance, concrete ideas for how researchers conducting benchmark studies could transparently communicate their expertise are still lacking. Existing discussions (including Contribution 3) typically distinguish only between "experts" and "non-experts", a dichotomy that is too coarse to provide sufficient clarity. Ideally, the benchmark community would agree on a standardized way of reporting expertise, accompanied by an assessment of how strongly expertise is expected to influence the performance of each method. Of course, developing such a reporting scheme is difficult due to the latent nature of expertise, and any solution will inevitably remain a simplification. Still, an ordinal scale might be a useful starting point, taking into account aspects such as years of experience with the method, diversity of practical use, and depth of theoretical understanding.

Improving clarity on the research question Analogous to expertise, the research question of a benchmark study is highly relevant (and even more central) for interpreting benchmark studies, yet difficult to formulate clearly (see Sections 3 and 4).
With respect to the population of DGMs, Contribution 4 proposes a step toward clarification by advocating the explicit specification of a database and eligibility criteria for selecting real datasets, which not only aims to improve representativeness of the intended

population of DGMs but also serves to clarify this population (importantly, the criteria aimed at representativeness are explicitly distinguished from technical criteria that address issues like data quality; see Contribution 4 for details). However, Contribution 4 only considers real-data-based parametric simulations, and the specific approach it proposes cannot be readily transferred to benchmark studies using other data types, which calls for further work in this direction.

For the remaining elements constituting the research question, namely methods of interest and evaluation perspective, the situation is even less developed. One possible explanation is the asymmetry in recognizing the generalization that inevitably occurs: many researchers acknowledge the step from specific DGMs to a population of DGMs, and this issue has also received attention in the literature (e.g., Boulesteix, Hable, et al., 2015; Herrmann et al., 2024; Strobl and Leisch, 2024). By contrast, the parallel generalization from the concrete method and evaluation setup to the broader notions of methods of interest and evaluation perspective, although likewise occurring in practice, is often not consciously recognized and remains implicit. This is understandable, however, since specifically the concept of an evaluation perspective is even more abstract than that of a population of DGMs (with the terminology itself only introduced in this thesis). Based on this background, an important direction for future research is to further elaborate and refine the definition and scope of the evaluation perspective, which could then provide the basis for developing more structured ways to specify it.

Overall, as in the case of expertise, it would be desirable to establish a standardized way of reporting the research question in benchmark studies. A useful point of orientation could be the ADEMP framework (aims, data-generating mechanisms, estimands, methods, and performance measures; Morris et al., 2019) for simulation studies, which illustrates how structured reporting can enhance clarity.

Advancing the proposed strategies to reduce optimistic bias The contributions of this thesis propose additional strategies for reducing optimistic bias alongside existing ones (see Section 4.4). However, these strategies still need to be examined in practice and made more accessible. In particular, Contribution 4 suggests a workflow for constructing DGMs in parametric simulation studies based on a systematically selected set of real datasets, but this workflow remains to be evaluated in actual simulation studies. Moreover, while the general multidimensional unfolding approach employed in Contribution 2 to report variability in results is already implemented in the `R` package `smacof` (de Leeuw & Mair, 2009), the specific visualization used in Contribution 2 is adapted to the benchmarking context, which necessitates additional manual modifications. To enable researchers conducting benchmark studies to generate this adapted visualization more readily, a dedicated `R` package would be valuable, and its utility should be tested with

RDFs beyond those considered in Contribution 2.

Despite these open tasks, this thesis has contributed to increasing awareness of how optimistic bias arises in benchmark studies and to outlining strategies for addressing it. Although optimistic bias can probably never be avoided entirely, recognizing and countering it is essential for ensuring that benchmark studies can genuinely support both applied and methodological research.

References

Agresti, A. (2010). *Analysis of ordinal categorical data* (2nd). John Wiley & Sons. https://doi.org/10.1002/9780470594001

Albertoni, R., Colantonio, S., Skrzypczyński, P., & Stefanowski, J. (2023). Reproducibility of machine learning: Terminology, recommendations and open issues. *arXiv: 2302.12691 [cs.AI]*. https://doi.org/10.48550/arXiv.2302.12691

Baik, B., Yoon, S., & Nam, D. (2020). Benchmarking RNA-seq differential expression analysis methods using spike-in and simulation data. *PLOS ONE*, *15*(4), e0232271. https://doi.org/10.1371/journal.pone.0232271

Bartz, E., Bartz-Beielstein, T., Zaefferer, M., & Mersmann, O. (2023). *Hyperparameter tuning for machine and deep learning with R: A practical guide*. Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5170-1

Binder, M., & Pfisterer, F. (2024). Sequential pipelines. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/sequential_pipelines.html

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

Bokulich, N. A., Ziemski, M., Robeson, M. S., & Kaehler, B. D. (2020). Measuring the microbiome: Best practices for developing and benchmarking microbiomics methods. *Computational and Structural Biotechnology Journal*, *18*, 4048–4062. https://doi.org/10.1016/j.csbj.2020.11.049

Bono, R., Blanca, M. J., Arnau, J., & Gómez-Benito, J. (2017). Non-normal distributions commonly used in health, education, and social sciences: A systematic review. *Frontiers in Psychology*, *8*, 1602. https://doi.org/10.3389/fpsyg.2017.01602

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications* (2nd ed.). Springer.

Boulesteix, A.-L. (2009). Over-optimism in bioinformatics research. *Bioinformatics*, *26*(3), 437–439. https://doi.org/10.1093/bioinformatics/btp648

Boulesteix, A.-L. (2013). On representative and illustrative comparisons with real data in bioinformatics: Response to the letter to the editor by Smith et al. *Bioinformatics*, *29*(20), 2664–2666. https://doi.org/10.1093/bioinformatics/btt458

Boulesteix, A.-L., Binder, H., Abrahamowicz, M., & Sauerbrei, W. (2018). On the necessity and design of studies comparing statistical methods. *Biometrical Journal*, *60*(1), 216–218. https://doi.org/10.1002/bimj.201700129

Boulesteix, A.-L., Groenwold, R. H. H., Abrahamowicz, M., Binder, H., Briel, M., Hornung, R., Morris, T. P., Rahnenführer, J., & Sauerbrei, W. (2020). Introduction to statistical simulations in health research. *BMJ Open*, *10*(12), e039921. https://doi.org/10.1136/bmjopen-2020-039921

Boulesteix, A.-L., Hable, R., Lauer, S., & Eugster, M. J. A. (2015). A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, *69*(3), 201–212. https://doi.org/10.1080/00031305.2015.1005128

Boulesteix, A.-L., Hoffmann, S., Charlton, A., & Seibold, H. (2020). A replication crisis in methodological research? *Significance*, *17*(5), 18–21. https://doi.org/10.1111/1740-9713.01444

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. A. (2013). A plea for neutral comparison studies in computational sciences. *PLoS ONE*, *8*(4), e61562. https://doi.org/10.1371/journal.pone.0061562

Boulesteix, A.-L., Stierle, V., & Hapfelmeier, A. (2015). Publication bias in methodological computational research. *Cancer Informatics*, *14*(S5), 11–19. https://doi.org/10.4137/CIN.S30747

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*(1), 138. https://doi.org/10.1186/s12874-017-0417-2

Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi Sepahvand, N., Raff, E., Madan, K., Voleti, V., Ebrahimi Kahou, S., Michalski, V., Arbel, T., Pal, C., Varoquaux, G., & Vincent, P. (2021). Accounting for variance in machine learning benchmarks [https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html]. In A. Smola, A. Dimakis, & I. Stoica (Eds.), *Proceedings of machine learning and systems* (pp. 747–769, Vol. 3).

Bouthillier, X., Laurent, C., & Vincent, P. (2019). Unreproducible research is reproducible [https://proceedings.mlr.press/v97/bouthillier19a.html]. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 725–734). PMLR.

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, *39*(3/4), 324. https://doi.org/10.2307/2334029

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth. https://doi.org/10.1201/9781315139470

Brombacher, E., Schilling, O., & Kreutz, C. (2025). Characterizing the omics landscape based on 10,000+ datasets. *Scientific Reports*, *15*(1), 3189. https://doi.org/10.1038/s41598-025-87256-5

Brooks, T. G., Lahens, N. F., Mrčela, A., & Grant, G. R. (2024). Challenges and best practices in omics benchmarking. *Nature Reviews Genetics*, *25*(5), 326–339. https://doi.org/10.1038/s41576-023-00679-6

Buchka, S., Hapfelmeier, A., Gardner, P. P., Wilson, R., & Boulesteix, A.-L. (2021). On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biology*, *22*(1), 152. https://doi.org/10.1186/s13059-021-02365-4

Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Nave, G., Nosek, B. A., Pfeiffer, T., Altmejd, A., Buttrick, N., Chan, T., Chen, Y., Forsell, E., Gampa, A., Heikensten, E., Hummer, L., Imai, T., ... Wu, H. (2018). Evaluating the replicability of social science experiments in Nature and Science between 2010 and 2015. *Nature Human Behaviour*, *2*(9), 637–644. https://doi.org/10.1038/s41562-018-0399-z

Chambers, C. D. (2013). Registered Reports: A new publishing initiative at Cortex. *Cortex*, *49*(3), 609–610. https://doi.org/10.1016/j.cortex.2012.12.016

Davey Smith, G. (2002). Data dredging, bias, or confounding. *BMJ*, *325*(7378), 1437–1438. https://doi.org/10.1136/bmj.325.7378.1437

de Leeuw, J., & Mair, P. (2009). Multidimensional Scaling Using Majorization: SMACOF in R. *Journal of Statistical Software*, *31*. https://doi.org/10.18637/jss.v031.i03

Dehghani, M., Tay, Y., Gritsenko, A. A., Zhao, Z., Houlsby, N., Diaz, F., Metzler, D., & Vinyals, O. (2021). The benchmark lottery. *arXiv:2107.07002 [cs.LG]*. https://doi.org/10.48550/arXiv.2107.07002

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets [https://jmlr.org/papers/v7/demsar06a.html]. *Journal of Machine Learning Research*, *7*, 1–30.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Duan, R., Gao, L., Gao, Y., Hu, Y., Xu, H., Huang, M., Song, K., Wang, H., Dong, Y., Jiang, C., Zhang, C., & Jia, S. (2021). Evaluation and comparison of multi-omics data integration methods for cancer subtyping. *PLOS Computational Biology*, *17*(8), e1009224. https://doi.org/10.1371/journal.pcbi.1009224

Duin, R. P. (1996). A note on comparing classifiers. *Pattern Recognition Letters*, *17*(5), 529–536. https://doi.org/10.1016/0167-8655(95)00113-1

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, *81*(394), 461–470. https://doi.org/10.1080/01621459.1986.10478291

Errington, T. M., Mathur, M., Soderberg, C. K., Denis, A., Perfito, N., Iorns, E., & Nosek, B. A. (2021). Investigating the replicability of preclinical cancer biology. *eLife*, *10*. https://doi.org/10.7554/elife.71601

Eugster, M. J. A., Hothorn, T., & Leisch, F. (2012). Domain-based benchmark experiments: Exploratory and inferential analysis. *Austrian Journal of Statistics*, *41*(1), 5–26. https://doi.org/10.17713/ajs.v41i1.185

Eugster, M. J., Leisch, F., & Strobl, C. (2014). (Psycho-)analysis of benchmark experiments: A formal framework for investigating the relationship between data sets and learning algorithms. *Computational Statistics and Data Analysis*, *71*, 986–1000. https://doi.org/10.1016/j.csda.2013.08.007

Fernández-Castilla, B., Jamshidi, L., Declercq, L., Beretvas, S. N., Onghena, P., & Van den Noortgate, W. (2020). The application of meta-analytic (multi-level) models with multiple random effects: A systematic review. *Behavior Research Methods*, *52*(5), 2031–2052. https://doi.org/10.3758/s13428-020-01373-9

Ferrari Dacrema, M., Boglio, S., Cremonesi, P., & Jannach, D. (2021). A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems*, *39*(2), 1–49. https://doi.org/10.1145/3434185

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_1

Franklin, J. M., Schneeweiss, S., Polinski, J. M., & Rassen, J. A. (2014). Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. *Computational Statistics & Data Analysis*, *72*, 219–226. https://doi.org/10.1016/j.csda.2013.10.018

Friedrich, S., & Friede, T. (2024). On the role of benchmarking data sets and simulations in method comparison studies. *Biometrical Journal*, *66*(1), 2200212. https://doi.org/10.1002/bimj.202200212

Guevara Morel, A. E., Varga, A. N., Heymans, M. W., Dongen, J. M., Schaik, D. J. F., Tulder, M. W., & Bosmans, J. E. (2022). Dealing with missing data in real-world data: A scoping review of simulation studies. *Preprint (version 1) available at Research Square*. https://doi.org/10.21203/rs.3.rs-1619388/v1

Gundersen, O. E., Coakley, K., Kirkpatrick, C., & Gil, Y. (2023). Sources of irreproducibility in machine learning: A review. *arXiv:2204.07610 [cs.LG]*. https://doi.org/10.48550/arXiv.2204.07610

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical Science*, *21*(1), 1–14. https://doi.org/10.1214/088342306000000060

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd). Springer. https://doi.org/10.1007/978-0-387-84858-7

Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., & Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLOS Biology*, *13*(3), e1002106. https://doi.org/10.1371/journal.pbio.1002106

Heinze, G., Boulesteix, A.-L., Kammer, M., Morris, T. P., White, I. R., & Simulation Panel of the STRATOS initiative. (2024). Phases of methodological research in biostatistics—building the evidence base for new methods. *Biometrical Journal*, *66*(1), 2200222. https://doi.org/10.1002/bimj.202200222

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 3207–3214. https://doi.org/10.1609/aaai.v32i1.11694

Henseler, J., Lee, N., Roemer, E., Kemény, I., Dirsehan, T., & Cadogan, J. W. (2024). Beware of the Woozle effect and belief perseverance in the PLS-SEM literature! *Electronic Commerce Research*, *24*(2), 715–744. https://doi.org/10.1007/s10660-024-09849-y

Herrmann, M., Lange, F. J. D., Eggensperger, K., Casalicchio, G., Wever, M., Feurer, M., Rügamer, D., Hüllermeier, E., Boulesteix, A.-L., & Bischl, B. (2024). Position: Why we must rethink empirical research in machine learning. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, & F. Berkenkamp (Eds.), *Proceedings of the 41st international conference on machine learning* (pp. 18228–18247, Vol. 235). PMLR. https://proceedings.mlr.press/v235/herrmann24b.html

Herrmann, M., Probst, P., Hornung, R., Jurinovic, V., & Boulesteix, A.-L. (2021). Large-scale benchmark study of survival prediction methods using multi-omics data. *Briefings in Bioinformatics*, *22*(3), bbaa167. https://doi.org/10.1093/bib/bbaa167

Hodges, C. B., Stone, B. M., Johnson, P. K., Carter, J. H., Sawyers, C. K., Roby, P. R., & Lindsey, H. M. (2022). Researcher degrees of freedom in statistical software contribute to unreliable results: A comparison of nonparametric analyses conducted in SPSS, SAS, Stata, and R. *Behavior Research Methods*, *55*(6), 2813–2837. https://doi.org/10.3758/s13428-022-01932-2

Hofner, B., Schmid, M., & Edler, L. (2015). Reproducible research in statistics: A review and guidelines for the Biometrical Journal. *Biometrical Journal*, *58*(2), 416–427. https://doi.org/10.1002/bimj.201500156

Hornik, K., & Meyer, D. (2007). Deriving consensus rankings from benchmarking experiments. In *Advances in data analysis* (pp. 163–170). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-70981-7_19

Hornung, R., & Wright, M. N. (2019). Block forests: Random forests for blocks of clinical and omics covariate data. *BMC Bioinformatics*, *20*(1), 358. https://doi.org/10.1186/s12859-019-2942-y

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Hothorn, T., Leisch, F., Zeileis, A., & Hornik, K. (2005). The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, *14*(3), 675–699. https://doi.org/10.1198/106186005X59630

Hullman, J., Kapoor, S., Nanayakkara, P., Gelman, A., & Narayanan, A. (2022). The worst of both worlds: A comparative analysis of errors in learning from data in psychology and machine learning. *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, 335–348. https://doi.org/10.1145/3514094.3534196

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLOS Medicine*, *2*(8), e124. https://doi.org/10.1371/journal.pmed.0020124

Ioannidis, J. P. A. (2008). Why most discovered true associations are inflated. *Epidemiology*, *19*(5), 640–648. https://doi.org/10.1097/ede.0b013e31818131e7

Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., & Boulesteix, A.-L. (2010). Over-optimism in bioinformatics: An illustration. *Bioinformatics*, *26*(16), 1990–1998. https://doi.org/10.1093/bioinformatics/btq323

John, L. K., Loewenstein, G., & Prelec, D. (2012). Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, *23*(5), 524–532. https://doi.org/10.1177/0956797611430953

Kandanaarachchi, S., & Smith-Miles, K. (2023). Comprehensive algorithm portfolio evaluation using item response theory. *Journal of Machine Learning Research*, *24*(177), 1–52. http://jmlr.org/papers/v24/20-1318.html

Kapoor, S., Cantrell, E. M., Peng, K., Pham, T. H., Bail, C. A., Gundersen, O. E., Hofman, J. M., Hullman, J., Lones, M. A., Malik, M. M., Nanayakkara, P., Poldrack, R. A., Raji, I. D., Roberts, M., Salganik, M. J., Serra-Garcia, M., Stewart, B. M., Vandewiele, G., & Narayanan, A. (2024). REFORMS: Consensus-based recommendations for machine-learning-based science. *Science Advances*, *10*(18), eadk3452. https://doi.org/10.1126/sciadv.adk3452

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Keogh, E., & Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, *7*(4), 349–371. https://doi.org/10.1023/a:1024988512476

Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and Social Psychology Review*, *2*(3), 196–217. https://doi.org/10.1207/s15327957pspr0203_4

Klau, S., Jurinovic, V., Hornung, R., Herold, T., & Boulesteix, A.-L. (2018). Priority-Lasso: A simple hierarchical approach to the prediction of clinical outcome using multi-omics data. *BMC Bioinformatics*, *19*(1), 322. https://doi.org/10.1186/s12859-018-2344-6

Kodalci, L., & Thas, O. (2024). Neutralise: An open science initiative for neutral comparison of two-sample tests. *Biometrical Journal*, *66*(1), 2200237. https://doi.org/10.1002/bimj.202200237

Kreutz, C. (2016). New concepts for evaluating the performance of computational methods. *IFAC-PapersOnLine*, *49*(26), 63–70. https://doi.org/10.1016/j.ifacol.2016.12.104

Kreutz, C. (2019). Guidelines for benchmarking of optimization-based approaches for fitting mathematical models. *Genome Biology*, *20*(1). https://doi.org/10.1186/s13059-019-1887-9

Kreutz, C., Can, N. S., Bruening, R. S., Meyberg, R., Mérai, Z., Fernandez-Pozo, N., & Rensing, S. A. (2020). A blind and independent benchmark study for detecting differentially methylated regions in plants. *Bioinformatics*, *36*(11), 3314–3321. https://doi.org/10.1093/bioinformatics/btaa191

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer New York. https://doi.org/10.1007/978-1-4614-6849-3

Langan, D., Higgins, J. P. T., & Simmonds, M. (2017). Comparative performance of heterogeneity variance estimators in meta-analysis: A review of simulation studies. *Research Synthesis Methods*, *8*(2), 181–198. https://doi.org/10.1002/jrsm.1198

Lange, F. J. D., Wilcke, J. C., Hoffmann, S., Herrmann, M., & Boulesteix, A.-L. (2025). On "confirmatory" methodological research in statistics and related fields. *Statistics in Medicine*, *44*(25–27), e70303. https://doi.org/https://doi.org/10.1002/sim.70303

Li, C., Dakkak, A., Xiong, J., & Hwu, W.-m. (2019). Challenges and pitfalls of machine learning evaluation and benchmarking. *arXiv:1904.12437 [cs.LG]*. https://doi.org/10.48550/arxiv.1904.12437

Lohmann, A., Astivia, O. L. O., Morris, T. P., & Groenwold, R. H. H. (2022). It's time! Ten reasons to start replicating simulation studies. *Frontiers in Epidemiology*, *2*, 973470. https://doi.org/10.3389/fepid.2022.973470

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. https://doi.org/10.1186/s13059-014-0550-8

Lucic, M., Kurach, K., Michalski, M., Gelly, S., & Bousquet, O. (2018). Are GANs created equal? A large-scale study [https://papers.neurips.cc/paper_files/paper/2018/hash/e46de7e1bcaaced9a54f1e9d0d2f800d-Abstract.html]. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 31). Curran Associates Inc.

Luijken, K., Lohmann, A., Alter, U., Claramunt Gonzalez, J., Clouth, F. J., Fossum, J. L., Hesen, L., Huizing, A. H. J., Ketelaar, J., Montoya, A. K., Nab, L., Nijman, R. C. C., Penning de Vries, B. B. L., Tibbe, T. D., Wang, Y. A., & Groenwold, R. H. H. (2024). Replicability of simulation studies for the investigation of statis-

tical methods: The replisims project. *Royal Society Open Science*, *11*(1). https://doi.org/10.1098/rsos.231003

Macià, N., Bernadó-Mansilla, E., Orriols-Puig, A., & Kam Ho, T. (2013). Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition*, *46*(3), 1054–1066. https://doi.org/10.1016/j.patcog.2012.09.022

Makino, M., Shimizu, K., & Kadota, K. (2023). Evaluation of clustering-based differential expression analysis methods for RNA-seq data. https://doi.org/10.21203/rs.3.rs-3374125/v1

Makino, M., Shimizu, K., & Kadota, K. (2024). Enhanced clustering-based differential expression analysis method for RNA-seq data. *MethodsX*, *12*, 102518. https://doi.org/10.1016/j.mex.2023.102518

Mandl, M. M., Weber, F., Wöhrle, T., & Boulesteix, A.-L. (2025). The impact of the storytelling fallacy on real data examples in methodological research. *arXiv:2503.03484 [stat.OT]*. https://doi.org/10.48550/arXiv.2503.03484

Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., & Weihs, C. (2015). Analyzing the BBOB results by means of benchmarking concepts. *Evolutionary Computation*, *23*(1), 161–185. https://doi.org/10.1162/evco_a_00134

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086

Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., du Sert, N. P., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., & Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, *1*(1), 1–9. https://doi.org/10.1038/s41562-016-0021

Nguyen, H., Shrestha, S., Draghici, S., & Nguyen, T. (2019). PINSPlus: a tool for tumor subtype discovery in integrated genomic data. *Bioinformatics*, *35*(16), 2843–2846. https://doi.org/10.1093/bioinformatics/bty1049

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(2), e1441. https://doi.org/10.1002/widm.1441

Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2024). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *Biometrical Journal*, *66*(1), 2200238. https://doi.org/10.1002/bimj.202200238

Norel, R., Rice, J. J., & Stolovitzky, G. (2011). The self-assessment trap: Can we all be better than average? *Molecular Systems Biology*, *7*(1), 537. https://doi.org/10.1038/msb.2011.70

Nuzzo, R. (2015). How scientists fool themselves — and how they can stop. *Nature*, *526*(7572), 182–185. https://doi.org/10.1038/526182a

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251). https://doi.org/10.1126/science.aac4716

Oreski, D., Oreski, S., & Klicek, B. (2017). Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, *52*, 109–119. https://doi.org/10.1016/j.asoc.2016.12.023

Osabe, T., Shimizu, K., & Kadota, K. (2021). Differential expression analysis using a model-based gene clustering algorithm for RNA-seq data. *BMC Bioinformatics*, *22*(1), 511. https://doi.org/10.1186/s12859-021-04438-4

Pawel, S., Bartoš, F., Siepe, B. S., & Lohmann, A. (2025). Handling missingness, failures, and non-convergence in simulation studies: A review of current practices and recommendations [Advance online publication]. *The American Statistician*. https://doi.org/10.1080/00031305.2025.2540002

Pawel, S., Kook, L., & Reeve, K. (2024). Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method. *Biometrical Journal*, *66*(1), 2200091. https://doi.org/10.1002/bimj.202200091

Pénichoux, J., Moreau, T., & Latouche, A. (2015). Simulating recurrent events that mimic actual data: A review of the literature with emphasis on event-dependence. *arXiv:1503.05798 [stat.AP]*. https://doi.org/10.48550/arXiv.1503.05798

Picard, D. (2023). Torch.manual_seed(3407) is all you need: On the influence of random seeds in deep learning architectures for computer vision. *arXiv:2109.08203 [cs.CV]*. https://doi.org/10.48550/arXiv.2109.08203

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F., Fox, E., & Larochelle, H. (2021). Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program) [https://jmlr.org/papers/v22/20-303.html]. *Journal of Machine Learning Research*, *22*, 1–20.

Plesser, H. E. (2018). Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, *11*. https://doi.org/10.3389/fninf.2017.00076

Probst, P., Boulesteix, A.-L., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, *20*(53), 1–32.

Rappoport, N., & Shamir, R. (2019). NEMO: cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, *35*(18), 3348–3356. https://doi.org/10.1093/bioinformatics/btz058

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2009). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Rofin, M., Mikhailov, V., Florinsky, M., Kravchenko, A., Shavrina, T., Tutubalina, E., Karabekyan, D., & Artemova, E. (2023). Vote'n'rank: Revision of benchmarking with social choice theory. *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2023.eacl-main.48

Sauer, C., Boulesteix, A.-L., Hanßum, L., Hodiamont, F., Bausewein, C., & Ullmann, T. (2024). Beyond algorithm hyperparameters: On preprocessing hyperparameters and associated pitfalls in machine learning applications. *arXiv:2412.03491 [stat.ML]*. https://doi.org/10.48550/arXiv.2412.03491

Sauer, C., Lange, F. J. D., Thurow, M., Dormuth, I., & Boulesteix, A.-L. (2025). Statistical parametric simulation studies based on real data. *arXiv:2504.04864 [stat.ME]*. https://doi.org/10.48550/arXiv.2504.04864

Schreck, N., Slynko, A., Saadati, M., & Benner, A. (2024). Statistical plasmode simulations — Potentials, challenges and recommendations. *Statistics in Medicine*, *43*(9), 1804–1825. https://doi.org/https://doi.org/10.1002/sim.10012

Schulz-Kümpel, H., Fischer, S. F., Hornung, R., Boulesteix, A.-L., Nagler, T., & Bischl, B. (2025). Constructing confidence intervals for "the" generalization error – a comprehensive benchmark study. *Journal of Data-centric Machine Learning Research*. https://openreview.net/forum?id=x7kCj9OU2c

Siepe, B. S., Bartoš, F., Morris, T. P., Boulesteix, A.-L., Heck, D. W., & Pawel, S. (2024). Simulation studies for methodological research in psychology: A standardized template for planning, preregistration, and reporting. *Psychological Methods*. https://doi.org/10.1037/met0000695

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366. https://doi.org/10.1177/0956797611417632

Smith, H., Sweeting, M., Morris, T., & Crowther, M. J. (2022). A scoping methodological review of simulation studies comparing statistical and machine learning approaches to risk prediction for time-to-event data. *Diagnostic and Prognostic Research*, *6*(1). https://doi.org/10.1186/s41512-022-00124-y

Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., & Bommert, A. (2024). Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression. *PLOS ONE*, *19*(5), e0299989. https://doi.org/10.1371/journal.pone.0299989

Strobl, C., & Leisch, F. (2024). Against the "one method fits all data sets" philosophy for comparison studies in methodological research. *Biometrical Journal*, *66*(1), 2200104. https://doi.org/10.1002/bimj.202200104

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*(4), 323–348. https://doi.org/10.1037/a0016973

Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics data integration, interpretation, and its application. *Bioinformatics and Biology Insights*, *14*, 117793221989905. https://doi.org/10.1177/1177932219899051

The Turing Way Community. (2025). *The Turing Way: A handbook for reproducible, ethical and collaborative research* (Version 1.2.3). Zenodo. https://doi.org/10.5281/zenodo.15213042

Ullmann, T., Beer, A., Hünemörder, M., Seidl, T., & Boulesteix, A.-L. (2023). Over-optimistic evaluation and reporting of novel cluster algorithms: An illustrative study. *Advances in Data Analysis and Classification*, *17*(1), 211–238. https://doi.org/10.1007/s11634-022-00496-5

Van Mechelen, I., Boulesteix, A.-L., Dangl, R., Dean, N., Hennig, C., Leisch, F., Steinley, D., & Warrens, M. J. (2023). A white paper on good research practices in benchmarking: The case of cluster analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *13*(6), e1511. https://doi.org/10.1002/widm.1511

Wagenmakers, E.-J., Wetzels, R., Borsboom, D., van der Maas, H. L. J., & Kievit, R. A. (2012). An agenda for purely confirmatory research. *Perspectives on Psychological Science*, *7*(6), 632–638. https://doi.org/10.1177/1745691612463078

Wagstaff, K. L. (2012). Machine learning that matters. *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1851–1856.

Wainer, J. (2023). A Bayesian Bradley-Terry model to compare multiple ML algorithms on multiple data sets. *Journal of Machine Learning Research*, *24*(341), 1–34. http://jmlr.org/papers/v24/22-0907.html

Weber, L. M., Saelens, W., Cannoodt, R., Soneson, C., Hapfelmeier, A., Gardner, P. P., Boulesteix, A.-L., Saeys, Y., & Robinson, M. D. (2019). Essential guidelines for computational method benchmarking. *Genome Biology*, *20*(1), 125. https://doi.org/10.1186/s13059-019-1738-8

Welvaert, M., & Rosseel, Y. (2014). A review of fMRI simulation studies. *PLoS ONE*, *9*(7), e101953. https://doi.org/10.1371/journal.pone.0101953

Wiggins, B. J., & Christopherson, C. D. (2019). The replication crisis in psychology: An overview for theoretical and philosophical psychology. *Journal of Theoretical and Philosophical Psychology*, *39*(4), 202–217. https://doi.org/10.1037/teo0000137

Wünsch, M., Herrmann, M., Noltenius, E., Mohr, M., Morris, T. P., & Boulesteix, A.-L. (2025). Rethinking the handling of method failure in comparison studies. *Statistics in Medicine*, *44*(23–24), e70257. https://doi.org/https://doi.org/10.1002/sim.70257

Xie, C., Jauhari, S., & Mora, A. (2021). Popularity and performance of bioinformatics software: The case of gene set analysis. *BMC Bioinformatics*, *22*(1), 191. https://doi.org/10.1186/s12859-021-04124-5

Yousefi, M. R., Hua, J., Sima, C., & Dougherty, E. R. (2010). Reporting bias when using real data sets to analyze classification performance. *Bioinformatics*, *26*(1), 68–76. https://doi.org/10.1093/bioinformatics/btp605

A Contribution 1: "Beyond algorithm hyperparameters: on preprocessing hyperparameters and associated pitfalls in machine learning applications"

This section is a reprint of:

Sauer, C., Boulesteix, A.-L., Hanßum, L., Hodiamont, F., Bausewein, C., & Ullmann, T. (2024). Beyond algorithm hyperparameters: On preprocessing hyperparameters and associated pitfalls in machine learning applications. *arXiv:2412.03491 [stat.ML]*. https://doi.org/10.48550/arXiv.2412.03491

Copyright:

Author contributions:

C. Sauer, A.-L. Boulesteix, and T. Ullmann conceptualized the paper. C. Sauer and T. Ullmann designed the methodology, incorporating comments from A.-L. Boulesteix and building on a pilot study conducted by L. Hanßum. C. Sauer wrote the R code for the empirical illustration, reusing selected elements from code originally developed by C. Sauer and T. Ullmann for a real-world prediction study. The original draft of the manuscript was written by C. Sauer. All authors contributed to the review and editing of the manuscript.

Beyond algorithm hyperparameters: on preprocessing hyperparameters and associated pitfalls in machine learning applications

Christina Sauer[1,2], Anne-Laure Boulesteix[1,2], Luzia Hanßum[1], Farina Hodiamont[3], Claudia Bausewein[3], and Theresa Ullmann*[4]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, Faculty of Medicine, LMU Munich, Munich, Germany

[2]Munich Center for Machine Learning (MCML), Munich, Germany

[3]Department of Palliative Medicine, University Hospital, LMU Munich, Munich, Germany

[4]Institute of Clinical Biometrics, Center for Medical Data Science, Medical University of Vienna, Vienna, Austria

August 15, 2025

Abstract

Adequately generating and evaluating prediction models based on supervised machine learning (ML) is often challenging, especially for less experienced users in applied research areas. Special attention is required in settings where the model generation process involves hyperparameter tuning, i.e. data-driven optimization of different types of hyperparameters to improve the predictive performance of the resulting model. Discussions about tuning typically focus on the hyperparameters of the ML algorithm (e.g., the minimum number of observations in each terminal node for a tree-based algorithm). In this context, it is often neglected that hyperparameters also exist for the preprocessing steps that are applied to the data before it is provided to the algorithm (e.g., how to handle missing feature values in the data). As a consequence, users experimenting with different preprocessing options to improve model performance may be unaware that this constitutes a form of hyperparameter tuning, albeit informal and unsystematic, and thus may fail to report or account for this optimization. To illuminate this issue, this paper reviews and empirically illustrates different procedures for generating and evaluating prediction models, explicitly addressing the different ways algorithm and preprocessing hyperparameters are typically handled by applied ML users. By highlighting potential pitfalls, especially those that may lead to exaggerated performance claims, this review aims to further improve the quality of predictive modeling in ML applications.

Keywords: predictive modeling, machine learning, preprocessing, hyperparameter optimization, tuning

*Corresponding author, e-mail: theresa.ullmann@meduniwien.ac.at

1 Introduction

Many applied research areas have recently seen an increase in the development of prediction models based on supervised machine learning (ML) algorithms. However, after initially generating widespread enthusiasm—partly due to the availability of user-friendly software that enables model development without requiring extensive expertise—ML-based prediction models are now undergoing critical reexamination (Ball, 2023; Kapoor & Narayanan, 2023; Pfob et al., 2022). Among other concerns, such as insufficient reporting of relevant aspects of the model development process, it has been found that the claimed predictive performance of many models is considerably exaggerated (Andaur Navarro et al., 2021; Dhiman et al., 2022a, 2022b; Kapoor & Narayanan, 2023). While some of the pitfalls leading to such optimistically biased performance claims (e.g., using the exact same observations for model generation and evaluation) typically occur only among very inexperienced applied ML users and are well known within the ML research community, others arise more subtly (Domingos, 2012; Hofman et al., 2023; Kapoor & Narayanan, 2023; Poldrack et al., 2020).

This is particularly true when the model generation process involves data-driven hyperparameter optimization, which is also referred to as hyperparameter tuning and is commonly employed in ML applications. The most prominent type of hyperparameters (HPs) are those associated with the learning algorithm, which specify its configuration (e.g., the minimum number of observations in each terminal node for tree-based algorithms). If selected by an adequate (and ideally automated) tuning procedure, HPs can substantially enhance the performance of the resulting prediction model. However, HP tuning also complicates model evaluation, as common procedures such as simple k-fold cross-validation no longer guarantee an unbiased assessment (Bischl et al., 2023; Hosseini et al., 2020).

An additional challenge comes from the fact that, beyond algorithm HPs, there are also preprocessing HPs, which specify the steps applied to the data before it is fed into the learning algorithm (e.g., selecting the set of features for prediction or determining how missing feature values are handled; Binder and Pfisterer, 2024; Bischl et al., 2023). While the tuning of algorithm HPs is rightfully considered important for model performance, the relevance of tuning preprocessing HPs should not be overlooked. Preprocessing steps can make or break a model's predictive performance, and solely relying on user expertise to specify these steps (which is the alternative to tuning) is often impractical and may result in arbitrary decisions (Kuhn & Johnson, 2013). Despite this, reports of tuning preprocessing HPs aside from feature selection are relatively rare. This could be because integrating preprocessing HPs into automated tuning workflows typically requires advanced programming expertise, which not all applied ML users have, or because this possibility is not widely recognized. Importantly, the limited use of automated tuning procedures for preprocessing HPs does not mean that these HPs are not being tuned at all. In fact, it appears fairly common for applied ML users to experiment informally with different preprocessing options (Hofman et al., 2023; Hosseini et al., 2020; Lones, 2024), often without realizing that this constitutes a form of (manual) HP tuning. If this type of tuning

is indeed conducted subconsciously, it will also remain unaccounted for during model evaluation, thereby increasing the risk of drawing overly optimistic conclusions about the model's performance.

To avoid such issues, it is essential to educate users in applied settings about the different types of HPs, the different forms of HP tuning, and how tuning can impact both the true and estimated performance of prediction models. Although valuable literature already exists describing the concept of HP tuning and various automated procedures (e.g., Bartz et al., 2023; Bischl et al., 2023; Feurer & Hutter, 2019), this research primarily adopts the perspective of ML methods researchers who are concerned with evaluating the overall performance of ML algorithms used to generate prediction models. This focus does not align with the perspective of applied ML users, who are more interested in the performance of a specific prediction model. Although this literature is still useful for them—since the general principles described there essentially hold for all types of audiences—applied ML users additionally need specific guidance for developing their "final model" (a notion that does not exist in the methodological context). Moreover, they may find it challenging to extract the relevant insights from literature aimed at a different audience with partly different needs. In contrast, literature explicitly directed toward applied ML users tends to either focus on general guidelines for ML-based predictive modeling, lacking detailed coverage of HP tuning (e.g., Collins, Dhiman, et al., 2024; Kapoor et al., 2024; Kuhn & Johnson, 2013; Lones, 2024; Pfob et al., 2022; Poldrack et al., 2020; van Royen et al., 2023), or addresses HP tuning only within specific research areas (e.g., Dunias et al., 2024; Hosseini et al., 2020). Additionally, much of the existing HP tuning literature does not consider preprocessing HPs. Exceptions include the review by Bischl et al., 2023, which, however, touches on this topic only briefly. This lack of detail is reasonable, given that preprocessing HPs can, in principle, be tuned using the same automated procedures as algorithm HPs. However, this perspective overlooks that preprocessing HPs are often tuned manually in applied settings, which carries implications different from those associated with automated tuning.

This paper aims to complement the existing literature by reviewing the implications and pitfalls of HP tuning in the generation and evaluation of prediction models from the perspective of applied ML users with varying levels of expertise. It explicitly distinguishes between preprocessing and algorithm HPs, as well as the different procedures commonly used to tune them in practice. A particular focus is placed on the potential for optimistically biased performance estimation, which is also illustrated using a real-world prediction problem from palliative care medicine.

The paper is structured as follows. Section 2 introduces the key concepts related to predictive modeling using ML, including the two types of HPs. In the next two sections, the challenges and pitfalls that arise in the generation and evaluation of prediction models are described, differentiating between the setting where all HPs are pre-specified (Section 3) and the setting where one or more HPs are selected through tuning (Section 4). Section 5 empirically illustrates the impact of different tuning and evaluation procedures on the estimated model performance.

Section 6 summarizes the key insights, discusses the limitations of the empirical study, and outlines future research directions.

2 General concepts of predictive modeling using supervised ML

2.1 Terminology and notation

The following terminology and notation is adapted from Bischl et al. (2023). Let $\mathcal{D}_{\text{train}}$ be a labeled data set with n_{train} observations. Accordingly, each observation i ($i = 1, \ldots, n_{\text{train}}$) consists of an outcome $y^{(i)}$ (i.e. the variable to be predicted, also referred to as label or target) and a p-dimensional feature vector $\boldsymbol{x}^{(i)}$ (i.e. the p variables used to predict $y^{(i)}$, also referred to as predictors), where $y^{(i)}$ and $\boldsymbol{x}^{(i)}$ can take any value from the outcome space $\mathcal{Y}$ and feature space $\mathcal{X}$, respectively. Two common types of prediction problems are regression, for which $y^{(i)}$ can be any real number (i.e. $\mathcal{Y} = \mathbb{R}$), and classification, for which $y^{(i)}$ can be one of g classes (i.e. $\mathcal{Y}$ is finite and categorical with $|\mathcal{Y}| = g$). We assume that the observations in $\mathcal{D}_{\text{train}}$ are independent and have been sampled from the same (unknown) probability distribution $\mathbb{P}_{xy}$.

The general aim of supervised ML is to "learn" a model from the data set $\mathcal{D}_{\text{train}}$ that is able to predict the outcome values of new observations. Essentially, a prediction model is a function $\hat{f} : \mathcal{X} \to \mathbb{R}^g$ that maps any observed feature vector $\boldsymbol{x}$ to a prediction vector $\hat{f}(\boldsymbol{x})$ in $\mathbb{R}^g$. The prediction vector $\hat{f}(\boldsymbol{x})$ either directly corresponds to the predicted outcome value (e.g., for regression, where $g = 1$) or can be transformed accordingly (e.g., for classification, where $\hat{f}(\boldsymbol{x})$ corresponds to predicted probabilities for each class and the predicted class could be the class with the highest probability). The prediction model results from a learning pipeline $\mathcal{I}$, which uses the data set $\mathcal{D}_{\text{train}}$ to find the function $\hat{f}$ that yields the best predictions for the true outcome values in $\mathcal{D}_{\text{train}}$. To stress that a prediction model $\hat{f}$ is based on learning pipeline $\mathcal{I}$ and data set $\mathcal{D}_{\text{train}}$, we write $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$. The prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ can usually be parameterized, meaning that it is defined by a set of parameters $\hat{\boldsymbol{\theta}}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ (simply denoted as $\hat{\boldsymbol{\theta}}$ when data set and learning pipeline are clear from context and $\boldsymbol{\theta}$ when referring to the parameters prior to estimation).

There are two key processes associated with $\mathcal{I}$ and $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$, which we will explore in more detail throughout the paper: (i) the training process, in which the learning pipeline $\mathcal{I}$ is applied to $\mathcal{D}_{\text{train}}$ and estimates the parameters $\hat{\boldsymbol{\theta}}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ and thus the prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$, and (ii) the prediction process, in which $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ is used to make predictions for an observation (whether from $\mathcal{D}_{\text{train}}$ or from a new data set) with feature vector $\boldsymbol{x}$, resulting in $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}(\boldsymbol{x})$. Note that to make predictions on a new data set, the outcome does not need to be observed (it would only be necessary for evaluating those predictions). The training and prediction processes serve as the foundation for more complex processes related to the development of prediction models, which we will address in Section 2.4.

2.2 Learning pipeline

Each learning pipeline $\mathcal{I}$ contains a learning algorithm as a central component but can also include several preprocessing steps that are performed before the algorithm is applied to the data. Since preprocessing steps are a particular focus of this paper, we use the term "learning pipeline" instead of the more common term "learner" to emphasize that $\mathcal{I}$ can consist of several components. Note that for now, we consider all components of $\mathcal{I}$ as fixed, but we will discuss the case in which they can be modified in Section 2.3.

2.2.1 Learning algorithm

The choice of learning algorithm usually depends on the specific prediction problem. For example, if the desired prediction model is a decision tree (which is the case for the real-world prediction problem considered in Section 5), a possible algorithm choice is the well-known Classification and Regression Tree algorithm (CART), which partitions the feature space $\mathcal{X}$ by a sequence of binary splits into terminal nodes and assigns a prediction value to each terminal node (Breiman et al., 1984). In this case, the parameters of the learning algorithm contained in $\hat{\boldsymbol{\theta}}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ are the splitting rules that generate the tree structure (i.e. which features are used with which threshold value) and the prediction values at each terminal node. The learning algorithm can also consist of multiple individual algorithms that are combined into one overall algorithm (e.g., random forests). These types of algorithms are referred to as ensemble methods, but will not be discussed further in this paper. In general, the choice of algorithm has a large impact on the hypothesis space of the learning pipeline, i.e. the set of prediction models the learning pipeline can generate. For example, selecting a standard linear regression as algorithm (with $\hat{\boldsymbol{\theta}}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ containing the regression coefficients) would imply that the corresponding learning pipeline would not be able to learn prediction models that do not correspond to linear combinations of the features (e.g., polynomials).

2.2.2 Preprocessing

While a data set can, in theory, be fed directly into the algorithm (i.e. the algorithm is the only component of the learning pipeline), it typically undergoes some modification first. This process can be referred to as data preprocessing and encompasses all the steps taken to transform the data set from its rawest available form into the final form provided as input to the learning algorithm (Kapoor et al., 2024). Data preprocessing steps are usually performed to improve the performance of the resulting prediction model, to enable the data to be (better) handled by the learning algorithm (Thomas, 2024), or to improve the interpretability of the resulting prediction model. To better illustrate the different characteristics of preprocessing steps and their implications on the training and prediction process, we consider a simple learning pipeline as an example, which is also depicted in Figure 1 (middle panel). It consists of two preprocessing steps, which are followed by the CART algorithm. The first preprocessing step is the replacement of missing feature values using mean imputation, and the second preprocessing step is the log-transformation of features.

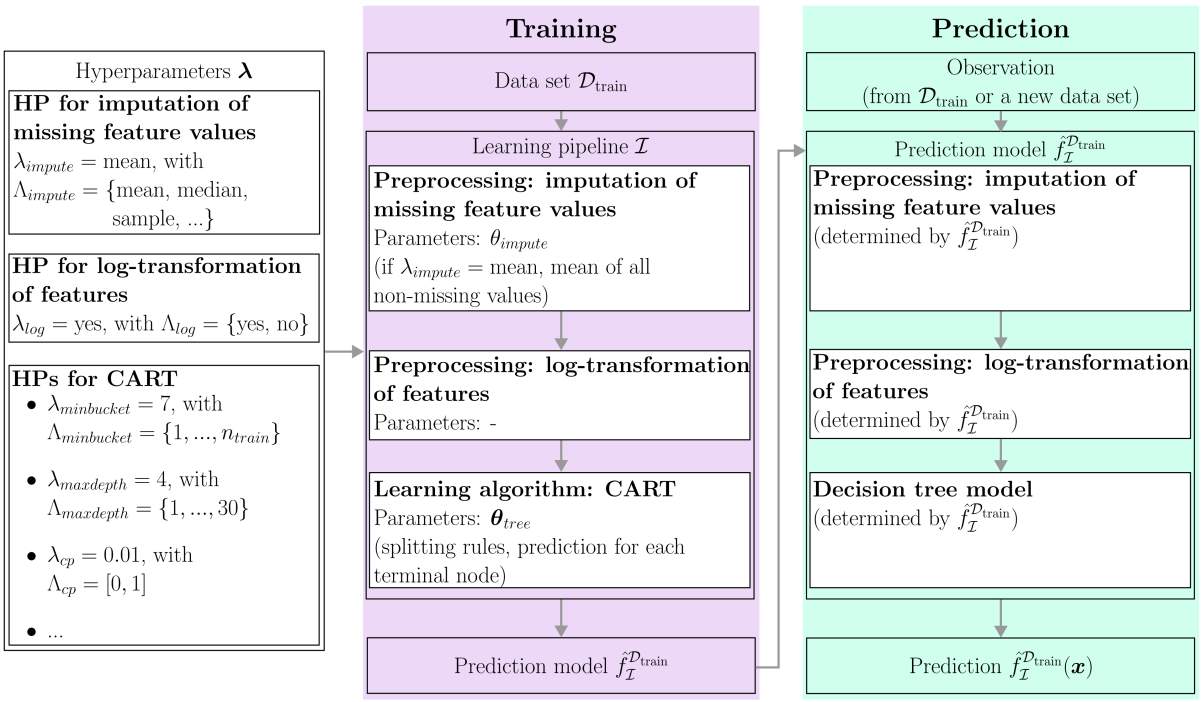

Figure 1: Example of a learning pipeline $\mathcal{I}$ consisting of two preprocessing steps and one learning algorithm. Left panel: HPs of the learning pipeline, with each HP set to an example value. Middle panel: Training process, where the learning pipeline is applied to the data set $\mathcal{D}_{\text{train}}$ to generate the prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$. Right panel: Prediction process, where a prediction for an observation with feature vector $\boldsymbol{x}$ is obtained by reapplying all preprocessing steps, followed by the prediction model resulting from the learning algorithm (here: a decision tree).

Parameterized vs. parameterless steps Based on this example learning pipeline, we can make a first distinction between preprocessing steps. This distinction concerns whether the steps have parameters estimated from $\mathcal{D}_{\text{train}}$ (with these parameters included in $\boldsymbol{\theta}$) or whether they are parameterless and are carried out independently for each observation (Binder & Pfisterer, 2024; Kapoor et al., 2024). In the example, the replacement of missing feature values is a parameterized preprocessing step, as it involves the parameter θ_{impute}, representing the mean of all non-missing values estimated from $\mathcal{D}_{\text{train}}$. In contrast, the log-transformation of features does not involve any parameters. Other examples of preprocessing steps with parameters include centering or scaling of features, where parameters such as the mean or standard deviation are estimated from $\mathcal{D}_{\text{train}}$. On the other hand, creating a new feature by summing multiple features serves as another example of a parameterless preprocessing step.

Application during prediction vs. training only The second key distinction in preprocessing steps concerns whether they are applied only during the training process as part of the learning pipeline or also during the prediction process. This distinction is closely related to whether a preprocessing step modifies only the feature distribution or also affects the outcome distribution. More formally, let $\boldsymbol{y}$ denote the outcome vector in $\mathcal{D}_{\text{train}}$. If, after applying all

preprocessing steps in the learning pipeline during training, $\boldsymbol{y}$ remains unchanged, we classify the step as affecting only the feature distribution. Otherwise, the step affects the outcome distribution, for example, by removing or adding observations or transforming outcome values. We first consider preprocessing steps that affect only the feature distribution. These comprise all preprocessing steps mentioned above, including those in the example learning pipeline. Additional examples are dimensionality reduction techniques (e.g., principal component analysis), feature selection, or data cleaning steps that do not alter the outcome distribution (e.g., correction of errors in features) (Kuhn & Johnson, 2013; Thomas, 2024). Preprocessing steps of this type must be applied not only during training but also during prediction, in the same sequence as in the learning pipeline. This ensures that the model produced by the learning algorithm receives the data in the same format during prediction as it did during training, preserving the validity of the model (Binder & Pfisterer, 2024). This requirement implies that these steps are not only components of the learning pipeline $\mathcal{I}$ but also part of the resulting prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$. Consequently, if a learning pipeline $\mathcal{I}$ includes h preprocessing steps that only affect the feature distribution, the prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ is not a single function but a function composition of $h+1$ functions (omitting $\mathcal{D}_{\text{train}}$ and $\mathcal{I}$ for simplicity of notation):

$$\hat{f}_{h+1}(\hat{f}_h(\ldots(\hat{f}_1(\boldsymbol{x})))), \tag{1}$$

where $\hat{f}_{h+1}$ corresponds to the model resulting from the learning algorithm, and $\hat{f}_h, \ldots, \hat{f}_1$ reflect the h preprocessing steps. Accordingly, a more accurate name for a prediction model would be prediction model *pipeline*, but for brevity, we will continue to use the former. Returning to the example learning pipeline, the resulting prediction model is a composition of three functions, $\hat{f}_3(\hat{f}_2(\hat{f}_1(\boldsymbol{x})))$, where $\hat{f}_1$, $\hat{f}_2$, and $\hat{f}_3$ correspond to the imputation step, the log-transformation step, and the decision tree model, respectively. When making a prediction for one or more observations, all three functions must be applied (see Figure 1, right panel). Importantly, if any functions constituting the prediction model are omitted during the prediction process, or if any preprocessing or algorithm parameters are re-estimated on a new data set for which predictions are to be made, the validity of the prediction model may be compromised. However, in practice, this pitfall is often unavoidable for users who wish to apply a model but were not involved in its development, as studies introducing new prediction models frequently fail to report the preprocessing steps performed prior to applying the learning algorithm (Kapoor et al., 2024).
In contrast to preprocessing steps that only affect the feature distribution, preprocessing steps that modify the outcome distribution are not necessarily applied during prediction. Here, we must distinguish between steps aimed at improving compatibility with the learning algorithm and those intended to alter the scope or interpretation of the prediction model. An example of the first type is (invertible) transformations applied to the outcome during training, such as a log-transformation to reduce skewness. To ensure predictions are returned on the correct scale, these transformations must be reversed during prediction (Thomas, 2024). For instance, if the outcome was log-transformed during training, the model will output $\log(\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}(\boldsymbol{x}))$, which

must then be exponentiated to restore the prediction to its original scale. Note that some other compatibility-focused steps are not applied at all during prediction. In the context of classification problems, this includes class-balancing steps such as oversampling, where observations from the least prevalent class are randomly resampled to overcome class imbalance effects during the training process (see, e.g., Kuhn and Johnson, 2013, for more details). In the notation of the prediction model as a function composition introduced above, preprocessing steps that are applied only in their inverted form or not at all during prediction are represented as inversion function or identity function, respectively.

In contrast, preprocessing steps that modify the outcome to alter the scope or interpretation of the prediction model should be consistently applied during prediction. For example, if a continuous outcome is discretized to convert a regression problem into a classification problem (Hofman et al., 2023), this (irreversible) transformation must also be applied to the true outcome during prediction in order to enable a meaningful comparison between the predictions and the actual outcome values. Such transformations of the outcome are not part of the prediction model itself (which maps $\boldsymbol{x}$ to predictions, not y), but must be performed alongside the prediction process. Moreover, since the outcome values are generally unknown when making predictions for observations from a new data set that does not correspond to $\mathcal{D}_{\text{train}}$, these transformations are typically not actual steps executed when making predictions but instead determine how the predictions are interpreted.

2.3 Hyperparameters

Until now, we have assumed that the learning pipeline $\mathcal{I}$ is fixed. However, individual components of $\mathcal{I}$ usually have several hyperparameters (HPs), which determine their specific configuration and thus substantially influence the resulting prediction model. This also applies to the learning pipeline example considered in the previous section, for which possible HPs are shown in the left panel of Figure 1 (see below for further explanation). In contrast to the parameters $\boldsymbol{\theta}$, which are estimated as outputs of the learning pipeline, the HPs serve as inputs. This means that they must be specified before the learning pipeline is applied to the data set (Bischl et al., 2023).

2.3.1 Additional notation for HPs

The following notation is based on Feurer and Hutter (2019). We denote the jth HP of a learning pipeline as λ_j, which is selected from its domain Λ_j (i.e. $\lambda_j \in \Lambda_j$). The domain of λ_j can generally be real-valued, integer-valued, binary, or categorical, as we will see in the examples given below. All J HPs of a learning pipeline can be summarized as a vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_J)$ and their overall configuration space as $\boldsymbol{\Lambda} = \Lambda_1 \times \Lambda_2 \cdots \times \Lambda_J$ (with $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$). Note that $\boldsymbol{\Lambda}$ may contain conditionality, meaning that some HPs might only be relevant when one or more other HPs are set to a certain value (see below for examples).

As described in Section 2.2, the learning pipeline consists of several preprocessing steps and

a learning algorithm. We can consequently differentiate between preprocessing and algorithm HPs, which we denote as $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_A$ (i.e. $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_P, \boldsymbol{\lambda}_A)$).

2.3.2 Algorithm HPs

Each learning algorithm usually has several HPs, which are specified by the software package used and can have a large impact on its complexity, speed, and other important properties of the algorithm (Bischl et al., 2023). For example, the HPs of the CART algorithm include the minimum number of observations in any terminal node ($\lambda_{minbucket}$), the maximum tree depth, with the root node counted as depth 0 ($\lambda_{maxdepth}$), and the factor by which a split needs to decrease the overall lack of fit to be attempted (λ_{cp}) (Therneau & Atkinson, 2022). In the CART implementation of the `R` package `mlr3` (Lang et al., 2019), the respective HP domains are $\Lambda_{minbucket} = \{1, \ldots, n_{\text{train}}\}$, $\Lambda_{maxdepth} = \{1, \ldots, 30\}$ (both being integer-valued domains), and $\Lambda_{cp} = [0, 1]$ (real-valued domain). Most algorithm HPs have default values that are specified by the software in which they are implemented (e.g., in `mlr3`, $\lambda_{minbucket} = 7$ per default).

Note that since there is usually more than one algorithm suitable for a given prediction problem, the choice of algorithm can also be seen as an HP of the learning pipeline (with the HPs associated with each algorithm representing conditional HPs that are only relevant when the respective algorithm is used; Bischl et al., 2023). This creates an even more flexible but also complex learning pipeline, which is why, in this paper, we assume that the algorithm has already been selected.

2.3.3 Preprocessing HPs

As mentioned above, it is not only possible to specify learning algorithm HPs but also preprocessing HPs (Binder & Pfisterer, 2024; Bischl et al., 2023). In principle, whenever multiple options exist for performing a preprocessing step, these options can be considered as different HP values of the respective preprocessing step.

First, the choice of whether a preprocessing step PS is applied at all can be considered as a binary HP λ_{PS} with $\Lambda_{PS} = \{\text{yes}, \text{no}\}$ (e.g., whether features should be log-transformed or not).

Second, there is often more than one possible option for performing a preprocessing step. For example, the influence of outliers in features can be reduced by replacing all values that are outside the range $[x_{min}, x_{max}]$ by x_{min} and x_{max}, respectively ("winsorizing"; Steyerberg, 2019). There are different options to specify x_{min} and x_{max}, which means that $\lambda_{x_{min}}$ and $\lambda_{x_{max}}$ are HPs of the winsorizing preprocessing step (e.g., Steyerberg, 2019, suggests percentiles such as $\lambda_{x_{min}} = 1\text{st percentile}$ and $\lambda_{x_{max}} = 99\text{th percentile}$).

Several possible options also exist for the imputation of missing feature values. For example, imputation can be based on the feature's mean or median, or on a sampled value from its empirical distribution (as illustrated in Thomas, 2024). This constitutes a (categorical) preprocessing HP λ_{impute} with $\Lambda_{impute} = \{\text{mean}, \text{median}, \text{sample}, \ldots\}$.

Another typical example of a preprocessing step with many possible options is feature selection. To define HPs in this context, we have to differentiate between filter and wrapper methods (the following explanations are based on Wright, 2024, who also provides more

details and additional examples). Filter methods are preprocessing steps that assign a numeric score to each feature (e.g., the correlation coefficient ρ between each feature and the outcome) and select a set of features according to this score (e.g., all features with $\rho > 0.2$). Consequently, the set of selected features is the parameter of the filter (i.e. θ_{filter}, with, e.g., $\hat{\theta}_{filter} = \{x_6, x_8, x_{21}, x_{25}\}$), while its specific configuration can be modified by its HPs. For example, there are different options to define the score (λ_{filter_1}, with $\Lambda_{filter_1} = \{\text{correlation, variance, importance score}, \dots\}$) and to select the features based on their score (λ_{filter_2}, with $\Lambda_{filter_2} = \{\text{top } r \text{ features, all features with a score} \geq \tau, \dots\}$, where r and τ themselves are HPs that are conditional on λ_{filter_2}). Instead of using filter methods, it is also possible to directly specify the set of features that should be selected. In this case, the set of selected features is an input rather than an output of the learning pipeline and is therefore the HP ($\lambda_{features}$) of the feature selection step. For example, if only the features x_6, x_9, and x_{21} should be used by the learning algorithm, then $\lambda_{features} = \{x_6, x_9, x_{21}\}$. In many applications, $\lambda_{features}$ is not specified once by the user, but different values of $\lambda_{features}$ are tried and evaluated on $\mathcal{D}_{\text{train}}$. This process is referred to as a wrapper method but is, in fact, a special case of HP tuning, which will be discussed in Section 4.1.

Note that the individual HP values can also be application-specific. For example, in the real-world prediction problem considered in Section 5, several options for aggregating 17 individual features covering physical symptoms, psycho-social burden, family needs, and practical problems of palliative care patients to a sum score are reasonable (see Section 5.2.2).

In addition to specifying the preprocessing steps, the order in which they appear in the learning pipeline can technically be considered an HP as well. For instance, in the learning pipeline shown in Figure 1, the log-transformation step could also be applied before the imputation step, resulting in a different $\hat{\theta}_{impute}$ and, therefore, potentially a different prediction model. However, we will not consider this type of preprocessing HP further in the remainder of this paper.

As already indicated by the examples above, many preprocessing HPs are conditional on other preprocessing HPs (e.g., the winsorizing HPs $\lambda_{x_{min}}$ and $\lambda_{x_{max}}$ are only relevant when winsorizing is the chosen method to reduce the influence of feature outliers, which could also be implemented by transforming the features instead). Moreover, in contrast to algorithm HPs, preprocessing HPs often cannot be set by a single software function argument (for example, all HPs of the CART algorithm named in the previous section can be specified within a single `R` function, using, e.g., the argument `minbucket` for $\lambda_{minbucket}$); instead, in many cases, the different options for a specific preprocessing step are implemented by different software packages. Consequently, there is often no formal HP domain, and defining the domain such that it contains all possible HP values may not even be feasible (e.g., for λ_{impute}, defining Λ_{impute} would require collecting all available methods for imputing missing values). Moreover, many preprocessing HPs do not have a formal default value, although the option of not applying a preprocessing step (if applicable and not leading to an error) seems to be a reasonable default value that we will adopt in the following.

In contrast to algorithm HPs, it seems that preprocessing HPs—apart from those related to feature selection—are rarely discussed or referred to as such in ML applications (see, e.g., the systematic reviews of Dhiman et al., 2022a, and Andaur Navarro et al., 2023, where such terms were not mentioned). ML methods research usually also focuses on algorithm HPs rather than preprocessing HPs. An exception is the benchmark study by Stüber et al. (2023), which, among other factors, examines the impact of using principal component analysis in radiomics-based survival analysis.

2.3.4 Selection of HPs

While it is usually possible to leave all HPs at their respective default value, it is common to modify them in an attempt to optimize the prediction model generated by the learning pipeline. This can also be necessary if there is no specified default value. The term "optimization" here often refers to the predictive performance of the model but can also take into account other criteria such as simplicity, interpretability, or runtime to generate the model (Bischl et al., 2023; de Hond et al., 2022; Domingos, 2012; Pfob et al., 2022). Note that the selection of HPs can be considered a "researcher degree of freedom" (Simmons et al., 2011), as it is one of many choices that users must make throughout the model development process (other choices are, e.g., how predictive performance is assessed; Hofman et al., 2017; Hosseini et al., 2020; Klau et al., 2020). We can distinguish between two primary types of HP selection: data-independent and data-dependent procedures. Data-independent HP selection does not make use of the data set $\mathcal{D}_{\text{train}}$ and is ideally based on the user's knowledge about the data set and learning algorithm. For example, sensible algorithm HPs can be selected when users are experienced with the learning algorithm or when corresponding recommendations from the literature (e.g., previous benchmark studies) are available (Bartz et al., 2023; Bischl et al., 2023). Similarly, some preprocessing HPs may be inferred from substantive knowledge about the data set (e.g., which set of features should be selected) or knowledge about how the learning algorithm is affected by certain data set characteristics (e.g., whether the algorithm is sensitive to outliers in features, which requires some form of transformation; Kuhn and Johnson, 2013). An example of data-independent HP selection on the basis of model simplicity is the specification of the maximum tree depth in the real-world prediction problem considered in Section 5, where the project team set the HP to $\lambda_{maxdepth} = 4$ to ensure that the resulting decision tree can be implemented in clinical practice. In cases where users have insufficient knowledge about the data and learning algorithm to ensure a reasonable HP selection but wish to avoid arbitrary or default HP values, it is possible to use the data set $\mathcal{D}_{\text{train}}$ to select optimal HP values. This process corresponds to a data-dependent HP selection, but terms such as HP tuning and (data-driven) HP optimization are more common (e.g., Bartz et al., 2023; Bischl et al., 2023; Probst et al., 2019). We will accordingly use the term HP tuning in the remainder of this paper. Note that HP tuning implies that not only the parameters $\boldsymbol{\theta}$ are estimated from the data set $\mathcal{D}_{\text{train}}$ but also one or more HPs in $\boldsymbol{\lambda}$. HP tuning thus generally complicates model generation and evaluation, which will be described in more detail in Section 4.

Importantly, there are HPs that should not be selected through tuning. For learning algorithms, this includes, for example, the number of trees ($\lambda_{num.trees}$) in the random forest algorithm for classification problems: Due to the monotonous relation between $\lambda_{num.trees}$ and model performance in most cases, the largest computationally feasible number of trees should be chosen (Probst & Boulesteix, 2018). Regarding preprocessing HPs, this typically applies to those associated with steps that alter the scope or interpretation of the prediction model (see Section 2.2.2). As such steps require careful specification, the corresponding HPs should be set based on user expertise (i.e. data-independently) rather than determined through tuning.
To indicate how the value of a HP λ_j has been specified, we write λ_j^{I} if the value is left at default value or selected independently of the data, and λ_j^{II} if the value was chosen through tuning.

2.4 Model development processes

The development of ML-based prediction models generally involves two key processes: (i) the generation of the prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ (model generation) and (ii) the evaluation of its predictive performance (model evaluation). Given our focus on HPs and their selection, we distinguish between two settings in the remainder of this paper. In Setting I, all HPs of the learning pipeline are pre-specified (i.e. either set to default values or selected independently of the data). In Setting II, one or more HPs are selected through tuning.
Before explaining the principles and potential pitfalls of model generation and evaluation for both settings in Sections 3 and 4, we first clarify their general concepts.

2.4.1 Model generation

We refer to the model generation process as the set of processes required to obtain the final prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$. In Setting I, the model generation process consists of a single training process, where the parameters that define the final prediction model are estimated from $\mathcal{D}_{\text{train}}$ using the learning pipeline $\mathcal{I}$ with pre-specified HPs. In Setting II, where one or more HPs are selected through tuning, the model generation process consists of a tuning process conducted on $\mathcal{D}_{\text{train}}$ (which yields the tuned HPs), followed by a training process, where, similar to Setting I, the parameters of the final prediction model are estimated from $\mathcal{D}_{\text{train}}$ using the learning pipeline $\mathcal{I}$ with tuned HPs.

2.4.2 Model evaluation

Once the final prediction model $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ has been generated, the next important step is its evaluation. Since many algorithms yield black-box models that cannot be easily interpreted, and are thus difficult to assess for plausibility without additional tools (see, e.g., Molnar, 2022), a key quantity in the evaluation of a model is its prediction error. In the context of this work, we will accordingly use the term "model evaluation" synonymously with determining a model's prediction error. The prediction error indicates how well a model performs on new observations that are independently drawn from the same distribution as the observations in $\mathcal{D}_{\text{train}}$ (i.e. from $\mathbb{P}_{xy}$). It is specified with respect to a loss function L, which assesses the discrepancy between true outcomes and predictions and constitutes the performance measure. Formally,

the prediction error of $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ can be defined as

$$\text{PE}(\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}) = \text{E}_{(\boldsymbol{x},y)\sim\mathbb{P}_{xy}}[L(\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}(\boldsymbol{x}), y)] \tag{2}$$

(Bischl et al., 2023; Boulesteix et al., 2015; Hastie et al., 2009). The loss function L can be chosen according to the prediction problem being addressed. For instance, a common choice for L in regression problems is the squared loss. In this case, the prediction error reflects the well-known mean squared error (MSE). Note that in equation (2), we assume for simplicity that L corresponds to a point-wise loss function, although many commonly used performance measures (e.g., the area under the receiver operating characteristic curve, AUC) would necessitate a more general definition (provided in Bischl et al., 2023). Nonetheless, all following statements regarding the prediction error hold regardless of this simplified (and more common) representation.

An estimate of the prediction error in equation (2) can be obtained by using $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ to make predictions for an additional data set with new observations drawn from $\mathbb{P}_{xy}$ (referred to as test data set $\mathcal{D}_{\text{test}}$). The prediction error can then be estimated by evaluating the loss function L for each observation and calculating the average across all observations (again, assuming a point-wise loss; Bischl et al., 2023; Hastie et al., 2009). The resulting prediction error estimate for $\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}$ can be denoted as $\widehat{\text{PE}}(\hat{f}_{\mathcal{I}}^{\mathcal{D}_{\text{train}}}, \mathcal{D}_{\text{test}})$. Note that the outcome values for $\mathcal{D}_{\text{test}}$ must be observed; otherwise, the loss function L cannot be evaluated.

The requirement for an additional data set, $\mathcal{D}_{\text{test}}$, for model evaluation can be challenging in applications where data resources are limited. Denoting $\mathcal{D}$ as the only available data set at the time of model generation and evaluation, there are two general approaches for defining $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$: (i) all available data are used for model generation, in which case $\mathcal{D}_{\text{test}}$ is inevitably a subset of $\mathcal{D}_{\text{train}}$ (i.e. $\mathcal{D}_{\text{train}} = \mathcal{D}$ and $\mathcal{D}_{\text{test}} \subseteq \mathcal{D}_{\text{train}}$), or (ii) the model is generated on a (proper) subset of the available data, with the remaining subset held back for model evaluation (i.e. $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ and $\mathcal{D}_{\text{test}} = \mathcal{D} \setminus \mathcal{D}_{\text{train}}$). For the first approach, there are several ways to define $\mathcal{D}_{\text{test}}$, each leading to a different evaluation procedure, which will be detailed in Section 3.2 (Setting I) and Section 4.2 (Setting II).

Depending on the chosen evaluation procedure, a potential issue can be data leakage, which occurs whenever information about the designated $\mathcal{D}_{\text{test}}$ is improperly available during the generation of the model to be evaluated (Hornung et al., 2023; Kapoor & Narayanan, 2023; Kapoor et al., 2024; Kaufman et al., 2012; Rosenblatt et al., 2024). Since, in this case, the observations in $\mathcal{D}_{\text{test}}$ no longer truly represent new observations to which the model will be applied, and the model thus has an unfair advantage when predicting these observations, the resulting prediction error estimate can be optimistically biased. Kapoor and Narayanan, 2023 identify three general types of data leakage, which may arise from: (i) overlap between the data used for model generation and evaluation, (ii) violation of the assumption that all observations are independently drawn from the same distribution, or (iii) use of illegitimate features. In this paper, we will focus on overlap-induced data leakage but provide additional information on the

other two types in Supplementary Section A. Furthermore, we encounter an example of one of the other types in our empirical illustration in Section 5.

Finally, note that in some applications of ML (e.g., in the context of healthcare research), the process of assessing a model's performance on observations from $\mathbb{P}_{xy}$ is referred to as internal validation. This is in contrast to external validation, which evaluates how well the model predicts observations from different distributions (e.g., different time points or healthcare settings; Collins, Dhiman, et al., 2024; de Hond et al., 2022; Van Calster et al., 2023; van Royen et al., 2023). As external validation is recommended to be performed in subsequent research only after successful internal validation (Collins, Dhiman, et al., 2024), we will focus on internal validation in this paper. Note that, in general, the term "evaluation" should be preferred over "validation" as the latter suggests that a "validated model" has a low prediction error, which is not necessarily the case (Collins, Dhiman, et al., 2024).

3 Setting I: Pre-specified HPs

In this section, we describe the model generation and evaluation process for Setting I. We accordingly assume that the learning pipeline $\mathcal{I}$ is configured by HP values that are either set to their default values or selected independently of the data, i.e. $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{I}}$. This aspect is emphasized by denoting the learning pipeline as $\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}$.

3.1 Model generation

As stated in Section 2.4, the model generation process in Setting I consists of a single training process. Moreover, as already outlined, "training" refers to the learning pipeline estimating the parameters $\boldsymbol{\theta}$ (which constitute the prediction model) from $\mathcal{D}_{\text{train}}$. For brevity, we will also refer to this process as "training the prediction model" although it is the learning pipeline that is being trained and subsequently yields the prediction model.

Importantly, all parameters in $\boldsymbol{\theta}$ must be estimated, including those from preprocessing steps. The estimation of preprocessing parameters follows the sequence of their corresponding steps in the learning pipeline $\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}$. This process is specified by the respective preprocessing step. For example, in the case of mean imputation, the corresponding parameter estimate is found by calculating the mean of all non-missing observations of the corresponding feature.

The parameters of the learning algorithm are usually estimated based on a loss function l that measures the discrepancy between the true outcome and a prediction vector for each observation i, i.e. $l(y^{(i)}, f(\boldsymbol{x}^{(i)}))$. The algorithm parameters are then found by minimizing $\sum_{i=1}^{n_{\text{train}}} l(y^{(i)}, f(\boldsymbol{x}^{(i)}))$ (see, e.g., Bischl et al., 2023, or Bartz et al., 2023, for more details). For example, in a regression problem where the learning algorithm corresponds to the CART algorithm, the splitting rules are found by minimizing the sum of squared errors and the prediction value for each terminal node corresponds to the mean of all outcome values in the respective node (Breiman et al., 1984). Note that the loss function l may, but does not necessarily have to, align with the loss function L from Section 2.4.2, which is used to estimate the prediction

error.

When estimating the parameters, the learning pipeline may not only capture the signal in $\mathcal{D}_{\text{train}}$ which represents the true underlying data-generating mechanism $\mathbb{P}_{xy}$, but it may also erroneously learn the specific pattern of noise (i.e. unexplained variation) in $\mathcal{D}_{\text{train}}$. The resulting prediction model is too adapted to $\mathcal{D}_{\text{train}}$ and will perform worse on new observations (drawn from $\mathbb{P}_{xy}$) than on the observations in $\mathcal{D}_{\text{train}}$. This is a well-known problem in prediction model training and is commonly referred to as overfitting (e.g., Bischl et al., 2023; de Hond et al., 2022; Hastie et al., 2009; Kuhn & Johnson, 2013; Poldrack et al., 2020; Steyerberg, 2019). The risk of obtaining an overfitted prediction model depends on both the data set $\mathcal{D}_{\text{train}}$ (specifically on its signal-to-noise ratio, which tends to decrease as the number of observations decreases) and on the learning pipeline $\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}$ used to train the model (Lones, 2024; Poldrack et al., 2020). The association between the characteristics of a learning pipeline and its tendency to overfit is not straightforward, but it is related to factors such as the size of its hypothesis space (i.e. the number of prediction models that can be trained by $\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}$) and the procedure by which the model is chosen from the hypothesis space (e.g., whether the hypothesis space is searched exhaustively; Domingos, 2012). These factors can vary greatly between learning pipelines, especially depending on the type of learning algorithm and the chosen HP values. Note that the learning pipeline may also suffer from underfitting rather than overfitting, which occurs if it is not flexible enough to adequately model the underlying data-generating mechanism (Hastie et al., 2009).

As mentioned above, after training the learning pipeline once (and only once) on $\mathcal{D}_{\text{train}}$, the generation of the final prediction model is completed. This implies that if the model is found to have a poor predictive performance in the subsequent evaluation (e.g., due to over- or underfitting), the result either has to be accepted or the HPs of the learning pipeline have to be modified based on the evaluation result. However, users should be aware that the latter approach corresponds to Setting II, which has different implications for model evaluation (Section 4). We denote the final prediction model as $\hat{f}^{\mathcal{D}_{\text{train}}}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}}$ to emphasize that it is the result of training a learning pipeline configured with HP values $\boldsymbol{\lambda}^{\text{I}}$.

3.2 Model evaluation

As outlined in Section 2.4.2, evaluating the prediction model $\hat{f}^{\mathcal{D}_{\text{train}}}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}}$ requires a test data set $\mathcal{D}_{\text{test}}$, which is used to estimate the model's prediction error. In that section, it was also stated that evaluation procedures can be differentiated based on whether model generation (which corresponds to model training in Setting I) has been performed on all available data (with $\mathcal{D}_{\text{train}} = \mathcal{D}$ and $\mathcal{D}_{\text{test}} \subseteq \mathcal{D}_{\text{train}}$) or only on a (proper) subset of the available data (with $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ and $\mathcal{D}_{\text{test}} = \mathcal{D} \setminus \mathcal{D}_{\text{train}}$). In the following sections, we examine the implications for model evaluation in more detail for both approaches. An additional graphical overview is provided in Figure 2.

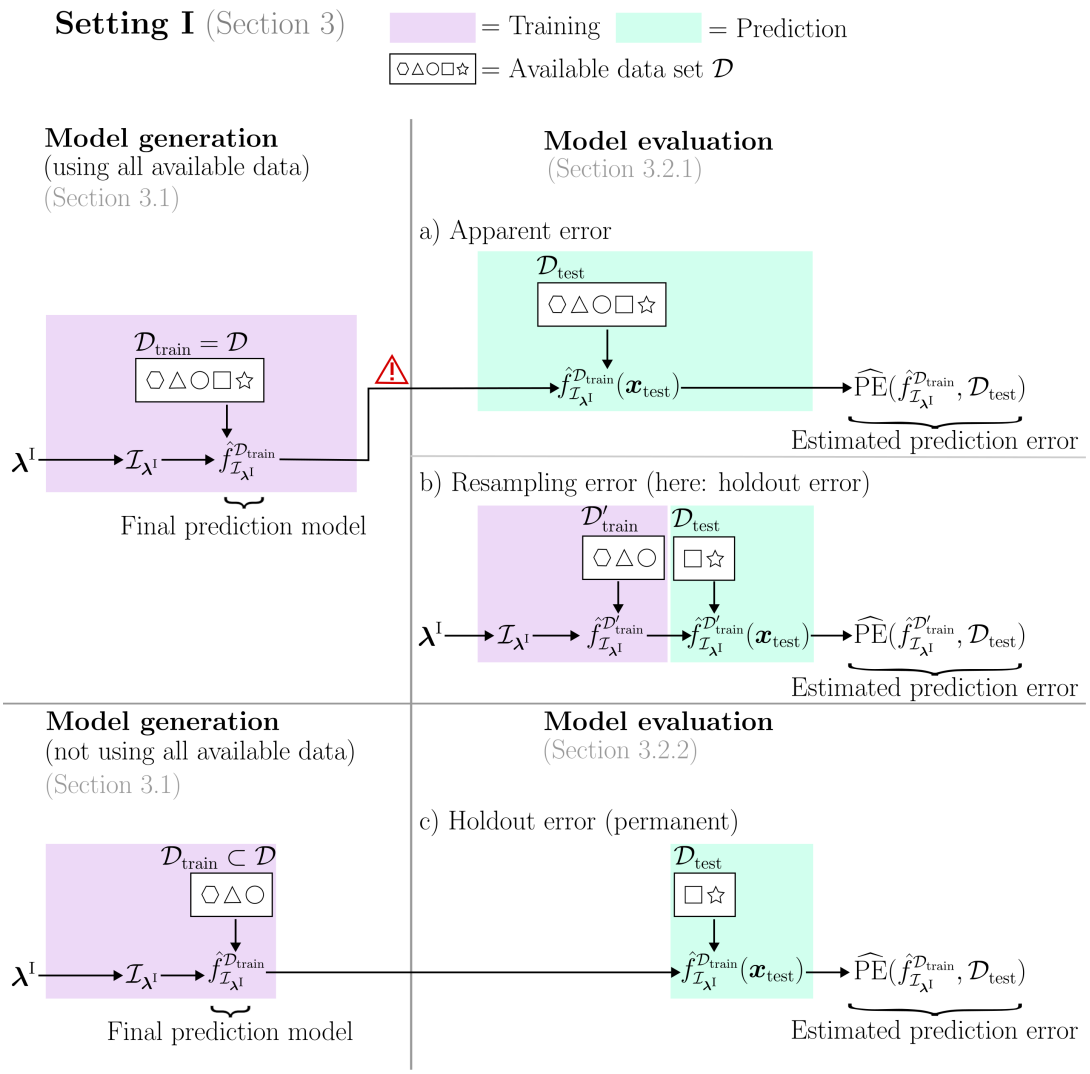

Figure 2: Overview of different model evaluation procedures and their relation to the model generation process if all HPs are pre-specified. Data leakage is present if any subset of $\mathcal{D}_{\text{test}}$ used for prediction error estimation has also been employed to generate the evaluated prediction model (which is not necessarily the final model). In the figure, the point at which data "leaks" into the model evaluation is marked by the red caution symbol.

3.2.1 Evaluation of a model generated on all available data

Apparent error A straightforward way to evaluate a prediction model trained on all available data is to estimate its prediction error using the same data set, i.e. $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{test}} = \mathcal{D}$. The resulting prediction error estimate is referred to as apparent error (see Figure 2, model evaluation a). As explained in Section 2.4.2, data leakage is present when information about the designated $\mathcal{D}_{\text{test}}$ is present during model generation. For the apparent error, this is clearly the case, as $\mathcal{D}_{\text{test}}$ is equal to $\mathcal{D}_{\text{train}}$. As a consequence, the apparent error is not able to detect any overfitting of the model (since the specific pattern of noise in $\mathcal{D}_{\text{train}}$ exactly corresponds to that in $\mathcal{D}_{\text{test}}$) and will therefore be affected by a (possibly substantial) optimistic bias. Although this evaluation

procedure is well-known to be flawed and has been frequently warned against in literature (e.g., Collins, Dhiman, et al., 2024; Efron, 1986; Hastie et al., 2009; Kuhn & Johnson, 2013; Poldrack et al., 2020), it is often still the only prediction error estimate that is reported in studies presenting new prediction models (Kapoor & Narayanan, 2023; Poldrack et al., 2020).

Resampling error To avoid the optimistic bias caused by the overlap between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, several procedures exist that partition $\mathcal{D}_{\text{train}}$ one or multiple times into two subsets for evaluation purposes while still training the final prediction model on the full data set. These procedures can be referred to as resampling methods and the resulting estimate as the resampling error (see Figure 2, model evaluation b). The following description is based on Simon, 2007, Kuhn and Johnson, 2013, Bischl et al., 2023, and Casalicchio and Burk, 2024; see their work for more details.

The simplest resampling method is the holdout or split-sample method, where $\mathcal{D}_{\text{train}}$ is randomly split into two subsets with different purposes: One subset, denoted as $\mathcal{D}'_{\text{train}}$, is used to retrain the same learning pipeline $\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}$ that has been used to obtain the final prediction model. This results in an additional prediction model $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}}^{\mathcal{D}'_{\text{train}}}$, whose prediction error is then estimated on the second subset, which serves as $\mathcal{D}_{\text{test}}$. The holdout method essentially has two drawbacks, whose impact on the prediction error varies according to the split ratio and the absolute number of observations in $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ (denoted as n'_{train} and n_{test}). First, while the holdout method ensures a clean separation between $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, it does not evaluate the actual prediction model trained on $\mathcal{D}_{\text{train}}$ but the additional prediction model trained on $\mathcal{D}'_{\text{train}}$, which does not necessarily coincide with the former. Since the additional prediction model is trained on fewer observations (i.e. $n'_{\text{train}} < n_{\text{train}}$), estimating its prediction error on $\mathcal{D}_{\text{test}}$ yields a pessimistically biased estimate for the prediction error of $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{I}}}}^{\mathcal{D}_{\text{train}}}$. Second, the smaller n_{test}, the more the prediction error estimate varies depending on which observations are assigned to $\mathcal{D}_{\text{test}}$ (i.e. the higher the variance of the holdout estimator). As a consequence, specifying the split ratio for the holdout method requires a careful trade-off between bias and variance.

A commonly used variation of holdout is k-fold cross-validation (CV), where $\mathcal{D}_{\text{train}}$ is randomly split into k subsets (or folds) of approximately the same size, with 5 or 10 being typical choices for k. Based on the k splits, the procedure described for the holdout method is repeated k times: In each repetition (in this context also referred to as resampling iteration), the learning pipeline is trained on $k-1$ subsets of $\mathcal{D}_{\text{train}}$ (constituting $\mathcal{D}'_{\text{train}}$), and the prediction error of the resulting model is estimated on the remaining subset (constituting $\mathcal{D}_{\text{test}}$). The final prediction error estimate is obtained by averaging the k prediction error estimates, which leads to the CV estimator having a smaller variance than a holdout estimator with the same split ratio. However, the prediction error estimate resulting from CV is also pessimistically biased because the evaluated prediction models are again trained on less than n_{train} observations, although this bias decreases with increasing k ($n'_{\text{train}} = \frac{k-1}{k} \cdot n_{\text{train}}$).

Other common resampling methods include repeated versions of holdout and CV (to reduce

the variance of the corresponding estimator) and bootstrapping. Repeated holdout and bootstrapping are similar in their execution, except that for repeated holdout, the observations constituting $\mathcal{D}'_{\text{train}}$ in each resampling iteration are drawn without replacement, while they are drawn with replacement for bootstrapping.

As stated above, all resampling methods require the learning pipeline to be retrained on one or multiple subsets $\mathcal{D}'_{\text{train}}$, each of which is a (proper) subset of $\mathcal{D}_{\text{train}}$ (i.e. $\mathcal{D}'_{\text{train}} \subset \mathcal{D}_{\text{train}}$). In this context, a flawed evaluation procedure would be to apply all preprocessing steps on the full data set $\mathcal{D}_{\text{train}}$ and retrain only the learning algorithm on $\mathcal{D}'_{\text{train}}$ during resampling. This "incomplete resampling" (Simon et al., 2003) results in another form of data leakage, as in each resampling iteration, the observations in the respective $\mathcal{D}_{\text{test}}$ subset have already been used to train part of the learning pipeline (i.e. the preprocessing steps). Incomplete resampling has been frequently warned against in the literature (e.g., de Hond et al., 2022; Hofman et al., 2023; Kapoor et al., 2024; Pfob et al., 2022; Poldrack et al., 2020), and the resulting optimistic bias has been demonstrated by illustrations on real data (e.g., Hornung et al., 2015; Rosenblatt et al., 2024) and corrected reanalyses of published studies (e.g., Kapoor & Narayanan, 2023; Neunhoeffer & Sternberg, 2019). Yet, it still seems to be a common pitfall in the evaluation of prediction models (see Kapoor and Narayanan, 2023, and references therein), which is probably caused by a lack of understanding of its implications. In addition, if the learning pipeline is not implemented as a single object that can be trained with a single function call such as `train(learning_pipeline)` (e.g., this is possible in `R` with the `mlr3` or `recipes` package by Lang et al., 2019, and Kuhn et al., 2024), each preprocessing step must be manually repeated in every resampling iteration. In such cases, users may consider incomplete resampling a time-saving shortcut, without realizing that it introduces data leakage. To avoid incomplete resampling, every component of the learning pipeline, including the preprocessing steps, must be retrained in each resampling iteration. The only preprocessing steps that can be safely applied to the full data set prior to resampling are those that are both parameterless and precede the first parameterized preprocessing step in the learning pipeline.

3.2.2 Evaluation of a model generated on a subset of the available data

If the final prediction model has been trained on a subset of the available data (i.e. $\mathcal{D}_{\text{train}} \subset \mathcal{D}$), its prediction error can be estimated using the remaining observations as $\mathcal{D}_{\text{test}}$ (see Figure 2, model evaluation c). This means that the training process does not need to be repeated, as there is no need to use resampling methods. Note that this procedure is technically equivalent to the holdout method introduced above, except that the model trained on $\mathcal{D}_{\text{train}}$, which corresponds to $\mathcal{D}'_{\text{train}}$ in the holdout method above, is the final prediction model and has not only been trained for evaluation purposes. Accordingly, the procedure is referred to as holdout or split-sample method as well, which can make it difficult to infer which procedure was used when the evaluation result of a model is reported. We use the terms temporary holdout (described in Section 3.2.1) and permanent holdout (described here) to distinguish the two procedures.

In principle, most points discussed in the previous section affecting temporary holdout (including

data leakage due to incomplete resampling) also apply to permanent holdout. Again, the only difference is that, for the temporary holdout, the model trained on a subset of the available data is used solely for evaluation purposes, whereas it serves as the final prediction model for the permanent holdout. Consequently, the prediction error estimate derived from the permanent holdout is not pessimistically biased; instead, it is an unbiased estimate of a prediction error that is indeed higher (i.e. worse) than that of a model using all available data. Since not using all available data for training the prediction model essentially corresponds to a loss of important information, the permanent holdout method is only recommended if the number of observations in $\mathcal{D}$ is sufficiently large or if repeating the training process is computationally expensive or infeasible (Collins, Dhiman, et al., 2024).

4 Setting II: HPs selected through tuning

In this section, we review the model generation and evaluation process for Setting II, where one or more HPs are selected through tuning.

4.1 Model generation

4.1.1 Overview

HP tuning generally aims to improve the predictive performance of a model (Bischl et al., 2023; Probst et al., 2019). Using the terminology introduced in Section 2.4.2, this corresponds to finding the HP configuration that minimizes the model's prediction error. To simplify notation, we will assume for now that all HPs are to be tuned, but will revisit the scenario where this does not apply later in this section. Under this assumption, the HP tuning problem can be formalized as:

$$\boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}}{\operatorname{argmin}} \operatorname{PE}(\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}), \tag{3}$$

where $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}$ is the final prediction model resulting from training the learning pipeline $\mathcal{I}$ configured with HPs $\boldsymbol{\lambda}$, and $\boldsymbol{\lambda}^*$ denotes the theoretical optimum (Bischl et al., 2023). The lowest prediction error (i.e. the best performance) that can be achieved using $\boldsymbol{\lambda}^*$ as HP configuration depends on several factors, such as the HPs to be tuned, the selected learning algorithm, the performance measure, and the prediction problem in general (Probst et al., 2019). Note that in the following, we refer to the prediction error of a model that results from training a learning pipeline determined by a candidate HP configuration $\boldsymbol{\lambda}^{(c)}$, i.e. $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}^{(c)}}}^{\mathcal{D}_{\text{train}}}$, simply as the prediction error of $\boldsymbol{\lambda}^{(c)}$ for brevity. It should also be noted that equation (3) represents the standard case of single-objective HP tuning, i.e. the optimization is performed with respect to one performance measure. However, HP tuning can also be conducted based on multiple performance measures or additional criteria such as model simplicity (Bischl et al., 2023; Dunias et al., 2024). Since such multi-objective HP tuning poses further challenges, we will only consider single-objective tuning in this paper.

While there exist different tuning procedures, the general model generation process involving

tuning can be described as follows: Given a set of C candidate HP configurations (selected before or during the tuning process), each HP configuration $\boldsymbol{\lambda}^{(c)}$ $(c = 1, \ldots, C)$ is evaluated on $\mathcal{D}_{\text{train}}$ by employing one of the model evaluation procedures introduced in Section 3.2.1. Accordingly, $\mathcal{D}_{\text{train}}$ is split into $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ (either once or multiple times), which are then used for training ($\mathcal{D}'_{\text{train}}$) and prediction error estimation ($\mathcal{D}_{\text{test}}$). In other words, the model evaluation that is performed once with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{I}}$ in Setting I to assess the prediction error of the final prediction model is performed multiple times for each candidate configuration (i.e. with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(c)}$) in the tuning process of Setting II. After having evaluated all candidate HP configurations, the HP configuration with the lowest (i.e. best) prediction error estimate is used as the final HP configuration. Following the notation introduced in Section 2.3.4, we refer to this configuration as $\boldsymbol{\lambda}^{\text{II}}$. Note that $\boldsymbol{\lambda}^{\text{II}}$ is also commonly denoted as $\hat{\boldsymbol{\lambda}}$, since it is an estimate of $\boldsymbol{\lambda}^*$ (Bischl et al., 2023). However, we adhere to $\boldsymbol{\lambda}^{\text{II}}$ to clearly distinguish it from Setting I, where $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{I}}$. After setting $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{II}}$, the learning pipeline $\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}$ undergoes a final training on $\mathcal{D}_{\text{train}}$, which yields the final prediction model $\hat{f}^{\mathcal{D}_{\text{train}}}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}}$.

Note that while the tuning process already results in a prediction error estimate for the final prediction model (the estimate based on which $\boldsymbol{\lambda}^{\text{II}}$ was selected during tuning), this value is not necessarily adopted as the final model evaluation result, as we will discuss in Section 4.2. In fact, it is also possible to use different performance measures for the prediction error estimation performed during tuning and the evaluation of the final model, but, for the sake of simplicity, we will assume that they are the same.

To summarize, during the model generation in Setting II, both the HPs $\boldsymbol{\lambda}$ and the parameters $\boldsymbol{\theta}$ of the final prediction model are optimized using the data set $\mathcal{D}_{\text{train}}$. However, the optimization is not performed jointly: first, the HPs $\boldsymbol{\lambda}$ are optimized in the tuning process. Second, the parameters $\boldsymbol{\theta}$ are optimized in one (final) training process. Note that HPs are still an input of the learning pipeline but can be seen as an output of the tuning process.

If only a subset of the HPs $\boldsymbol{\lambda}$ are to be tuned, the tuning process described above is applied exclusively to those HPs, while the pre-specified HPs remain fixed throughout the process. For example, assume that from all J HPs in $\boldsymbol{\lambda}$, the HPs $\boldsymbol{\lambda}_{1:j} = \lambda_1, ..\lambda_j$ are pre-specified and the HPs $\boldsymbol{\lambda}_{j+1:J} = \lambda_{j+1}, \ldots, \lambda_J$ are to be tuned. In this case, the tuning process yields a HP configuration $\boldsymbol{\lambda}^{\text{II}}_{j+1:J}$, and the final prediction model is trained with $\boldsymbol{\lambda}_{1:j} = \boldsymbol{\lambda}^{\text{I}}_{1:j}$ and $\boldsymbol{\lambda}_{j+1:J} = \boldsymbol{\lambda}^{\text{II}}_{j+1:J}$. Since the tuning process is conceptually the same when not all HPs are optimized—untuned HPs are simply kept fixed—we will continue to assume that all HPs are tuned to maintain notational simplicity.

When choosing a tuning procedure, it is important to consider that the tuning process is limited in terms of both data availability and computation time: First, as outlined above, each candidate HP configuration, $\boldsymbol{\lambda}^{(c)}$, is evaluated using one of the evaluation procedures described in Section 3.2.1 for Setting I. As explained there, the specified $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ subsets contain a limited number of observations (i.e. n'_{train} and $n_{\text{test}} \leq n_{\text{train}}$) and could overlap, potentially leading to unreliable prediction error estimates for each $\boldsymbol{\lambda}^{(c)}$. Second, the computational bud-

get available for the tuning process is typically limited, which restricts both the number of evaluated HP configurations and the time spent evaluating each configuration (i.e. estimating its prediction error). Due to these limitations and the resulting trade-offs (discussed in more detail in Section 4.1.3), choosing an adequate tuning procedure is often non-trivial. Yet, guidance is still lacking, and many of the existing recommendations are based on rules of thumb rather than empirical benchmarks (see Bischl et al., 2023, for an overview). Inadequate tuning procedures can result in a $\boldsymbol{\lambda}^{\text{II}}$ that yields a final prediction model with worse prediction error than $\boldsymbol{\lambda}^*$ (potentially even worse than setting all HPs to their default values) and/or an overly time-consuming tuning process (i.e. a more efficient tuning procedure could have achieved the same prediction error in less time).

4.1.2 Automated vs. manual tuning

Before describing different tuning procedures in more detail, we note that their specification generally depends on whether the tuning process is fully automated or performed manually. We consider the tuning process as automated if the relevant tuning components only need to be specified as a function argument, which is possible in several ML software frameworks (see Bischl et al., 2023, for an overview). In contrast, we refer to the tuning process as manual if the candidate HP configurations are evaluated by repeatedly calling the same function(s), altering only the argument that specifies the HP configuration.

Compared to automated tuning, manual tuning is more time-consuming, error-prone, and less reproducible, as it is usually an informal and unsystematic process. On the other hand, automated tuning is usually more difficult to implement and requires more programming expertise than manual tuning. As a consequence, although manual tuning is generally advised against (e.g., Bartz et al., 2023; Bischl et al., 2023), it is likely still a common yet often unreported approach in many ML applications (Hofman et al., 2023; Hosseini et al., 2020; Lones, 2024). Note that this may be particularly true for the tuning of preprocessing HPs $\boldsymbol{\lambda}_P$: As discussed in Section 2.3.3, preprocessing HPs are often not identified as HPs. Consequently, users trying out different preprocessing options might not be aware that this corresponds to (manual) HP tuning and could be automated. Moreover, if the HPs to be tuned include application-specific preprocessing HPs, the barrier to using automated tuning is further increased, as these HPs may not yet be integrated into the corresponding software and require custom implementation. As a consequence, given the potentially different characteristics of the tuned HPs (especially preprocessing HPs $\boldsymbol{\lambda}_P$ vs. algorithm HPs $\boldsymbol{\lambda}_A$), we cannot rule out that in practice, they are selected by a combination of automated and manual tuning (see Section 5.2.3 for a concrete example).

4.1.3 Tuning procedures

As stated above, the selected tuning procedure will affect both the duration of the tuning process and the prediction error of the final prediction model. In the following, we will review the individual components that characterize each tuning procedure and describe how they impact the tuning process.

Search space When tuning an HP λ_j, it is often not reasonable to consider all possible HP values (i.e. all values in Λ_j). For example, this applies if certain values of λ_j are already known to cause overfitting or convergence issues. Moreover, when λ_j is a preprocessing HP, Λ_j may not even be formally specified (see Section 2.3.3). To perform HP tuning, it is thus essential to specify a search space $\tilde{\Lambda}_j$ for each HP, where $\tilde{\Lambda}_j$ is a bounded subset of Λ_j and determines the HP values that are considered for tuning (Bischl et al., 2023). For example, if the HPs of the CART algorithm, λ_{cp} and $\lambda_{minsplit}$ with $\Lambda_{cp} = [0, 1]$ and $\Lambda_{minbucket} = \{1, \ldots, n_{\text{train}}\}$, are tuned, their search spaces could be defined as $\tilde{\Lambda}_{cp} = [0.001, 0.1]$ and $\tilde{\Lambda}_{minbucket} = \{5, \ldots, 25\}$. The (overall) search space of all J HPs is denoted as $\tilde{\boldsymbol{\Lambda}} = \tilde{\Lambda}_1 \times \cdots \times \tilde{\Lambda}_J$.
It is important to consider that defining a search space $\tilde{\boldsymbol{\Lambda}}$ restricts the tuning process to finding the optimal HP configuration within $\tilde{\boldsymbol{\Lambda}}$, denoted as $\tilde{\boldsymbol{\lambda}}^*$, and not within $\boldsymbol{\Lambda}$, i.e. $\boldsymbol{\lambda}^*$. Given a search space $\tilde{\boldsymbol{\Lambda}}$, the tuning problem specified in equation (3) thus updates to

$$\tilde{\boldsymbol{\lambda}}^* = \underset{\boldsymbol{\lambda} \in \tilde{\boldsymbol{\Lambda}}}{\operatorname{argmin}} \operatorname{PE}(\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}). \tag{4}$$

Choosing a search space involves the following trade-off: If the search space is too small, the prediction error achieved by $\tilde{\boldsymbol{\lambda}}^*$ and $\boldsymbol{\lambda}^*$ may differ greatly. On the other hand, if the search space is too large, this decreases the chance of finding $\tilde{\boldsymbol{\lambda}}^*$ (or a HP configuration that leads to a comparable prediction error) within a given computational budget (Bischl et al., 2023).
Note that in contrast to automated tuning, the search space is usually not formally specified when performing manual tuning and may be extended during the tuning process (e.g., when the user initially planned to try two preprocessing options but then comes up with an additional option during tuning).

Termination criterion Unless the specified search space $\tilde{\boldsymbol{\Lambda}}$ is very small, such as when only a few categorical HPs are tuned, evaluating all HP configurations in the search space can be computationally challenging or even infeasible. For example, even if λ_{cp} and $\lambda_{minbucket}$ are the only HPs being tuned, with the search spaces as specified above and $\tilde{\Lambda}_{cp}$ being searched in increments of 0.001, $C = 100 \times 21 = 2{,}100$ candidate HP configurations would need to be evaluated. Accordingly, one or several criteria must be specified to terminate the tuning process once it is met. The trade-off to consider when choosing a termination criterion is that the tuning process should neither stop before finding $\tilde{\boldsymbol{\lambda}}^*$ nor should it continue longer than necessary, which would result in an inefficient use of resources and, as we will discuss below, increase the risk of overtuning (Bischl et al., 2023).
In automated tuning procedures, commonly used criteria are based on the number of evaluations or the runtime. However, additional criteria such as reaching a certain performance level or stagnation of performance might also be reasonable (Bartz et al., 2023; Bischl et al., 2023). Similar termination criteria, though often more intuitive than formally specified, may also exist for manual tuning when, for example, the user stops searching when satisfied by the reached performance level or gives up searching after a certain amount of time.

Search strategy Since, in many cases, only a subset of all HP configurations in the search space can be evaluated before the tuning process is terminated, the way in which the sequence of evaluations is determined, also called search strategy or HPO algorithm (Bischl et al., 2023; Elsken et al., 2019), is another important component of the tuning procedure. Search strategies can be characterized by several aspects, such as the amount of time they spend inferring new candidate HP configurations from already evaluated ones (known as the inference vs. search trade-off; Bischl et al., 2023). For example, search strategies such as evolutionary algorithms and Bayesian optimization consider the distribution and results of previously evaluated HP configurations to propose new configurations. In contrast, the commonly used random search strategy simply draws HP configurations from a predefined, typically uniform, distribution without taking into account past evaluations (see, e.g., Feurer and Hutter, 2019, Bischl et al., 2023, or Bartz et al., 2023, for more details and other search strategies). In the special case where only the set of selected features is tuned, a well-known automated search strategy is backward or forward feature selection (see, e.g., Hastie et al., 2009).
Note that the described search strategies are formally used only in automated tuning, as there is usually no specified search strategy when tuning is conducted manually. However, the results of previous evaluations may still be considered in manual tuning when selecting new HP configurations to evaluate.

Joint vs. sequential tuning In automated tuning procedures, all HPs are usually tuned jointly, i.e. each evaluated HP configuration potentially considers different values of each HP. However, the HPs could also be tuned sequentially, i.e. the complete tuning procedure is repeated for each HP (Probst et al., 2019; Waldron et al., 2011). For example, in a setting with three HPs (i.e. $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$), λ_1 would be tuned first with λ_2 and λ_3 set to default, which yields λ_1^{II}. Then, λ_2 is tuned with $\lambda_1 = \lambda_1^{\text{II}}$ and λ_3 set to its default. Finally, λ_3 is tuned with $\lambda_1 = \lambda_1^{\text{II}}$ and $\lambda_2 = \lambda_2^{\text{II}}$, yielding λ_3^{II}. As sequential tuning does not consider any interaction effects between the HPs, it is generally less likely to yield a $\boldsymbol{\lambda}^{\text{II}}$ comparable to $\tilde{\boldsymbol{\lambda}}^*$ than joint tuning. On the other hand, sequential tuning demands less time, with the maximum number of evaluations increasing linearly rather than exponentially with the number of HPs to tune, as is the case with joint tuning. Hence, it could be a realistic approach for manual tuning.

Prediction error estimation As outlined above, the prediction error of each HP configuration considered for tuning can be estimated using one of the evaluation procedures described in Section 3.2.1. In principle, all issues discussed there also apply to the tuning context. However, instead of leading to potentially invalid performance claims about the final prediction model (which was the case in Section 3.2.1), using an inadequate evaluation procedure for HP tuning initially only increases the risk of failing to select a $\boldsymbol{\lambda}^{\text{II}}$ with a (true) prediction error that is comparable to the prediction error of $\tilde{\boldsymbol{\lambda}}^*$. In other words, if the prediction error of each candidate HP configuration is not estimated adequately, this will initially only affect the model generation process, but not (yet) the evaluation of the final prediction model. Still, the consequences can

be detrimental.

For example, if each HP configuration is evaluated based on its apparent error (i.e. for each $\boldsymbol{\lambda}^{(c)}$, a model is trained and evaluated on $\mathcal{D}_{\text{train}}$, which also serves as $\mathcal{D}_{\text{test}}$), the tuning procedure will, due to the optimistically biased prediction error estimation, typically select the HP configuration that results in the model with the highest degree of overfitting. Although this approach should clearly be avoided, it might still be common practice in manual tuning as it is time-efficient (only one model per HP configuration needs to be trained, which in this case also corresponds to the final model) and may seem intuitive to inexperienced users.

Due to the optimistic bias of the apparent error, the standard approach for automated HP tuning is to employ a resampling method. In the case of k-fold CV, which is a common choice for HP tuning (Bischl et al., 2023), this means that for each candidate HP configuration $\boldsymbol{\lambda}^{(c)}$, k models are trained and evaluated on different subsets of $\mathcal{D}_{\text{train}}$.

While resampling methods provide an improvement over using the apparent error, the corresponding estimators also exhibit a certain degree of pessimistic bias and variance (with the degree of bias and variance depending on the resampling method used, as discussed in Section 3.2.1). A potential pitfall arising from the variance is that the winning HP configuration, $\boldsymbol{\lambda}^{\text{II}}$, may have been selected simply because the trained prediction model(s) using $\boldsymbol{\lambda}^{\text{II}}$ performed particularly well by chance on the specified test data set(s) $\mathcal{D}_{\text{test}}$, which are the same for each evaluated HP configuration. This means that the HP selection has essentially been overfitted to the respective test data set(s) $\mathcal{D}_{\text{test}}$, which in this context is also referred to as overtuning, overhyping, or oversearching (Bischl et al., 2023; Cawley & Talbot, 2010; Feurer & Hutter, 2019; Hosseini et al., 2020; Ng, 1997; Quinlan & Cameron-Jones, 1995). If the true prediction error of $\boldsymbol{\lambda}^{\text{II}}$ is still comparable to the prediction error of $\tilde{\boldsymbol{\lambda}}^{*}$, overtuning effects are negligible. However, there might also be scenarios in which the *true* prediction error of $\boldsymbol{\lambda}^{\text{II}}$ is no better, or even worse, than that of the default HP configuration, but its *estimated* prediction error is drastically deflated (i.e. over-optimistic), as the corresponding prediction model(s) that were trained during resampling incidentally fit very well to the specific noise pattern in the respective test data set(s) $\mathcal{D}_{\text{test}}$. This has been demonstrated in several experiments where tuning was conducted on null data (i.e. data without any true signal), yet the prediction error estimate of the selected HP configuration $\boldsymbol{\lambda}^{\text{II}}$ was substantially smaller (i.e. better) than its true prediction error indicating random prediction (Bischl et al., 2023; Boulesteix & Strobl, 2009; Hosseini et al., 2020; Varma & Simon, 2006).

Note that since the HPs are overfitted to the test data set(s) $\mathcal{D}_{\text{test}}$, which are not seen during training on the corresponding $\mathcal{D}'_{\text{train}}$, overtuning occurs on a higher level than overfitting of the model parameters (see Section 3.1). Accordingly, overtuning effects may only be visible after evaluating a large number of HP configurations (Bischl et al., 2023). However, literature suggests that the risk of overtuning does not only depend on the number of evaluated HP configurations but also, for example, on the search strategy, the type of tuned HP, and the number of observations in $\mathcal{D}_{\text{train}}$ (Cawley & Talbot, 2010; Hosseini et al., 2020; Wainer & Cawley, 2021).

In general, overtuning is considered an open problem of HP tuning, and although strategies have been suggested to avoid it (e.g., using different splits for each evaluation, Nagler et al., 2024), there are no commonly agreed-upon solutions (Feurer & Hutter, 2019).
Importantly, when overtuning is addressed in the literature, it is typically assumed that the prediction error estimation is performed through resampling methods. However, as discussed above, this estimation can alternatively be based on the apparent error. In cases where an inadequate HP configuration is selected due to the use of the apparent error for prediction error estimation, this can be considered a more extreme and direct form of overtuning since the test data set(s) $\mathcal{D}_{\text{test}}$ are seen during model training. We will refer to the two types of overtuning as resampling-induced and apparent error-induced overtuning.

4.2 Model evaluation

As outlined in Section 4.1.1, the model generation process in Setting II results in a final prediction model $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}}^{\mathcal{D}_{\text{train}}}$. Evaluating this model is generally more complex than evaluating a prediction model with pre-specified HPs (Setting I), since it must be taken into account that the model generation process involved HP tuning. Similar to Section 3.2, we will in the following differentiate between cases in which the model generation (i.e. the HP tuning followed by a final training) is performed on the full data set (i.e. $\mathcal{D}_{\text{train}} = \mathcal{D}$) vs. a (proper) subset of the available data (i.e. $\mathcal{D}_{\text{train}} \subset \mathcal{D}$). A graphical overview of model evaluation in Setting II is provided in Figure 3.

4.2.1 Evaluation of a model generated on all available data

Apparent error As in Setting I, reporting the apparent error for model evaluation is inappropriate in Setting II (see Figure 3, model evaluation a). In this case, however, the designated test data set $\mathcal{D}_{\text{test}} = \mathcal{D}_{\text{train}} = \mathcal{D}$ is even used twice during model generation: first during the HP tuning process and then again during the final training process. Depending on the specific tuning procedure employed, this can introduce an even greater optimistic bias compared to, for example, using default HP values. Although the apparent error is generally not suitable for assessing a model's performance, some users who performed tuning via resampling may mistakenly believe it now reflects a form of resampling error. This was noted by Neunhoeffer and Sternberg (2019), who also reference a paper that appears to have fallen into this pitfall.

Resampling error Similar to Setting I, an alternative evaluation procedure in Setting II is to employ a resampling method (see Figure 3, model evaluation b). In principle, the chosen resampling method is carried out as described in Section 3.2.1, except that in each resampling iteration, the model is trained on $\mathcal{D}'_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{test}}$ with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{II}}$ instead of $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\text{I}}$. Unfortunately, unlike in Setting I, using resampling methods for model evaluation in Setting II results in data leakage: Although in each resampling iteration, $\mathcal{D}_{\text{test}}$ is not involved in training $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}}^{\mathcal{D}'_{\text{train}}}$ (the model trained on $\mathcal{D}'_{\text{train}}$ for evaluation purposes), it is used in the tuning process performed on $\mathcal{D}_{\text{train}}$ (including $\mathcal{D}_{\text{test}}$) to obtain $\boldsymbol{\lambda}^{\text{II}}$. Accordingly, since not every model generation

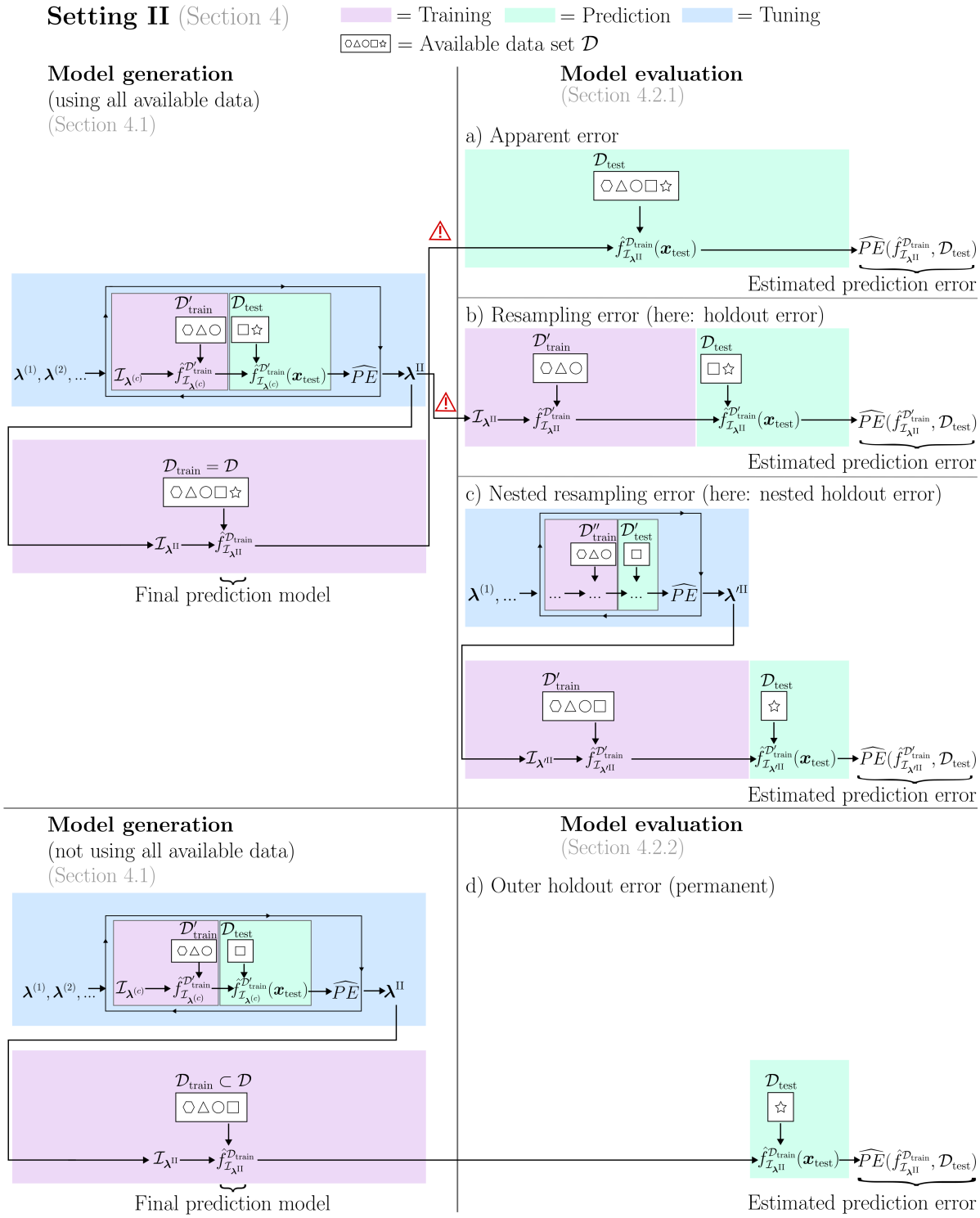

Figure 3: Overview of different model evaluation procedures and their relation to the model generation process if tuning is based on (temporary) holdout and all HPs are tuned. Data leakage is present if any subset of $\mathcal{D}_{\text{test}}$ used for prediction error estimation has also been employed to generate the evaluated prediction model (which is not necessarily the final model). In the figure, the point at which data "leaks" into the model evaluation is marked by the red caution symbol.

step resulting in $\hat{f}^{\mathcal{D}'_{\text{train}}}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}}$ is conducted exclusively on $\mathcal{D}'_{\text{train}}$, information from $\mathcal{D}_{\text{test}}$ is available during the model generation process (specifically, during tuning). Based on the definition given in Section 2.4.2, this constitutes a form of data leakage and may result in an optimistically biased resampling error (Hosseini et al., 2020; Wainer & Cawley, 2021). While the inadequacy of the apparent error is widely recognized, the described pitfall associated with the resampling error is less well known and will go undetected by those not involved in model development if HP tuning is not reported (Hosseini et al., 2020; Lones, 2024).

The potential optimistic bias becomes evident when considering the following typical practice: As outlined in Section 4.1.1, the tuning process already returns a prediction error estimate for the final prediction model (the estimate based on which $\boldsymbol{\lambda}^{\text{II}}$ was selected). Given that tuning was performed with a resampling method (e.g., CV), computation time can be saved by directly using this value as the resampling-based evaluation result. However, if the selected HP configuration $\boldsymbol{\lambda}^{\text{II}}$ is the result of overtuning, this will not be detected in the model evaluation process, as the deflated prediction error estimate is simply adopted here. In principle, adopting the resampling prediction error estimate from tuning in Setting II behaves analogously to (resampling-induced) overtuning as using the apparent error does to overfitting in Setting I. This is because both procedures are unable to discern that either the selected HPs (overtuning) or the selected parameters (overfitting) have been adapted too much to the respective test data set(s) $\mathcal{D}_{\text{test}}$.

As stated in Section 4.1.3, the extent to which overtuning occurs depends on the specific tuning procedure. If the HP selection is mildly overtuned, the prediction error estimate obtained from the tuning process may only exhibit a slight optimistic bias. However, as an extreme case, we can again consider the experiments from Section 4.1.3 in which HP tuning has been performed on null data (Bischl et al., 2023; Boulesteix & Strobl, 2009; Hosseini et al., 2020; Varma & Simon, 2006). Here, the difference between the prediction error estimate of the selected HP configuration and the true prediction error indicating random prediction is substantial, and adopting the former as the final evaluation result for a useless prediction model is clearly a biased approach.

Note that data leakage is also present if the specified $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ subsets used for tuning and evaluation are not identical. This is the case if additional resampling iterations are conducted during evaluation, if different resampling methods are used during tuning and evaluation (e.g., holdout and k-fold CV), or if the apparent error is used for tuning.

Nested resampling error The optimistic bias of the resampling error arises because, in each resampling iteration, not all steps of the model generation process are performed exclusively on $\mathcal{D}'_{\text{train}}$. A natural extension, therefore, is to ensure that the complete model generation is applied only to $\mathcal{D}'_{\text{train}}$ in every iteration (see Figure 3, model evaluation c). Specifically, this implies that the tuning process is not only performed once on $\mathcal{D}_{\text{train}}$ in order to generate the final prediction model but also on every $\mathcal{D}'_{\text{train}}$ specified during resampling (for evaluation

purposes). If the tuning process itself is based on a resampling method (i.e. if tuning is not performed using the apparent error, which is hardly ever the case if the currently described model evaluation procedure is employed), this results in two nested resampling methods. Accordingly, this procedure is called nested resampling, where the resampling method that initially splits $\mathcal{D}_{\text{train}}$ into $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ is the outer resampling loop and the resampling method creating additional splits within each $\mathcal{D}'_{\text{train}}$ (resulting in subsets denoted as $\mathcal{D}''_{\text{train}}$ and $\mathcal{D}'_{\text{test}}$) is the inner resampling loop (e.g., Bischl et al., 2023; Hosseini et al., 2020; Wainer & Cawley, 2021). To distinguish nested resampling from the resampling methods discussed above and in Section 3.2.1, we will refer to the latter as simple resampling where necessary.

The most straightforward form of nested resampling is the nested holdout method, where $\mathcal{D}_{\text{train}}$ is split once into $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, and $\mathcal{D}'_{\text{train}}$ is further divided into $\mathcal{D}''_{\text{train}}$ and $\mathcal{D}'_{\text{test}}$. In this setup, the best HP configuration for $\mathcal{D}'_{\text{train}}$ is determined by training and evaluating a model for each candidate HP configuration on $\mathcal{D}''_{\text{train}}$ (for training) and $\mathcal{D}'_{\text{test}}$ (for prediction error estimation). We denote this configuration as $\boldsymbol{\lambda}'^{\text{II}}$, as it may differ from the final prediction model's configuration, $\boldsymbol{\lambda}^{\text{II}}$, which has been obtained by tuning the model on $\mathcal{D}_{\text{train}}$ rather than $\mathcal{D}'_{\text{train}}$. Using the HP configuration $\boldsymbol{\lambda}'^{\text{II}}$, the model is then trained on $\mathcal{D}'_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{test}}$, which has remained unseen throughout the entire model generation process. Note that nested holdout is commonly referred to as train-validation-test split (Bischl et al., 2023), which, using the notation above, could also be referred to as $\mathcal{D}''_{\text{train}}$-$\mathcal{D}'_{\text{test}}$-$\mathcal{D}_{\text{test}}$-split. Instead of holdout, any other resampling method can be used for inner and outer resampling, and it is also possible to combine different resampling methods. For example, k-fold CV can be used for outer resampling and holdout for inner resampling, since in the inner resampling, precise prediction error estimation is less critical as long as a sufficiently good $\boldsymbol{\lambda}'^{\text{II}}$ is selected in each iteration (Bischl et al., 2023; Hosseini et al., 2020).

While nested resampling prevents data leakage, it also has several disadvantages. First, it can be very computationally expensive, since the tuning process, which can already be time-consuming when conducted once, has to be repeated for each $\mathcal{D}'_{\text{train}}$ specified by the outer resampling loop (Bischl et al., 2023; Wainer & Cawley, 2021). Second, it is usually not feasible to conduct nested resampling with manual tuning. Apart from being even more time-demanding than nested resampling with automated tuning, it is often not possible to repeat the same tuning procedure more than once due to the informal nature of manual tuning (e.g., the user might not remember which candidate HP configurations have been evaluated during tuning). Third, like simple resampling, nested resampling does not provide an estimate of the prediction error for the final model $\hat{f}^{\mathcal{D}_{\text{train}}}_{\mathcal{I}_{\boldsymbol{\lambda}^{\text{II}}}}$. However, while both methods evaluate models trained on $\mathcal{D}'_{\text{train}}$ rather than $\mathcal{D}_{\text{train}}$ (with $n'_{\text{train}} < n_{\text{train}}$), simple resampling at least uses the same HP configuration $\boldsymbol{\lambda}^{\text{II}}$ as the final prediction model. In contrast, nested resampling does not necessarily evaluate models with the same HP configuration, as each inner resampling loop may select a different configuration (see the nested holdout example above, which evaluates a model based on $\boldsymbol{\lambda}'^{\text{II}}$ instead of $\boldsymbol{\lambda}^{\text{II}}$). This makes the nested resampling result more difficult to interpret (Hosseini

et al., 2020). The described disadvantages could explain why nested resampling estimates are not commonly reported in studies presenting new prediction models, as indicated by a recent systematic review on clinical prediction models (Andaur Navarro et al., 2023).

4.2.2 Evaluation of a model generated on a subset of the available data

As in Setting I (see Section 3.2.2), it is also possible in Setting II to use only a subset of the available data for model generation (i.e. $\mathcal{D}_{\text{train}} \subset \mathcal{D}$) and reserve the remaining observations exclusively for evaluation (i.e. $\mathcal{D}_{\text{test}} = \mathcal{D} \setminus \mathcal{D}_{\text{train}}$; see Figure 3, model evaluation d; Hosseini et al., 2020). This approach essentially corresponds to nested resampling with holdout as the outer resampling method, except that the holdout is permanent, meaning that the prediction model generated on $\mathcal{D}_{\text{train}}$ (equivalent to $\mathcal{D}'_{\text{train}}$ in the previous section) serves as the final prediction model. Similar to Setting I, we thus distinguish the two evaluation procedures by referring to them as temporary outer holdout (described in Section 4.2.1) and permanent outer holdout (described here). We also again note that there might be some confusion in the terminology, as a permanent outer holdout combined with a (temporary) inner holdout can, just like its temporary counterpart, also be referred to as a train-validation-test split.

The statements regarding the temporary vs. permanent holdout in Setting I also apply to Setting II: Compared to the temporary outer holdout, the permanent outer holdout does not exhibit a pessimistic bias as it actually evaluates the final prediction model. However, this comes at the cost of not using all available data for model generation. Accordingly, the same recommendation as in Section 3.2.2 applies: a permanent outer holdout should only be employed if the number of observations in $\mathcal{D}$ is sufficiently large or if it is computationally expensive or practically infeasible to repeat the model generation process. Note that the second point is particularly relevant in Setting II due to the increased effort of model generation (Collins, Dhiman, et al., 2024).

5 Empirical illustration of different model generation and evaluation procedures

In this section, we illustrate different procedures for model generation and evaluation and assess their impact on prediction error estimates from available vs. new data. We specifically focus on the selection of HPs and the potential for data leakage.

5.1 Real-world prediction problem

Our illustration is based on a real-world prediction problem from the COMPANION study (Hodiamont et al., 2022). This study aimed to develop a casemix classification for adult palliative care patients in Germany that considers the complexity of each patient's palliative care situation to assign them to a class reflecting their resource needs. A casemix classification for palliative care patients has been deemed necessary, as the differentiation of patients based on their diagnosis, which corresponds to the current practice in Germany, has been found to be inappropriate for predicting resource needs in the context of palliative care. Despite yielding

many important insights, the COMPANION project was ultimately unable to develop a prediction model with sufficient predictive performance, even after exploring various model generation approaches. However, this makes it a good example to illustrate how optimistically biased evaluation procedures can present prediction models in a more favorable light.

To develop a casemix classification that relates patients' resource needs to the complexity of their palliative care situation, the COMPANION team formulated a prediction problem where each observation represents a patient's palliative care phase. The outcome $y^{(i)}$, defined as the average cost per day in palliative care phase i, serves as an empirical proxy for resource needs in the corresponding phase. The set of features $\boldsymbol{x}^{(i)}$ intended to reflect the palliative care situation of each phase consists of (i) the type of palliative care phase (categorical), (ii) patient age (integer-valued), (iii) two cognitive features (confusion and agitation; both ordinal), (iv) the Australia-modified Karnofsky Performance Status (AKPS; Abernethy et al., 2005) that measures the patients' functional status (ordinal), and (v) the Integrated Palliative care Outcome Scale (IPOS; Murtagh et al., 2019), which is a score that is based on 17 ordinal variables covering physical symptoms, psycho-social burden, family needs, and practical problems. Accordingly, the number of features provided to the learning algorithm is $p = 6$. All types of data were collected by the clinical staff of participating palliative care teams.

It is important to note that although the study aimed to identify a casemix classification, the continuous nature of the specified outcome variable (i.e. average cost per day) inherently makes the prediction problem a regression task. To ensure that the obtained prediction model still produces classes that are also interpretable and can be implemented in practice, a decision tree approach was chosen (e.g., using the CART algorithm, discussed in Sections 2-4), despite potential limitations on predictive performance. In the resulting decision tree, each terminal node represents a casemix class (defined by the features that capture the complexity of the palliative care situation) and predicts the average cost per day for that class. Notably, decision trees were also used in the casemix classifications developed for palliative care patients in Australia (Eagar et al., 2004) and the UK (Murtagh et al., 2023), which served as the basis for many decisions in the development of the German casemix classification.

The COMPANION study collected data from three palliative care settings (specialist palliative care units, palliative care advisory teams, and specialist palliative home care), with a casemix classification to be developed for each setting. In our illustration, we only consider the data from the specialist palliative home care setting. We apply several parameterless preprocessing steps to the raw data set, which correspond to those used in the COMPANION study and are considered as pre-specified in our illustration (e.g., the removal of dead patients; more details can be found in Supplementary Section B.2.1). The resulting data set contains 1,449 palliative care phases; descriptive statistics are provided in Table S1.

Note that while our experimental setup described in the following section is based on the COMPANION study, not all aspects align with how the actual study was conducted, as some elements have been simplified or modified for illustrative purposes.

5.2 Experimental setup

5.2.1 Overview

The aim of our study is to illustrate different model generation and evaluation procedures and examine their impact on prediction error estimates derived from available data compared to those obtained from new data. Additionally, we examine how these estimates are affected by performance measure, sample size, and learning algorithm, resulting in a total of 96 distinct analysis settings. Before providing more details on these, we first outline the general procedure that is carried out for each analysis setting:

(i) The COMPANION data set with 1,449 observations (i.e. palliative care phases) introduced above is randomly split into two subsets of equal size, which we denote as $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{new}}$ (with $n_{\text{train}} = 724$ and $n_{\text{new}} = 725$). We assume that $\mathcal{D}_{\text{train}}$ is the only data set available for both model generation and evaluation. Consistent with the notation used in previous sections, this implies $\mathcal{D}_{\text{train}} = \mathcal{D}$. The desired output is a prediction model as described above (i.e. a decision tree that predicts the average patient costs based on several features reflecting the palliative care situation).

(ii) We use $\mathcal{D}_{\text{train}}$ exclusively to generate and evaluate a prediction model. Although the specific procedure is determined by the analysis setting, each model is generated using all available data (which is already implied by referring to the available data as $\mathcal{D}_{\text{train}}$). The learning pipeline used for each training process and its HPs are described in Section 5.2.2. Since the HP selection in the considered analysis settings can be either data-independent or achieved through tuning, we refer to the chosen HP configuration as $\boldsymbol{\lambda}$ rather than $\boldsymbol{\lambda}^{\text{I}}$ or $\boldsymbol{\lambda}^{\text{II}}$ in the following to keep the notation general. Step (ii) results in a model $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}$ and an associated prediction error estimate, which we denote as $\widehat{\text{PE}}_{\text{train}}$. In an ML application, $\widehat{\text{PE}}_{\text{train}}$ would be the reported error.

(iii) The prediction model $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}$ is evaluated on the second data set $\mathcal{D}_{\text{new}}$, which represents observations that are drawn from the same distribution as the observations in $\mathcal{D}_{\text{train}}$ but were unseen during the generation of $\hat{f}_{\mathcal{I}_{\boldsymbol{\lambda}}}^{\mathcal{D}_{\text{train}}}$. This step should therefore yield an unbiased estimate of the model's prediction error, denoted as $\widehat{\text{PE}}_{\text{new}}$ (however, see the note on clustering in Section 5.3 and Supplementary Section B.5). Note that, in principle, the estimation of $\widehat{\text{PE}}_{\text{new}}$ resembles a permanent holdout approach, where $\mathcal{D}_{\text{new}}$ is held out during model generation. However, it is not truly a holdout, as $\mathcal{D}_{\text{new}}$ is unavailable during model evaluation. This is also why $\mathcal{D}_{\text{new}}$ is not referred to as $\mathcal{D}_{\text{test}}$; throughout the paper, the notation $\mathcal{D}_{\text{test}}$ is used exclusively for subsets of the available data.

Performing steps (i) to (iii) results in a vector $(\widehat{\text{PE}}_{\text{train}}, \widehat{\text{PE}}_{\text{new}})$, which includes the prediction error estimates derived from available and new data, respectively. By comparing these estimates, we can determine whether $\widehat{\text{PE}}_{\text{train}}$ correctly reflects the predictive performance of the model or if it is affected by any form of bias. Ideally, $\widehat{\text{PE}}_{\text{train}}$ should be equal to $\widehat{\text{PE}}_{\text{new}}$, indicating that

the model evaluation conducted on $\mathcal{D}_{\text{train}}$ yields an unbiased estimate prediction error estimate (although small differences do not necessarily indicate bias, as $\widehat{\text{PE}}_{\text{new}}$ is also an estimate). To ensure that the difference between the two prediction error estimates is not driven by a specific data split, steps (i) to (iii) are repeated 50 times for each analysis setting (using the same 50 splits for each analysis setting). Since we consider 96 analysis settings and 50 repetitions of splitting the initial COMPANION data set, our illustration generates $96 \times 50 = 4{,}800$ vectors of $(\widehat{\text{PE}}_{\text{train}}, \widehat{\text{PE}}_{\text{new}})$. Note that each analysis setting may produce 50 different prediction models, as in each repetition, $\mathcal{D}_{\text{train}}$ contains different observations.

The described setup is implemented in the software environment `R` (R Core Team, 2022) using the `mlr3` package framework (Lang et al., 2019). While the COMPANION data set cannot be made publicly available, the `R` code and the individual prediction error estimates can be found at https://github.com/NiesslC/overoptimistic_trees.

As stated above, we consider a total of 96 analysis settings. These result from a full factorial variation of four factors: two performance measures, two sample sizes, two learning algorithms, and twelve combinations of model generation and evaluation procedures (yielding the total of $2 \times 2 \times 2 \times 12 = 96$ analysis settings). The two considered sample sizes are (i) $n_{\text{train}} = 724$ (the sample size of $\mathcal{D}_{\text{train}}$ after splitting the original data set) and (ii) $n_{\text{train}} = 362$ (half of the observations in $\mathcal{D}_{\text{train}}$ being randomly deleted). Note that $\mathcal{D}_{\text{new}}$ is not affected by this variation and still has $n_{\text{new}} = 725$ observations. The two performance measures considered in our illustration are the Root Mean Squared Error (RMSE) and the coefficient of determination (R^2), which are commonly used performance measures and have also been employed to evaluate other decision-tree-based prediction models for palliative care patients (Eagar et al., 2004; Murtagh et al., 2023; see Supplementary Section B.3 for more information on both performance measures). Note that in each analysis setting, we use the same performance measure for both the model evaluations performed during model generation (i.e. tuning) and the evaluation of the final prediction model. The two learning algorithms and twelve combinations of model generation and evaluation procedures are described in Sections 5.2.2 and 5.2.3, respectively.

5.2.2 Learning pipeline and HPs

The learning pipeline $\mathcal{I}$ applied in each training process consists of six preprocessing steps, followed by a learning algorithm (see Figure 4 for an overview). While the full learning pipeline actually consists of more preprocessing steps (referred to in Section 5.1 and detailed in Supplementary Section B.2.1), we will, for simplicity, not further consider them in the illustration, as they are considered as pre-specified (i.e. have no HPs that are relevant for tuning) and are both parameterless and precede the first parameterized preprocessing step in the learning pipeline (i.e. can safely be applied to the full data set).

Preprocessing steps Here, we provide a brief overview of the six preprocessing steps in $\mathcal{I}$ applied during each training process and outline their associated HPs. Additional details can be found in Figure 4, and a comprehensive description is available in Supplementary Section B.2.2.

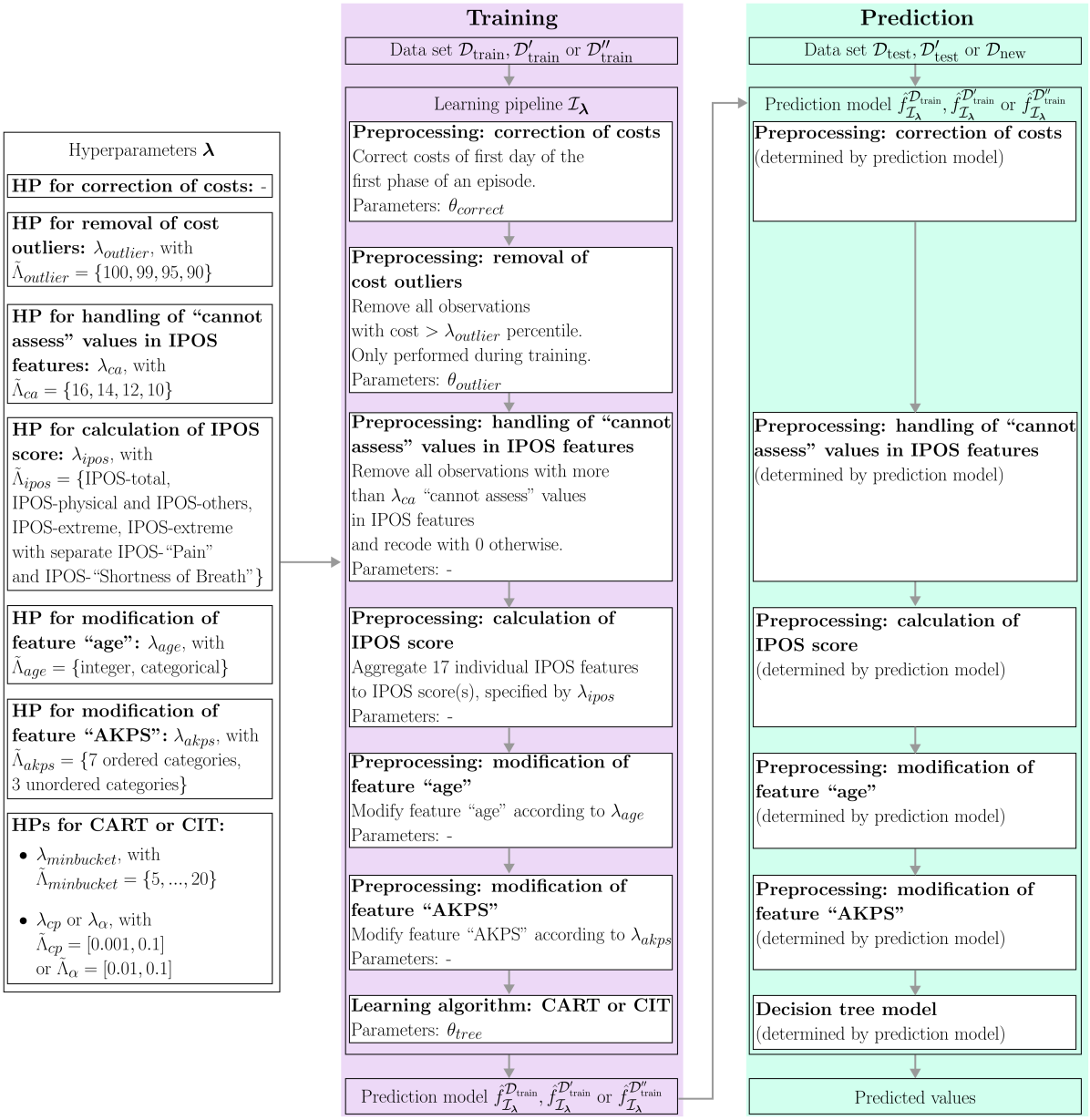

Figure 4: Overview of the learning pipeline $\mathcal{I}$ used in the illustration (middle panel). In addition, the considered HPs, their search spaces (left panel), and the steps applied during prediction (right panel) are shown.

The six preprocessing steps serve one of three purposes: (i) correction of the outcome variable (correction of costs), (ii) handling of problematic observations (removal of cost outliers and handling of "cannot assess" values in IPOS features), and (iii) calculation or modification of features (calculation of the IPOS score, modification of the feature "age", and modification of the feature "AKPS"). As discussed in Section 2.2.2, preprocessing steps can be distinguished based on different characteristics, which also applies to the six preprocessing steps considered in this section. Two of the six steps have parameters: the correction of costs (with $\theta_{correct}$) and the removal of cost outliers (with $\theta_{outlier}$). These two steps, along with another step (handling

of "cannot assess" values in IPOS features), alter the outcome distribution, but the removal of cost outliers is not applied during prediction.
All preprocessing steps, except for the correction of costs, include HPs: $\lambda_{outlier}$, λ_{ca}, λ_{ipos}, λ_{age}, and λ_{akps}. Consistent with the notation introduced in Section 2.3.1, we collectively refer to them as $\boldsymbol{\lambda}_P$. For these HPs, it is not possible to define a HP domain Λ_j that contains all possible configurations; therefore, we only specify a search space $\tilde{\Lambda}_j$ for each HP (see Figure 4). Each search space is categorical, offering 2 or 4 values, all of which have been discussed and deemed reasonable during the COMPANION project. The first HP value in each search space is set as the default and corresponds to the value ultimately selected for the COMPANION project.

Learning algorithm After applying all preprocessing steps to the data, it is provided to the learning algorithm, which then yields a prediction model (i.e. a decision tree). We consider two learning algorithms: (i) the CART algorithm (introduced in Section 2.2.1; `R` package `rpart`; Therneau and Atkinson, 2022), and (ii) the Conditional Inference Tree algorithm (CIT; `R` package `partykit`; Hothorn and Zeileis, 2015; Hothorn et al., 2006; Zeileis et al., 2008). As stated in Sections 2.2.1 and 3.1, the CART algorithm builds a decision tree model by partitioning the feature space $\mathcal{X}$ into terminal nodes using a sequence of binary splits. Since we are considering a regression problem, the splitting rules are determined by minimizing the sum of squared errors, and the prediction value $\hat{f}(\boldsymbol{x})$ for each terminal node is the mean of all outcome values (here: costs) in that node (Breiman et al., 1984). The CIT algorithm also employs recursive binary partitioning, but instead of minimizing a simple loss function that represents node impurity (here: the sum of squared errors), it uses statistical test procedures to find the optimal splits. This approach has the advantage that, unlike the CART algorithm, the CIT algorithm is not affected by selection bias toward features with many possible splits or missing values (Hothorn et al., 2006).
For both algorithms, we consider two HPs for tuning that determine when the algorithm stops splitting. The first HP is $\lambda_{minbucket}$, which specifies the minimum number of observations in any terminal node. The smaller $\lambda_{minbucket}$, the larger the number of terminal nodes in the resulting decision tree and the higher the risk of overfitting. We set the search space of $\lambda_{minbucket}$ to $\{5, \ldots, 20\}$ for tuning. If $\lambda_{minbucket}$ is not tuned, we set the HP to its default, $\lambda_{minbucket} = 7$. The second HP is either λ_{cp} (for CART) or λ_{α} (for CIT). Both HPs serve a similar purpose: λ_{cp} determines the factor by which a split must improve the overall lack of fit to be attempted (which, in case of a regression problem, corresponds to improving the overall R^2 of the model by at least λ_{cp}). The HP λ_{α} is the numerical significance level that must be met in the statistical testing procedure conducted by CIT to implement a split. Accordingly, the smaller λ_{cp} or the higher λ_{α}, the higher the risk of overfitting. We specify the search space for λ_{cp} and λ_{α} as $[0.001, 0.1]$ and $[0.01, 0.1]$, respectively. If λ_{cp} and λ_{α} are not tuned, we use their default values of $\lambda_{cp} = 0.01$ and $\lambda_{\alpha} = 0.05$.
All other HPs of CART and CIT are not tuned and, except for one HP, follow the default values

from their corresponding implementation in the `mlr3` package (Lang et al., 2019), which largely align with the defaults of the underlying packages (i.e. `rpart` and `partykit`; Foss and Kotthoff, 2024). The exception is $\lambda_{maxdepth}$, which we set to 4 to align with the COMPANION project, where this value was chosen to ensure that the resulting decision tree model would be useful in clinical practice.

We refer to the algorithm HPs that are considered for tuning (i.e. $\lambda_{minbucket}$ and λ_{cp} or λ_{α}) as $\boldsymbol{\lambda}_A$. The remaining algorithm HPs that are not tuned in any of the analysis settings will not be considered further for simplicity.

5.2.3 Model generation and evaluation procedures

We consider twelve different combinations of model generation and evaluation procedures that could be employed in step (ii) of our illustration (see Section 5.2.1) to obtain a prediction model with associated $\widehat{\text{PE}}_{\text{train}}$. They represent an exemplary yet non-exhaustive selection of procedures that are used in ML applications. The twelve combinations are based on five model generation procedures, where for three of them, we apply two different procedures to evaluate the final prediction model, and for the other two, we use three different evaluation procedures (resulting in a total of $3 \times 2 + 2 \times 3 = 12$ combinations).

Before describing the procedures in more detail, there are a few general points to consider. First, as already stated in Section 5.2.1, all model generation procedures use the full data set $\mathcal{D}_{\text{train}}$ that was created by the respective repetition, i.e. we do not consider the permanent holdout evaluation procedures introduced in Sections 3.2.2 and 4.2.2 (which would imply $\mathcal{D}_{\text{train}} \subset \mathcal{D}$). Second, since the prediction model used in this illustration is a decision tree, it is theoretically possible to manually assess the plausibility of the generated models in addition to estimating their prediction error. However, in addition to not being feasible for all 96×50 generated models, this step is also often not part of the evaluation process in practice, as many ML-based prediction models are not interpretable by humans without additional tools. Therefore, we do not perform this assessment. Third, whenever $\mathcal{D}_{\text{train}}$ is (temporarily) split as part of a resampling method (either during model generation or evaluation), we use the same splits (e.g., the same 10 CV folds) across all procedures to ensure that differences in prediction error estimates are not due to variations in the data splits of $\mathcal{D}_{\text{train}}$.

We now present the procedures in more detail, first describing the model generation procedure and then the associated evaluation procedures to estimate the prediction error of the resulting model. The following paragraph titles refer to the model generation procedures and can be read as "Setting - Tuning Procedure (- HPs tuned)". An overview of all generation and evaluation procedures is provided in Table 1.

I-no tuning The simplest model generation procedure corresponds to Setting I, where all HPs are set to their default values (i.e. no tuning is performed), and the learning pipeline only needs to be trained once on the data set $\mathcal{D}_{\text{train}}$.

For this model generation procedure, we evaluate the resulting model by (i) the apparent error

Table 1: Overview of the twelve combinations of model generation and evaluation procedures examined in the illustration. They result from five model generation procedures, each paired with two or three evaluation procedures.

Setting	**Model generation on $\mathcal{D}_{\text{train}}$**								**Model evaluation on $\mathcal{D}_{\text{train}}$**	
				Tuning procedure						
	Model generation name	**Pre-specified HPs**	**Tuned HPs**	**Search space**	**Termination criterion**	**Search strategy**	**Joint vs. sequential tuning**	**Prediction error estimation**	**Prediction error estimation**	**Data leakage possible**
I	I-no tuning	$\boldsymbol{\lambda}_P$, $\boldsymbol{\lambda}_A$	-	-	-	-	-	-	Apparent	Yes
									10-fold CV	No
II	II-manual-P	$\boldsymbol{\lambda}_A$	$\boldsymbol{\lambda}_P$	See Figure 4	None	Exhaustive search	Sequential	Apparent	Apparent	Yes
									10-fold CV	Yes
II	II-automated-A	$\boldsymbol{\lambda}_P$	$\boldsymbol{\lambda}_A$	See Figure 4	60 evaluations	Random search	Joint	10-fold CV	Apparent	Yes
									10-fold CV	Yes
									10-2-fold nested CV	No
II	II-combined-PA	-	$\boldsymbol{\lambda}_P$, $\boldsymbol{\lambda}_A$	II-manual-P for $\boldsymbol{\lambda}_P$ and II-automated-A for $\boldsymbol{\lambda}_A$ (for each configuration of $\boldsymbol{\lambda}_P$)					Apparent	Yes
									10-fold CV	Yes
II	II-automated-PA	-	$\boldsymbol{\lambda}_P$, $\boldsymbol{\lambda}_A$	See Figure 4	210 evaluations	Random search	Joint	10-fold CV	Apparent	Yes
									10-fold CV	Yes
									10-2-fold nested CV	No

and (ii) the 10-fold CV error. The former is affected by data leakage and may thus exhibit a substantial optimistic bias (see Section 3.2.1).

II-manual-P In this model generation procedure, the preprocessing HPs ($\boldsymbol{\lambda}_P$) are tuned, while the algorithm HPs ($\boldsymbol{\lambda}_A$) are set to their default values. It aims to represent inexperienced users who either lack the confidence or the programming skills to tune algorithm HPs but manually experiment with different preprocessing options, without realizing that this is a form of HP tuning. As discussed in Sections 4.1.2 and 4.1.3, manual tuning procedures typically differ from automated tuning procedures, which is reflected by the procedure II-manual-P. First, the HPs are tuned sequentially (i.e. each HP is tuned individually, with previously tuned HPs set to their selected values and subsequently tuned HPs set to their default values). Second, during the tuning of each HP, the apparent error is used to estimate the prediction error of each candidate HP configuration. The order in which the HPs are tuned sequentially is λ_{ipos}, λ_{age}, λ_{akps}, $\lambda_{outlier}$, λ_{ca} (which reflects a user who first experiments with variations in the features before removing observations, though any other order is also possible). If more than one HP value yields the same prediction error estimate, the first value that was evaluated is selected. Since the preprocessing HPs are tuned sequentially (i.e. one at a time), and only two (λ_{age}, λ_{akps}) or four (λ_{ipos}, $\lambda_{outlier}$, λ_{ca}) values per HP are available, only 16 ($= 2 \times 2 + 4 \times 3$) configurations of $\boldsymbol{\lambda}_P$ need to be evaluated during tuning. Therefore, no criterion is specified to terminate tuning before all configurations are evaluated.
Similar to the first model generation procedure (I-no tuning), we consider the apparent error and the 10-fold CV error to evaluate the final prediction model. However, the 10-fold CV error is now affected by data leakage, potentially leading to an optimistic bias due to (apparent error-induced) overtuning (see Section 4.2.1). Note that we do not consider evaluation procedures involving nested resampling for II-manual-P, as this is typically not feasible if manual tuning was used for model generation (see Section 4.2.1).

II-automated-A This model generation procedure represents a standard procedure in many ML applications, where the algorithm HPs $\boldsymbol{\lambda}_A$ are selected through automated tuning, while the preprocessing HPs $\boldsymbol{\lambda}_P$ are set to their default values (e.g., because users are not aware that they can be tuned). Even when tuning is fully automated, the procedures used in practice are often simple and based on rules of thumb (Bischl et al., 2023), which we aim to reflect in our illustration: we employ a random search algorithm, terminate the tuning after 60 evaluations (which corresponds to 30 times the dimension of the search space, as there are 2 HPs in $\boldsymbol{\lambda}_A$), and use 10-fold CV for prediction error estimation. The tuning procedure is performed jointly for all HPs, which is the standard practice for automated tuning.
As with the previous model generation procedures, we report both the apparent error and the 10-fold CV error. Note that, since the 10-fold CV error for the selected HP configuration, $\boldsymbol{\lambda}_A^{\text{II}}$, has already been calculated during tuning, we use this value as the 10-fold CV error estimate of the final prediction model to avoid performing additional resampling iterations. Similar to the

procedure II-manual-P, data leakage is present in both evaluation procedures and may result in optimistically biased prediction error estimates. Specifically, the optimistic bias in the 10-fold CV error would arise from (resampling-induced) overtuning. Since the procedure II-automated-A is fully automated, we additionally estimate the prediction error using nested CV. Here, we use 10 folds for the outer resampling loop and 2 folds for the inner resampling loop (the small number of inner folds saves computation time, and we only need to achieve correct HP selection rather than precise error estimation here; this is also recommended by Bischl et al., 2023). As discussed in Section 4.2.1, this evaluation procedure is not affected by data leakage.

II-combined-PA As a fourth model generation procedure, we tune both preprocessing and algorithm HPs (i.e. $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_A$), but with two different tuning procedures. More specifically, the preprocessing HPs are tuned as in II-manual-P, and for each candidate configuration of the preprocessing HPs, the algorithm HPs are tuned as in II-automated-A. Although this procedure might initially seem unintuitive and overly complex, it actually mirrors a realistic scenario for users who can tune algorithm HPs but may not be aware of or able to tune preprocessing HPs: Consider a user who has programmed three functions: (i) `preprocess_data`, which takes the raw data set as input and returns the preprocessed data set; (ii) `tune_algorithm`, which tunes the algorithm HPs as specified in II-automated-A based on the preprocessed data set and returns the selected HPs $\boldsymbol{\lambda}_A^{\text{II}}$; and (iii) `get_apparent_error`, which takes the preprocessed data set and a learning algorithm with HPs $\boldsymbol{\lambda}_A^{\text{II}}$ as input and returns the apparent error of the resulting model. Suppose the user initially plans to run these three functions once but is dissatisfied with the apparent error reported by `get_apparent_error`. They would then modify `preprocess_data` to try, for example, a different way of aggregating the IPOS score (i.e. using a different λ_{ipos}) and rerun `tune_algorithm` and `get_apparent_error`. After testing all values for λ_{ipos}, they would proceed to adjust λ_{age}, λ_{akps}, and so forth, updating the algorithm HPs by running `tune_algorithm` before calling `get_apparent_error` for each tried preprocessing configuration $\boldsymbol{\lambda}_P$. Note that since 16 configurations for $\boldsymbol{\lambda}_P$ are tried (see II-manual-P), and for each configuration of $\boldsymbol{\lambda}_P$, 60 candidate configurations for $\boldsymbol{\lambda}_A$ are evaluated (see II-automated-A), $60 \times 16 = 960$ HP configurations are assessed in total. The user would ultimately select the preprocessing HPs $\boldsymbol{\lambda}_P^{\text{II}}$ that yield the best apparent error and the algorithm HPs $\boldsymbol{\lambda}_A^{\text{II}}$ returned by `tune_algorithm` after setting $\boldsymbol{\lambda}_P^{\text{II}}$ in `preprocess_data`.
For this model generation procedure, we again consider the apparent error and the 10-fold CV error to evaluate the resulting prediction model. Note that the apparent error estimate corresponds to the best apparent error achieved during tuning and can therefore be directly adopted for evaluation. More specifically, it is the output of `get_apparent_error` after running `preprocess_data` with $\boldsymbol{\lambda}_P^{\text{II}}$ and then `tune_algorithm`. The 10-fold CV error estimate can also directly be taken from the tuning procedure and corresponds to the 10-fold CV estimate which was calculated during the execution of `tune_algorithm` after running `preprocess_data` with

$\boldsymbol{\lambda}_P^{\text{II}}$. For the reasons discussed in the previous model generation procedures, both the apparent error and the 10-fold CV error estimates are subject to data leakage.

II-automated-PA The final model generation procedure is similar to the procedure II-automated-A described above, except that the set of jointly tuned HPs now also includes the five preprocessing HPs, $\boldsymbol{\lambda}_P$, and the number of evaluations is increased to 210. As in II-automated-A, this corresponds to 30 times the dimension of the search space, as there are now 7 tuned HPs. This procedure represents a conceptually simple way to incorporate preprocessing HPs into the tuning process and is recommended by Bischl et al., 2023. However, as noted in Section 4.1.2, integrating preprocessing HPs into an automated tuning procedure requires advanced programming expertise, which may explain why this procedure is not standard practice yet.
We use the same three model evaluation procedures as in II-automated-A, with the same considerations discussed in II-automated-A also applying here.

5.3 Results

Figure 5 illustrates the differences between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$ for each of the 96 analysis settings (with 50 repetitions per setting). Additionally, the absolute values of $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$, as well as the selected HPs (for analysis settings where HPs are tuned), are presented in Figures S2 to S6.
Before examining the prediction error differences in more detail, we first consider the absolute values of $\widehat{\text{PE}}_{\text{new}}$ (displayed in Figure S2). Here, the general observation can be made that across all analysis settings, none of the generated models demonstrates sufficient predictive performance, which was expected and aligns with the findings of the COMPANION project. Of course, this result does not imply that HP tuning is generally not useful; rather, it demonstrates that tuning alone is not a guaranteed solution for obtaining a well-performing model for any prediction problem. Even in the analysis settings with the best median prediction errors (averaged across 50 repetitions), the median $\widehat{\text{PE}}_{\text{new}}$ reaches only 0.074 for R^2 ($n_{\text{train}} = 724$, CIT, II-manual-P) and 42.1 for RMSE ($n_{\text{train}} = 724$, CIT, II-automated-PA). For reference, the median $\widehat{\text{PE}}_{\text{new}}$ for RMSE using a naive model that predicts the mean of $\mathcal{D}_{\text{train}}$ on $\mathcal{D}_{\text{new}}$ is 44.0 for the smaller sample size and 43.5 for the larger sample size, which is only slightly worse than the result from the decision tree models. While small effects of sample size and learning algorithm on $\widehat{\text{PE}}_{\text{new}}$ can be observed (with larger sample sizes and using the CIT instead of the CART algorithm resulting in smaller prediction errors), no clear pattern emerges for the model generation procedure.
We will now analyze the differences between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$. To ensure consistent interpretation of their signs across both performance measures, the prediction error differences in Figure 5 are presented as $\widehat{\text{PE}}_{\text{new}} - \widehat{\text{PE}}_{\text{train}}$ for RMSE and $\widehat{\text{PE}}_{\text{train}} - \widehat{\text{PE}}_{\text{new}}$ for R^2. With this definition, a positive median difference indicates that the prediction error estimate $\widehat{\text{PE}}_{\text{train}}$ is optimistically biased, while a negative median difference suggests a pessimistic bias.
As stated in Section 5.2.3, depending on the model evaluation procedure, $\widehat{\text{PE}}_{\text{train}}$ corresponds

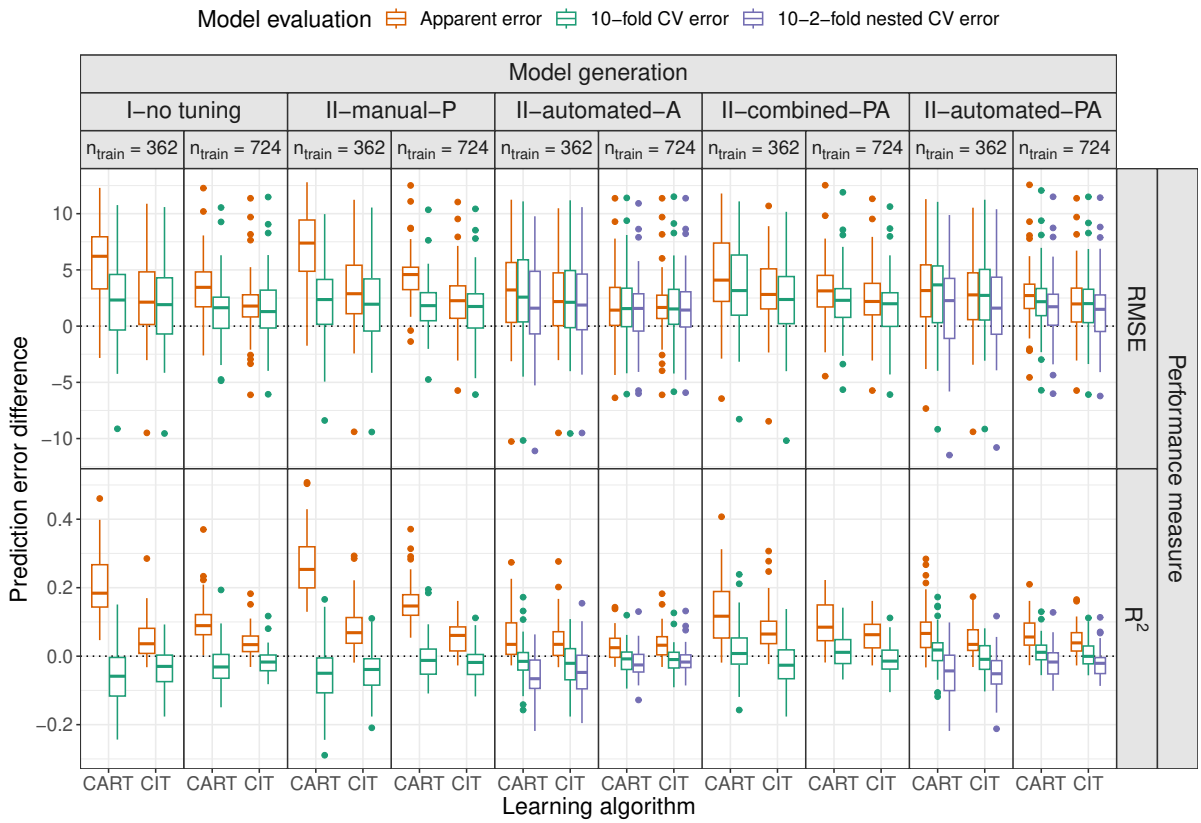

Figure 5: Resulting prediction error differences for 96 analysis settings, with each boxplot summarizing 50 repetitions of a specific setting. The prediction error differences are calculated as $\widehat{\text{PE}}_{\text{new}} - \widehat{\text{PE}}_{\text{train}}$ for RMSE and $\widehat{\text{PE}}_{\text{train}} - \widehat{\text{PE}}_{\text{new}}$ for R^2. For both performance measures, a positive median difference (averaged over the 50 repetitions) indicates that $\widehat{\text{PE}}_{\text{train}}$ is optimistically biased, while a negative median difference suggests a pessimistic bias.

to one of three prediction error estimates: (i) the apparent error, (ii) the 10-fold CV error, or (iii) the 2-fold-within-10-fold CV error. We structure the reporting of the results according to these three evaluation procedures.

Apparent error Figure 5 shows that, across the considered model generation procedures, the median prediction error differences vary the most for the apparent error. Despite this variation, the median differences are consistently positive in all analysis settings. Although there are individual repetitions with negative differences, these results clearly indicate that the apparent error is optimistically biased. As discussed in Section 3.2.1, this problem arises due to data leakage, or more specifically, the fact that this evaluation procedure uses observations for prediction error estimation that were already seen during model generation, which in turn allows potential overfitting and overtuning (if HPs are tuned) of the model to go undetected.
The optimistic bias of the apparent error is most pronounced in analysis settings where the preprocessing HPs $\boldsymbol{\lambda}_P$ are tuned manually (II-manual-P). This is not surprising, as this procedure specifically selects the HP values that optimize the apparent error. Here, the bias is

largest when the smaller sample size and the CART algorithm are used for model generation, resulting in a median difference of 7.39 for RMSE and 0.253 for R^2. Note that while the absolute values of $\widehat{\text{PE}}_{\text{train}}$ still do not indicate good predictive performance in these analysis settings (see Figure S2), the median R^2 values resulting from the CART algorithm (0.234 and 0.176 for the two sample sizes) are comparable to the prediction errors reported for the Australian and UK decision tree models (0.17 and 0.27), which were generally deemed viable (Eagar et al., 2004; Murtagh et al., 2023). Regarding the selected HPs, particularly for λ_{ipos} (which specifies how the IPOS score is calculated) and λ_{ca} (which determines how "cannot assess" values in IPOS features are handled), alternative values are frequently chosen instead of the defaults (see Figures S3a to S6a). This suggests that these alternative values may present a high potential for overfitting, thereby improving the apparent error.

In the analysis settings where both the preprocessing and the algorithm HPs are tuned using different procedures (II-combined-PA), the optimistic bias of the apparent error is similar for the CIT algorithm or slightly smaller for the CART algorithm compared to the II-manual-P procedure. Again, the optimistic bias is largest in the analysis settings where a smaller sample size and the CART algorithm are considered, resulting in a median difference of 4.09 for RMSE and 0.117 for R^2. The slight decrease in optimistic bias can be attributed to the fact that, across all analysis settings using the II-combined-PA procedure, the algorithm HP $\lambda_{minbucket}$ is set to a higher value than its default of $\lambda_{minbucket} = 7$, which results in a reduced risk of overfitting (see Figures S3b to S6b). In the analysis settings where no HPs are tuned (I-no tuning), the optimistic bias of the apparent error is also reduced slightly compared to the II-manual-P procedure. For the smaller sample size combined with the CART algorithm, the observed median difference is 6.21 for RMSE and 0.184 for R^2. The reduction in optimistic bias compared to II-manual-P is expected, as I-no tuning does not involve HP tuning.

The lowest optimistic bias for the apparent error is observed in the analysis settings where either only $\boldsymbol{\lambda}_A$ (II-automated-A) or both $\boldsymbol{\lambda}_P$ and $\boldsymbol{\lambda}_A$ (II-automated-PA) are tuned automatically, with the largest median difference being 3.22 for RMSE and 0.035 for R^2. This is not surprising, as in these procedures, all HPs are selected based on their associated CV error estimate rather than the apparent error. Notably, across all analysis settings, the HP values for $\boldsymbol{\lambda}_P$ selected by the II-automated-PA procedure differ from those chosen by the II-manual-P and II-combined-PA procedures (see Figures S3a to S6a).

CV error If $\widehat{\text{PE}}_{\text{train}}$ corresponds to the CV error, the resulting median prediction error differences indicate that this error is, as expected, generally less optimistic than the apparent error. The only exception occurs in a few analysis settings using RMSE as performance measure, where the apparent error differences are close to zero; here, the median differences of apparent error and CV error are approximately equal.

In the analysis settings without HP tuning, the R^2 differences exhibit a negative median difference, with the median difference closest to zero, -0.059, observed for the smaller sample size

combined with the CART algorithm. This pessimistic bias is an expected result, as CV evaluates models trained on fewer observations than the final prediction model (see Section 3.2.1). In contrast to R^2, the prediction error differences for RMSE in the analysis settings without tuning are mostly positive. Although the median differences are small (with the largest median difference being 2.32 in the analysis setting where both the smaller sample size and the CART algorithm are considered), the overall distribution of the prediction error differences in each setting suggests the presence of an optimistic bias. This finding is unexpected, as prediction errors estimated by CV in a setting where no HPs are tuned should not exhibit an optimistic bias but rather a pessimistic bias (as observed for R^2). However, this can be attributed to the fact that both $\widehat{\text{PE}}_{\text{train}}$ based on CV and $\widehat{\text{PE}}_{\text{new}}$ are affected by data leakage stemming from a violation of the assumption that all observations are independently drawn from the same distribution (see Section 2.4.2 and Supplementary Section A). This type of data leakage is distinct from the leakage caused by the overlap between the data used for model generation and evaluation, which is the primary focus of this paper. Specifically, the COMPANION data set exhibits a clustering structure that is not accounted for during the split into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{new}}$ or during the creation of CV splits on $\mathcal{D}_{\text{train}}$, resulting in a potential optimistic bias for both $\widehat{\text{PE}}_{\text{new}}$ (due to the initial split) and $\widehat{\text{PE}}_{\text{train}}$ (due to the CV splits). As $\widehat{\text{PE}}_{\text{train}}$ is also subject to a larger clustering-induced optimistic bias than $\widehat{\text{PE}}_{\text{new}}$, the bias does not cancel out when taking their difference and is therefore evident in Figure 5. Notably, the different levels of clustering-induced optimistic bias in $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$ appear to have less impact on R^2, where, as described above, the prediction error differences are mostly negative. Further details on the impact of the clustering structure on the results, including an explanation of why it was not considered when performing the splits, are provided in Supplementary Section B.5.

The additional source of optimistic bias introduced by the clustering structure of the data is also relevant when interpreting the prediction error differences in the analysis settings with HP tuning. While our primary focus here is on overlap-induced data leakage that arises since the observations used for the CV-based error estimation have already been seen during HP tuning (thus hindering the detection of potential overtuning), we have to consider that any observed optimistic bias may as well stem from clustering-induced data leakage. Consequently, we compare the prediction error differences in analysis settings with HP tuning to those in settings without tuning (where only clustering-induced data leakage is present) rather than directly comparing them to zero. Based on this assessment, the impact of overlap-induced data leakage on $\widehat{\text{PE}}_{\text{train}}$ appears to be limited. This is particularly true for RMSE, where the CV error differences are generally comparable to those resulting from the I-no tuning procedure. For R^2, the median differences tend to be closer to zero compared to the I-no tuning procedure. In some analysis settings involving the smaller sample size and the CART algorithm, there is even a positive median difference (with the largest median difference of 0.018 observed in the setting where II-automated-PA is used in combination with the smaller sample size and the CART algorithm). Consequently, there appears to be a small overtuning effect that is not detected by the CV

error due to overlap-induced data leakage. However, the median differences are too close to zero, and the variation within each analysis setting is too large to definitively determine which bias ultimately predominates, i.e. whether the CV error is overall optimistic or pessimistic in these settings.

Nested CV error In the analysis settings using the II-automated-A or II-automated-PA procedures for model generation, the prediction error differences of the nested CV error can also be analyzed. As expected, we observe the tendency for the nested CV error to be more pessimistic than the simple CV error (indicated by the smaller differences compared to the CV error; however, in some settings, the median differences for simple and nested CV errors are approximately equal). Although the nested CV error is not affected by the optimistic bias that may result from undetected overtuning effects (see Section 4.2.1), the median differences for RMSE are positive, indicating the presence of an optimistic bias. As discussed above for the simple CV error, this is due to the clustering-induced optimistic bias, which appears to outweigh the pessimistic bias typically associated with nested resampling. In the analysis settings using R^2 as performance measure, the distribution of the prediction error differences indicates that the nested CV error is pessimistically biased.

To summarize, the choice of model generation and evaluation procedure generally affects the difference between the prediction error estimates derived from available data and new data. As expected, when the evaluation procedure is based on the apparent error, the resulting estimate exhibits an optimistic bias, which varies depending on the model generation procedure. As likewise expected, the simple CV error is less optimistic than the apparent error, while the nested CV error is even less optimistic. The corresponding prediction error differences are less variable across model generation procedures compared to the apparent error. For simple CV, this indicates that, in the considered experimental setup, the tuning procedures do not introduce relevant overtuning effects on error estimation. Instead, the main source of bias for simple CV is either the clustering-induced optimistic bias (or, more precisely, the different bias level relative to $\widehat{\text{PE}}_{\text{new}}$) or the pessimistic bias arising from the use of fewer observations during evaluation. This also holds true for the nested CV error.

6 Discussion and conclusion

This paper reviewed and empirically demonstrated the implications and potential pitfalls of HP tuning in the generation and evaluation of prediction models from the perspective of applied ML users, with a specific focus on the distinction between preprocessing and algorithm HPs. While HP tuning is generally a powerful tool for improving model performance, it also introduces potential sources of error. In the model generation process, failing to select an adequate tuning procedure can result in a prediction model that performs no better, or even worse, than a model using default HP settings. During model evaluation, failing to properly account for HP

tuning can lead to optimistically biased prediction error estimates. The risk of such errors is especially high for preprocessing HPs, as they are often tuned subconsciously.

To provide different examples of model generation and evaluation procedures in the context of HP tuning and to examine their impact on the difference between prediction error estimates from available and new data, we conducted an illustrative study using a real-world prediction problem from palliative care medicine. Although both the apparent error and CV error can, in theory, be optimistically biased when HPs are tuned, this was consistently true only for the apparent error (with the highest optimistic bias occurring in analysis settings that imitated manual tuning of preprocessing HPs without considering algorithm HPs). In contrast, the prediction error differences for the CV error appeared not to be considerably compromised by data leakage, as these differences were comparable to the analysis settings without HP tuning.

In addition to explicitly considering preprocessing HPs and manual tuning procedures, our illustrative study stands out from other investigations on HP tuning by not only using real data but also building most of the setup (including the learning pipeline, HPs, and performance measures) on a real-world project. While this ensures that the observed results are realistic and not derived from overly simplified or extreme setups, they are not generalizable beyond this specific context because the considered real-world project and the derived setup are not representative of other ML applications. By using real data, our illustration was also limited in that we could only compare the prediction error estimates from the available data set to those from a new data set (which, due to the clustering structure, was also over-optimistic) instead of comparing it to the true prediction errors. Nevertheless, it was still possible to compare differences across analysis settings and derive tendencies. Finally, the illustration could have been extended by treating the learning algorithm as a tunable HP. However, with the given setup, doing so would offer limited insights, as it is reasonably predictable that the resampling-based tuning procedures would select the CIT algorithm, while the tuning procedures based on the apparent error would favor the CART algorithm.

Based on these conceptual and empirical insights, it is clear that to ensure HP tuning becomes a benefit rather than a pitfall, applied ML users must take care throughout the entire model development process. First, they should thoroughly consider which HPs (including preprocessing HPs) are to be tuned and which are not. An adequate tuning procedure that fits the specific prediction problem should then be specified. Unfortunately, this is typically non-trivial, as it depends on various factors such as sample size and the specific HPs to be tuned. More research is needed to better guide users in this respect (see Bischl et al., 2023, for an overview of current recommendations). In general, it is recommended to use automated tuning procedures instead of manual ones (see again Bischl et al., 2023, for automated tuning implementations in `R` and `Python`). If automated tuning is not feasible, users should at least ensure that the manual tuning procedure is error-free, reproducible, and resampling-based. For model evaluation, only two evaluation procedures are guaranteed to be unaffected by data leakage caused by HP tuning: (i) nested resampling (if the entire data set is used for model generation) or (ii) a permanent

(outer) holdout (if only a subset of the available data is used for model generation). However, similar to the tuning procedure, there is a lack of guidance on how to choose between these approaches and how to specify them (e.g., which resampling methods to use for nested resampling). Although simple resampling may turn out to be a viable option in some applications (including our example), this can generally not be known in advance. Therefore, we discourage its use in settings involving HP tuning, as well as any other evaluation procedures that could result in data leakage.

Regardless of how model generation and evaluation are performed, it is essential that they and all other relevant details (e.g., the complete learning pipeline and its HPs) are transparently reported in both code and text form. For this purpose, users may rely on checklists such as REFORMS (Kapoor et al., 2024; intended for all applied research fields using ML) or TRIPOD+AI (Collins, Moons, et al., 2024; intended for clinical prediction models). While transparency does not imply correctness, it allows readers to identify potential issues, such as data leakage, and to critically interpret the claimed model performance. Moreover, it emphasizes the existence and importance of preprocessing and its HPs, while the current lack of transparency can create the impression that the data were not preprocessed at all or that no alternative preprocessing options were explored. To further enhance transparency and encourage applied ML users to be more intentional about their choices, it is also possible to preregister the entire model development process, for example, by using the template proposed by Hofman et al., 2023.

In conclusion, by addressing the implications and pitfalls of HP tuning from an applied perspective and emphasizing often-overlooked aspects, we hope that this review can further enhance the quality of ML-based predictive modeling.

Funding Information

This work was supported by the German Research Foundation (BO3139/9-1, BO3139/7) to ALB. The authors of this work take full responsibility for its content.

Acknowledgments

The authors thank Patrick Callahan for language corrections and Julian Lange for useful literature input.

Conflicting interests

The authors have declared no conflicts of interest for this article.

References

Abernethy, A. P., Shelby-James, T., Fazekas, B. S., Woods, D., & Currow, D. C. (2005). The Australia-modified Karnofsky Performance Status (AKPS) scale: A revised scale for

contemporary palliative care clinical practice [ISRCTN81117481]. *BMC Palliative Care*, *4*, 7. https://doi.org/10.1186/1472-684x-4-7

Andaur Navarro, C. L., Damen, J. A. A., Takada, T., Nijman, S. W. J., Dhiman, P., Ma, J., Collins, G. S., Bajpai, R., Riley, R. D., Moons, K. G. M., & Hooft, L. (2021). Risk of bias in studies on prediction models developed using supervised machine learning techniques: Systematic review. *BMJ*, *375*, n2281. https://doi.org/10.1136/bmj.n2281

Andaur Navarro, C. L., Damen, J. A. A., van Smeden, M., Takada, T., Nijman, S. W. J., Dhiman, P., Ma, J., Collins, G. S., Bajpai, R., Riley, R. D., Moons, K. G. M., & Hooft, L. (2023). Systematic review identifies the design and methodological conduct of studies on machine learning-based prediction models. *Journal of Clinical Epidemiology*, *154*, 8–22. https://doi.org/10.1016/j.jclinepi.2022.11.015

Ball, P. (2023). Is AI leading to a reproducibility crisis in science? *Nature*, *624*(7990), 22–25. https://doi.org/10.1038/d41586-023-03817-6

Bartz, E., Bartz-Beielstein, T., Zaefferer, M., & Mersmann, O. (2023). *Hyperparameter tuning for machine and deep learning with r: A practical guide*. Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5170-1

Binder, M., & Pfisterer, F. (2024). Sequential pipelines. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/sequential_pipelines.html

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/https://doi.org/10.1002/widm.1484

Boulesteix, A.-L., Hable, R., Lauer, S., & Eugster, M. J. A. (2015). A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, *69*(3), 201–212. https://doi.org/10.1080/00031305.2015.1005128

Boulesteix, A.-L., & Strobl, C. (2009). Optimal classifier selection and negative bias in error rate estimation: An empirical study on high-dimensional prediction. *BMC Medical Research Methodology*, *9*, 85. https://doi.org/10.1186/1471-2288-9-85

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth. https://doi.org/10.1201/9781315139470

Casalicchio, G., & Burk, L. (2024). Evaluation and benchmarking. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/evaluation_and_benchmarking.html

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, *11*, 2079–2107.

Collins, G. S., Dhiman, P., Ma, J., Schlussel, M. M., Archer, L., Van Calster, B., Harrell, F. E., Martin, G. P., Moons, K. G. M., van Smeden, M., Sperrin, M., Bullock, G. S., & Riley, R. D. (2024). Evaluation of clinical prediction models (part 1): From development to external validation. *BMJ*, *384*, e074819. https://doi.org/10.1136/bmj-2023-074819

Collins, G. S., Moons, K. G. M., Dhiman, P., Riley, R. D., Beam, A. L., Van Calster, B., Ghassemi, M., Liu, X., Reitsma, J. B., van Smeden, M., Boulesteix, A.-L., Camaradou, J. C., Celi, L. A., Denaxas, S., Denniston, A. K., Glocker, B., Golub, R. M., Harvey, H., Heinze, G., ... Logullo, P. (2024). TRIPOD+AI statement: Updated guidance for reporting clinical prediction models that use regression or machine learning methods. *BMJ*, *385*, e078378. https://doi.org/10.1136/bmj-2023-078378

de Hond, A. A. H., Leeuwenberg, A. M., Hooft, L., Kant, I. M. J., Nijman, S. W. J., van Os, H. J. A., Aardoom, J. J., Debray, T. P. A., Schuit, E., van Smeden, M., Reitsma, J. B., Steyerberg, E. W., Chavannes, N. H., & Moons, K. G. M. (2022). Guidelines and quality criteria for artificial intelligence-based prediction models in healthcare: A scoping review. *npj Digital Medicine*, *5*, 2. https://doi.org/10.1038/s41746-021-00549-7

Debray, T. P. A., Collins, G. S., Riley, R. D., Snell, K. I. E., Van Calster, B., Reitsma, J. B., & Moons, K. G. M. (2023). Transparent reporting of multivariable prediction models developed or validated using clustered data (TRIPOD-Cluster): Explanation and elaboration. *BMJ*, *380*, e071058. https://doi.org/10.1136/bmj-2022-071058

Dhiman, P., Ma, J., Andaur Navarro, C. L., Speich, B., Bullock, G., Damen, J. A. A., Hooft, L., Kirtley, S., Riley, R. D., Van Calster, B., Moons, K. G. M., & Collins, G. S. (2022a). Methodological conduct of prognostic prediction models developed using machine learning in oncology: A systematic review. *BMC Medical Research Methodology*, *22*, 101. https://doi.org/10.1186/s12874-022-01577-x

Dhiman, P., Ma, J., Andaur Navarro, C. L., Speich, B., Bullock, G., Damen, J. A. A., Hooft, L., Kirtley, S., Riley, R. D., Van Calster, B., Moons, K. G. M., & Collins, G. S. (2022b). Risk of bias of prognostic models developed using machine learning: A systematic review in oncology. *Diagnostic and Prognostic Research*, *6*, 13. https://doi.org/10.1186/s41512-022-00126-w

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87. https://doi.org/10.1145/2347736.2347755

Dunias, Z. S., Van Calster, B., Timmerman, D., Boulesteix, A.-L., & van Smeden, M. (2024). A comparison of hyperparameter tuning procedures for clinical prediction models: A simulation study. *Statistics in Medicine*, *43*(6), 1119–1134. https://doi.org/10.1002/sim.9932

Eagar, K., Green, J., & Gordon, R. (2004). An Australian casemix classification for palliative care: Technical development and results. *Palliative Medicine*, *18*(3), 217–226. https://doi.org/10.1191/0269216304pm875oa

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, *81*(394), 461–470. https://doi.org/10.1080/01621459.1986.10478291

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search. In *The springer series on challenges in machine learning* (pp. 63–77). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_3

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer International Publishing. https://doi.org/10.1007/978-3-030-05318-5_1

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., & Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods*, *50*(5), 2016–2034. https://doi.org/10.3758/s13428-017-0971-x

Foss, N., & Kotthoff, L. (2024). Data and basic modeling. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/data_and_basic_modeling.html

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd). Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Hodiamont, F., Schatz, C., Gesell, D., Leidl, R., Boulesteix, A.-L., Nauck, F., Wikert, J., Jansky, M., Kranz, S., & Bausewein, C. (2022). COMPANION: Development of a patient-centred complexity and casemix classification for adult palliative care patients based on needs and resource use – a protocol for a cross-sectional multi-centre study. *BMC Palliative Care*, *21*, 18. https://doi.org/10.1186/s12904-021-00897-x

Hofman, J. M., Chatzimparmpas, A., Sharma, A., Watts, D. J., & Hullman, J. (2023). Pre-registration for predictive modeling. *arXiv:2311.18807v1 [cs.LG]*. https://arxiv.org/abs/2311.18807

Hofman, J. M., Sharma, A., & Watts, D. J. (2017). Prediction and explanation in social systems. *Science*, *355*(6324), 486–488. https://doi.org/10.1126/science.aal3856

Hornung, R., Bernau, C., Truntzer, C., Wilson, R., Stadler, T., & Boulesteix, A.-L. (2015). A measure of the impact of CV incompleteness on prediction error estimation with application to PCA and normalization. *BMC Medical Research Methodology*, *15*, 95. https://doi.org/10.1186/s12874-015-0088-9

Hornung, R., Nalenz, M., Schneider, L., Bender, A., Bothmann, L., Bischl, B., Augustin, T., & Boulesteix, A.-L. (2023). Evaluating machine learning models in non-standard settings: An overview and new findings. *arXiv:2310.15108v1 [stat.ML]*. https://arxiv.org/abs/2310.15108

Hosseini, M., Powell, M., Collins, J., Callahan-Flintoft, C., Jones, W., Bowman, H., & Wyble, B. (2020). I tried a bunch of things: The dangers of unexpected overfitting in classification

of brain data. *Neuroscience & Biobehavioral Reviews*, *119*, 456–467. https://doi.org/10.1016/j.neubiorev.2020.09.036

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Hothorn, T., & Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, *16*, 3905–3909. https://jmlr.org/papers/v16/hothorn15a.html

Kapoor, S., Cantrell, E. M., Peng, K., Pham, T. H., Bail, C. A., Gundersen, O. E., Hofman, J. M., Hullman, J., Lones, M. A., Malik, M. M., Nanayakkara, P., Poldrack, R. A., Raji, I. D., Roberts, M., Salganik, M. J., Serra-Garcia, M., Stewart, B. M., Vandewiele, G., & Narayanan, A. (2024). REFORMS: Consensus-based recommendations for machine-learning-based science. *Science Advances*, *10*(18), eadk3452. https://doi.org/10.1126/sciadv.adk3452

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9), 100804. https://doi.org/https://doi.org/10.1016/j.patter.2023.100804

Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, *6*(4), 15. https://doi.org/10.1145/2382577.2382579

Klau, S., Martin-Magniette, M.-L., Boulesteix, A.-L., & Hoffmann, S. (2020). Sampling uncertainty versus method uncertainty: A general framework with applications to omics biomarker selection. *Biometrical Journal*, *62*(3), 670–687. https://doi.org/10.1002/bimj.201800309

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer New York. https://doi.org/10.1007/978-1-4614-6849-3

Kuhn, M., Wickham, H., & Hvitfeldt, E. (2024). *recipes: Preprocessing and feature engineering steps for modeling* [R package version 1.0.10, https://recipes.tidymodels.org/]. https://github.com/tidymodels/recipes

Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., & Bischl, B. (2019). mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software*, *4*(44), 1903. https://doi.org/10.21105/joss.01903

Lones, M. A. (2024). How to avoid machine learning pitfalls: a guide for academic researchers. *arXiv:2108.02497v5 [cs.LG]*. http://arxiv.org/abs/2108.02497

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable* (2nd). https://christophm.github.io/interpretable-ml-book

Murtagh, F. E. M., Guo, P., Firth, A., Yip, K. M., Ramsenthaler, C., Douiri, A., Pinto, C., Pask, S., Dzingina, M., Davies, J. M., O'Brien, S., Edwards, B., Groeneveld, E. I.,

Hocaoglu, M., Bausewein, C., & Higginson, I. J. (2023). A casemix classification for those receiving specialist palliative care during their last year of life across England: The C-CHANGE research programme. *Programme Grants for Applied Research*, *11*(7), 1–78. https://doi.org/10.3310/plrp4875

Murtagh, F. E. M., Ramsenthaler, C., Firth, A., Groeneveld, E. I., Lovell, N., Simon, S. T., Denzel, J., Guo, P., Bernhardt, F., Schildmann, E., van Oorschot, B., Hodiamont, F., Streitwieser, S., Higginson, I. J., & Bausewein, C. (2019). A brief, patient- and proxy-reported outcome measure in advanced illness: Validity, reliability and responsiveness of the Integrated Palliative care Outcome Scale (IPOS). *Palliative Medicine*, *33*(8), 1045–1057. https://doi.org/10.1177/0269216319854264

Nagler, T., Schneider, L., Bischl, B., & Feurer, M. (2024). Reshuffling resampling splits can improve generalization of hyperparameter optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)* (pp. 40486–40533). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2024/hash/47811ee68103bfcde7ca2223fccefb3a-Abstract-Conference.html

Neunhoeffer, M., & Sternberg, S. (2019). How cross-validation can go wrong and what to do about it. *Political Analysis*, *27*(1), 101–106. https://doi.org/10.1017/pan.2018.39

Ng, A. Y. (1997). Preventing "overfitting" of cross-validation data. In D. H. Fisher (Ed.), *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997)* (pp. 245–253). Morgan Kaufmann Publishers Inc.

Pfob, A., Lu, S. C., & Sidey-Gibbons, C. (2022). Machine learning in medicine: A practical introduction to techniques for data pre-processing, hyperparameter tuning, and model comparison. *BMC Medical Research Methodology*, *22*, 282. https://doi.org/10.1186/s12874-022-01758-8

Poldrack, R. A., Huckins, G., & Varoquaux, G. (2020). Establishment of best practices for evidence for prediction: A review. *JAMA Psychiatry*, *77*(5), 534–540. https://doi.org/10.1001/jamapsychiatry.2019.3671

Probst, P., & Boulesteix, A.-L. (2018). To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research*, *18*(181), 1–18. http://jmlr.org/papers/v18/17-269.html

Probst, P., Boulesteix, A.-L., & Bischl, B. (2019). Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, *20*(53), 1–32.

Quinlan, J. R., & Cameron-Jones, R. M. (1995). Oversearching and layered search in empirical learning. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, *2*, 1019–1024.

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Rosenblatt, M., Tejavibulya, L., Jiang, R., Noble, S., & Scheinost, D. (2024). Data leakage inflates prediction performance in connectome-based machine learning models. *Nature Communications*, *15*, 1829. https://doi.org/10.1038/s41467-024-46150-w

Sela, R. J., & Simonoff, J. S. (2011). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*(2), 169–207. https://doi.org/10.1007/s10994-011-5258-3

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366. https://doi.org/10.1177/0956797611417632

Simon, R., Radmacher, M. D., Dobbin, K., & McShane, L. M. (2003). Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *Journal of the National Cancer Institute*, *95*(1), 14–18. https://doi.org/10.1093/jnci/95.1.14

Simon, R. (2007). Resampling strategies for model assessment and selection. In *Fundamentals of data mining in genomics and proteomics* (pp. 173–186). Springer US. https://doi.org/10.1007/978-0-387-47509-7_8

Steyerberg, E. W. (2019). *Clinical prediction models: A practical approach to development, validation, and updating* (2nd). Springer International Publishing. https://doi.org/10.1007/978-3-030-16399-0

Stüber, A. T., Coors, S., Schachtner, B., Weber, T., Rügamer, D., Bender, A., Mittermeier, A., Öcal, O., Seidensticker, M., Ricke, J., Bischl, B., & Ingrisch, M. (2023). A comprehensive machine learning benchmark study for radiomics-based survival analysis of CT imaging data in patients with hepatic metastases of CRC. *Investigative Radiology*, *58*(12), 874–881. https://doi.org/10.1097/rli.0000000000001009

Therneau, T., & Atkinson, B. (2022). *rpart: Recursive Partitioning and Regression Trees* [R package version 4.1.19]. https://CRAN.R-project.org/package=rpart

Thomas, J. (2024). Preprocessing. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/preprocessing.html

Van Calster, B., Steyerberg, E. W., Wynants, L., & van Smeden, M. (2023). There is no such thing as a validated prediction model. *BMC Medicine*, *21*, 70. https://doi.org/10.1186/s12916-023-02779-w

van Royen, F. S., Asselbergs, F. W., Alfonso, F., Vardas, P., & van Smeden, M. (2023). Five critical quality criteria for artificial intelligence-based prediction models. *European Heart Journal*, *44*(46), 4831–4834. https://doi.org/10.1093/eurheartj/ehad727

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*, 91. https://doi.org/10.1186/1471-2105-7-91

Wainer, J., & Cawley, G. (2021). Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, *182*, 115222. https://doi.org/10.1016/j.eswa.2021.115222

Waldron, L., Pintilie, M., Tsao, M.-S., Shepherd, F. A., Huttenhower, C., & Jurisica, I. (2011). Optimized application of penalized regression methods to diverse genomic data. *Bioinformatics*, *27*(24), 3399–3406. https://doi.org/10.1093/bioinformatics/btr591

Wright, M. N. (2024). Feature selection. In B. Bischl, R. Sonabend, L. Kotthoff, & M. Lang (Eds.), *Applied machine learning using mlr3 in R*. CRC Press. https://mlr3book.mlr-org.com/feature_selection.html

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, *17*(2), 492–514. https://doi.org/10.1198/106186008X319331

Supplementary Material

A Other leakage types

As stated in Section 2.4.2, Kapoor and Narayanan, 2023 identify three general types of data leakage, which may arise from: (i) overlap between the data used for model generation and evaluation, (ii) violation of the assumption that all observations are independently drawn from the same distribution, or (iii) use of illegitimate features. While our paper primarily addresses overlap-induced data leakage, we will now provide additional details on the other two types.

A.1 Violation of the i.i.d. assumption

In the following, we first consider the case of Setting I with $\mathcal{D}_{\text{train}} = \mathcal{D}$ and discuss the implications for $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ and Setting II afterwards.

Even with a strict separation between the data used for model generation and evaluation, achieved through the use of resampling methods, data leakage can still occur if the assumption that all observations in $\mathcal{D}_{\text{train}}$ are independently drawn from the same distribution is violated. This assumption, also known as the i.i.d. assumption, was stated in Section 2.1. Non-i.i.d. settings may, for example, arise when $\mathcal{D}_{\text{train}}$ is a clustered data set, i.e. when the observations originate from different clusters (e.g., study centers). Observations within clusters are typically more similar than observations between clusters, where similarity can refer to both the feature vector $\boldsymbol{x}^{(i)}$ or the outcome $y^{(i)}$ (Hornung et al., 2023). If the prediction model is intended to be applied to observations from other clusters than those present in $\mathcal{D}_{\text{train}}$ in the future, resampling methods that are based on random sampling (i.e. ignoring the cluster structure) will be optimistically biased since in each resampling iteration, the observations in $\mathcal{D}_{\text{test}}$ are more similar to $\mathcal{D}'_{\text{train}}$ than observations originating from new clusters (Hornung et al., 2023; Kapoor & Narayanan, 2023; Rosenblatt et al., 2024). Although the level of optimistic bias depends on the specific clustering structure (e.g., cluster size and correlation within clusters), it is generally recommended to perform grouped resampling at cluster level, where all observations in a cluster are either assigned to $\mathcal{D}'_{\text{train}}$ or $\mathcal{D}_{\text{test}}$ in each resampling iteration (Bischl et al., 2023; Hornung et al., 2023). In the context of healthcare research, this type of resampling is referred to as internal-external validation (Collins, Dhiman, et al., 2024; Debray et al., 2023). For other examples of non-i.i.d. settings and corresponding resampling methods, see Hornung et al. (2023) and the references therein.

Our elaborations also apply to the case of Setting I with $\mathcal{D}_{\text{train}} \subset \mathcal{D}$, with a permanent holdout used instead of a (temporary) resampling method; here, one simply replaces $\mathcal{D}_{\text{train}}$ with $\mathcal{D}$ and $\mathcal{D}'_{\text{train}}$ with $\mathcal{D}_{\text{train}}$.

In Setting II, where resampling is typically used for both model generation (tuning) and evaluation, data leakage due to the violation of the i.i.d. assumption biases the prediction error estimate of the final model only when the non-i.i.d. data structure is ignored during model evaluation. This occurs specifically in the outer resampling loop of nested resampling (for

$\mathcal{D}_{\text{train}} = \mathcal{D}$) or in the permanent outer holdout (for $\mathcal{D}_{\text{train}} \subset \mathcal{D}$). However, it is recommended to also take into account the non-i.i.d. data structure during tuning, both for the final prediction model and, if nested resampling is used, within the inner resampling loop, to ensure consistency (Hornung et al., 2023).

A.2 Use of illegitimate features

If $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ include features that are generally not available for new observations to which the model will be applied in practice, these features can be considered illegitimate, and if included in the final prediction model, constitute another type of data leakage. An example raised by Kapoor and Narayanan, 2023 is the use of anti-hypertensive drugs as a feature for predicting hypertension. Note that this type of data leakage is conceptually different from the other two types, as it stems from a design issue that is independent of the model evaluation procedure.

B Additional information on the empirical illustration

B.1 Descriptive statistics

Table S1 provides descriptive statistics of the COMPANION data set used in the empirical illustration.

B.2 Preprocessing steps

B.2.1 Initial preprocessing steps

In the following, we describe the parameterless and pre-specified preprocessing steps that are applied to the full COMPANION data set in its rawest version available. Note that the raw data set is on patient contact level, which was the unit for data collection (Hodiamont et al., 2022). The initial preprocessing steps are:

(i) data cleaning steps (e.g., correct variable types and labels),

(ii) the removal of contacts with palliative care phase "bereavement", AKPS = 0 ("dead"), or costs = 0,

(iii) the aggregation of the contact level data into palliative care phase level data (the outcome is constructed by summing the costs of all patient contacts and dividing by the number of days in the corresponding phase; for features that may vary during a phase, the highest value of the first day is used),

(iv) the removal of palliative care phases (one phase with an extreme and implausible cost value is removed; phases with "missing" values in either one or both cognitive features or in one of the individual IPOS features are removed; phases with "missing" or "cannot assess" in the AKPS feature are removed), and

(v) the replacement of "cannot assess" values with "absent" in the two cognitive features.

Table S1: Distribution of the outcome variable and features in the COMPANION data set after applying the initial preprocessing steps (described in Supplementary Section B.2.1). In addition, two preprocessing steps from the learning pipeline $\mathcal{I}$ (see Section 5.2.2 and Supplementary Section B.2.2) have been performed: the correction of costs and the aggregation of the IPOS score (default version).

	n =1,449
Average cost per day per palliative care phase (€)	
Mean (SD)	49.0 (43.1)
Median [Min, Max]	35.9 [0.315, 357]
Palliative care phase	
stable	453 (31.3%)
unstable	281 (19.4%)
deteriorating	486 (33.5%)
terminal	229 (15.8%)
Age (years)	
Mean (SD)	74.7 (12.2)
Median [Min, Max]	76.0 [23, 102]
Confusion	
absent	950 (65.6%)
mild	248 (17.1%)
moderate	144 (9.9%)
severe	107 (7.4%)
Agitation	
absent	837 (57.8%)
mild	306 (21.1%)
moderate	217 (15.0%)
severe	89 (6.1%)
AKPS	
(10) comatose or barely rousable	79 (5.5%)
(20) totally bedfast and requiring extensive nursing care by professionals and/or family	381 (26.3%)
(30) almost completely bedfast	242 (16.7%)
(40) in bed more than 50% of the time	270 (18.6%)
(50) considerable assistance and frequent medical care required	265 (18.3%)
(60) able to care for most needs; but requires occasional assistance	151 (10.4%)
(70) cares for self; unable to carry on normal activity or to do active work	38 (2.6%)
(80) normal activity with effort; some signs or symptoms of disease	14 (1.0%)
(90) able to carry on normal activity; minor sign of symptoms of disease	9 (0.6%)
IPOS total score	
Mean (SD)	24.8 (7.98)
Median [Min, Max]	25.0 [2.00, 55.0]

These preprocessing steps yield a data set with 1,449 observations.

B.2.2 Preprocessing steps in the learning pipeline

In this section, we detail the six preprocessing steps of the learning pipeline $\mathcal{I}$ that is applied in each training process, including their associated HPs. An overview of these preprocessing steps is given in Figure 4.

Correction of costs As stated in Section 5.1, the outcome variable $y^{(i)}$ is defined as the average cost per day in palliative care phase i, which is intended to reflect the resource needs in that phase. This variable is calculated based on the staff time used to care for a patient and their relatives on each day of the corresponding palliative care phase. However, analyses have shown that if a palliative care phase is the first phase in an episode of care (see Supplementary Section B.5 for more information on episodes of care), the staff time and thus the costs of the first day are increased regardless of the complexity of the palliative care situation (e.g., due to time-consuming admission interviews). For this reason, the first-day costs of the first phase of an episode are adjusted using a factor based on comparisons with the costs of the first days in later phases of an episode. This factor is initially calculated for each palliative care team and then averaged to obtain a single overall correction factor, denoted as $\theta_{correct}$. This preprocessing step accordingly includes a parameter that must be estimated from the data set, though it does not involve any HPs in our illustration. Moreover, it is a step that modifies the outcome (albeit slightly), not for compatibility with the learning algorithm, but to change the interpretation of the prediction model, which now intends to predict a corrected version of the outcome. Accordingly, this step is also applied during prediction.

Removal of cost outliers The distribution of the outcome variable in the COMPANION data set is right skewed, i.e. some palliative care phases have exceptionally high costs (see Table S1). Since it is not possible to definitively attribute these values to data entry errors, they are not permanently removed from the data set. However, since the prediction values calculated by the corresponding decision tree algorithm in each terminal node can be sensitive to outliers, removing cost outliers during the training process could improve model performance. Importantly, this preprocessing step is only applied during training and not during prediction, i.e. when the final prediction model is used to make predictions on a data set, no cost outliers are removed. Removing them during prediction could artificially improve the model's performance, as cost outliers are typically difficult to predict correctly (see also Kapoor & Narayanan, 2023). The definition of outliers is generally not straightforward, as many possible options exist (Kuhn & Johnson, 2013; Steyerberg, 2019). We denote the corresponding HP as $\lambda_{outlier}$. In our illustration, we define all cost values higher than the $\lambda_{outlier}$th cost percentile as outliers, with $\lambda_{outlier} \in \{100, 99, 95, 90\}$. If $\lambda_{outlier} = 100$ (the default value), no outliers are removed. Note that this preprocessing step includes the parameter $\theta_{outlier}$, which corresponds to the percentile calculated according to $\lambda_{outlier}$.

Handling of "cannot assess" values in IPOS features As outlined in Section 5.1, the set of features to generate the prediction model includes the Integrated Palliative care Outcome Scale (IPOS; Murtagh et al., 2019), which is a score based on 17 individual features covering physical symptoms, psycho-social burden, family needs, and practical problems. Each of the 17 features is ordinal and can take values from 0 to 4, where 0 and 4 correspond to the least and highest symptom or concern severity, respectively. For example, for the features IPOS-"Pain" and IPOS-"Shortness of Breath", a value of 0 corresponds to "not at all" and a value of 4 corresponds to "overwhelmingly" (see Figure S1 for an overview of all 17 features). In its default version (see the next preprocessing step), the IPOS score is constructed by summing all 17 features, resulting in a score that ranges from 0 to 68. However, each IPOS feature also includes missing values, which are either due to missing data entries (coded as "missing") or because the response option "cannot assess" was selected during the IPOS assessment. For example, assessing whether a patient is burdened by pain (IPOS-"Pain") can be challenging for clinical staff if the patient is comatose.

While observations affected by the first type of missing values ("missing") do not occur often and are removed as part of the initial preprocessing steps described in Supplementary Section B.2.1, handling the "cannot assess" values is more challenging. If all observations with at least one "cannot assess" response were removed, almost half of the COMPANION data set would be discarded (see Table S2; this would also apply approximately to any subset $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{new}}$ of the COMPANION data set). To avoid the loss of valuable information, an alternative approach is to treat "cannot assess" values as 0 (i.e. least symptom or concern severity), based on the assumption that an unobserved burden does not initiate a care mandate and therefore does not result in costs. However, it is not clear whether this assumption is valid for observations where many or even all IPOS features are recorded as "cannot assess" (e.g., if 15 out of 17 IPOS features are recorded as "cannot assess", these features might not have been assessed at all). It could thus be a reasonable approach to set "cannot assess" values to 0 but exclude observations with many "cannot assess" values, as they potentially result in incorrect IPOS scores. Specifying the exact threshold for the maximum number of "cannot assess" values is, however, not straightforward. It can be denoted as HP λ_{ca}, and ranges from 0 to 17 (observations with more than λ_{ca} "cannot assess" values are removed; if $\lambda_{ca} = 17$, no observations are removed). In our illustration, we consider the values $\{16, 14, 12, 10\}$ for λ_{ca}, with $\lambda_{ca} = 16$ being the default.

This preprocessing step does not have any parameters. Since it removes observations, it modifies the distribution of the outcome variable. We argue that if observations with more than λ_{ca} "cannot assess" values are found to yield unreliable IPOS scores, the resulting prediction model should not be used for future observations where this criterion applies, implying that the corresponding preprocessing step alters the scope of the model (such that it cannot be used for observations with more than λ_{ca} IPOS features recorded as "cannot assess"). Accordingly, this step is also applied during the prediction process. As shown in Table S2, the change in the outcome distribution is, however, minimal because the values considered for λ_{ca} remove only

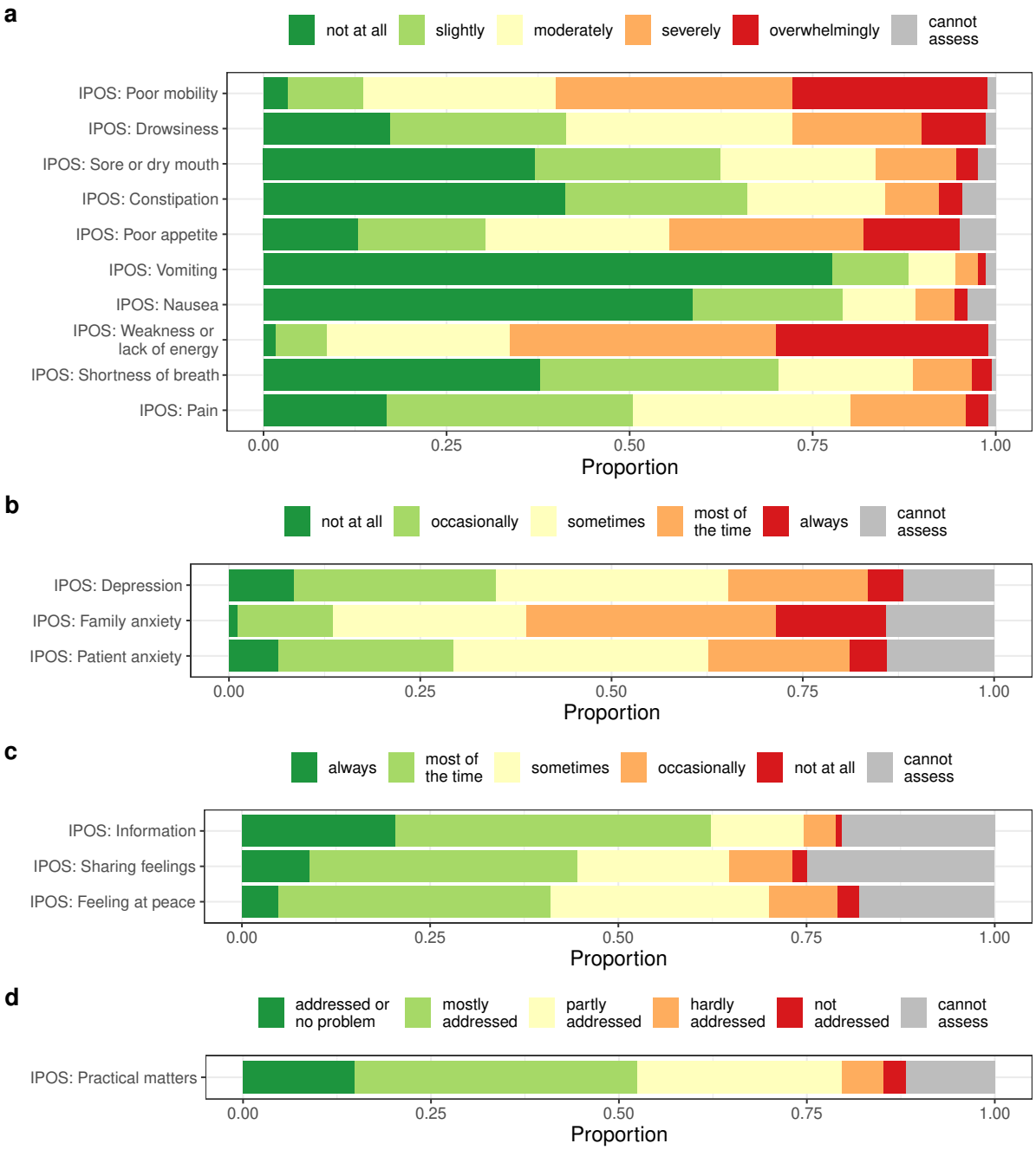

Figure S1: Distribution of the 17 individual IPOS features in the COMPANION data set after applying the initial preprocessing steps (described in Supplementary Section B.2.1). a: Physical symptoms. b: Emotional symptoms. c: Communication issues. d: Practical issues.

a small number of observations (9 observations for $\lambda_{ca} = 10$ and 0 observations for $\lambda_{ca} = 16$) from the full COMPANION data set with 1,449 observations. As discussed in Section 2.3.4, it is recommended to specify HPs of preprocessing steps that affect the outcome distribution based on user expertise rather than tuning. However, given that this step only removes a few observations and because specifying λ_{ca} based on user expertise is challenging, we argue that λ_{ca} can be tuned.

Table S2: Outcome distribution (average cost per day per palliative care phase) in the full COMPANION data set (after applying the initial preprocessing steps described in Supplementary Section B.2.1) if observations with more than $\lambda_{ca} \in \{0, 10, 12, 14, 16\}$ "cannot assess" values in the 17 individual IPOS features are removed. The minimum and maximum number of "cannot assess" values are 0 and 17, respectively.

$\lambda_{ca} = 0$	
Mean (SD)	48.62 (45.12)
Median [Min, Max]	34.96 [1.11, 356.70]
Missing	662 (45.7%)
$\lambda_{ca} = 10$	
Mean (SD)	49.03 (43.14)
Median [Min, Max]	35.91 [0.32, 356.70]
Missing	9 (0.6%)
$\lambda_{ca} = 12$	
Mean (SD)	48.98 (43.09)
Median [Min, Max]	35.91 [0.32, 356.70]
Missing	3 (0.2%)
$\lambda_{ca} = 14$	
Mean (SD)	48.99 (43.07)
Median [Min, Max]	35.92 [0.32, 356.70]
Missing	2 (0.1%)
$\lambda_{ca} = 16$	
Mean (SD)	48.98 (43.05)
Median [Min, Max]	35.92 [0.32, 356.70]
Missing	0 (0.0%)

Calculation of IPOS score After removing observations based on their individual IPOS feature values, the next preprocessing step is to construct the IPOS score from these features. Aggregating the individual IPOS features into an IPOS score can be done in several ways, and we denote the corresponding HP as λ_{ipos}. A straightforward and commonly used option is to simply sum the values of all 17 IPOS features, which we denote as IPOS-total (the default of λ_{ipos}).

Instead of aggregating all 17 IPOS features into one score, it is also possible to generate multiple IPOS scores based on the subscales in which the features can be divided (Murtagh et al., 2019). These subscales are: (i) physical symptoms (10 features), (ii) emotional symptoms (4 features), and (iii) communication/practical issues (3 features) (see Figure S1). In our illustration, we consider the generation of two subscale scores: one score that sums the features corresponding to the physical symptoms (IPOS-physical; $[0, 40]$) and one score that sums the remaining features (IPOS-others; $[0, 28]$). Note that in this case, the number of features provided to the learning algorithm increases from $p = 6$ to $p = 7$.

A third option to construct the IPOS score is to sum all 17 IPOS features as in the IPOS-total score, but recode them (before summing) as 1 if their value is $\in \{3, 4\}$ (i.e. takes one of the two

most extreme values), and 0 otherwise. This score will be referred to as the IPOS-extreme score and ranges from 0 to 17. It was developed by the COMPANION team and was motivated by the possibly too strict assumption made by the previous preprocessing step, namely that "cannot assess" values are equivalent to a value of 0. This assumption is relaxed by the IPOS-extreme score, which only requires assuming that the true value of an IPOS feature recorded as "cannot assess" is $\in \{0, 1, 2\}$ and not necessarily equal to 0.
The fourth considered IPOS score option is similar to the IPOS-extreme score, except that the features IPOS-"Pain" and IPOS-"Shortness of Breath" are excluded from the score (which now ranges from 0 to 15) and are instead provided separately on their original ordinal scale to the learning algorithm. The motivation for this version is that pain and shortness of breath may be strong predictors of the costs associated with a palliative care phase. Therefore, model performance might be improved by including IPOS-"Pain" and IPOS-"Shortness of Breath" as individual features rather than aggregating them into the IPOS-extreme score. If this IPOS option is used, the number of features provided to the learning algorithm increases from $p = 6$ to $p = 8$.
This preprocessing step does not have any parameters. Moreover, it does not alter the outcome distribution, which is why it is applied during both training and prediction.

Modification of feature "age" In the COMPANION data set, age is measured on an integer scale and ranges from 23 to 102 years (see Table S1). In its default configuration, this feature is provided to the learning algorithm on its original integer scale, without any preprocessing. Alternatively, it could be transformed into a categorical feature with six categories, using the years 50, 60, 70, 80, and 90 as cutpoints. This option could improve the model's prediction error, as, for example, the CART algorithm suffers from a selection bias towards features with many possible splits (Hothorn et al., 2006). We refer to the HP that specifies the used option as λ_{age}, with no modification of age as default. This preprocessing step has the same characteristics as the aggregation of individual IPOS features into a score (i.e. no parameters, applied during training and prediction).

Modification of feature "AKPS" The Australia-modified Karnofsky Performance Status (AKPS; Abernethy et al., 2005), which measures patients' functional status on an ordinal scale, takes values of $\{10, 20, ..., 90\}$ in the COMPANION data set, with AKPS = 10 corresponding to "comatose or barely rousable" and AKPS = 90 to "able to carry on normal activity; minor sign of symptoms of disease" (see Table S1). In its default configuration, AKPS is considered ordinal, with the three highest categories, 70, 80, and 90, merged due to their low frequency. However, it might also be reasonable to transform AKPS into an unordered categorical variable, as costs may not monotonically decrease or increase with AKPS, but could be highest when the patient has, for example, an AKPS of 50, which corresponds to "considerable assistance and frequent medical care required". In this case, we collapse the AKPS categories even further to avoid overfitting, resulting in AKPS $\in \{10\text{-}20, 30\text{-}50, 60\text{-}90\}$. We refer to the corresponding

HP as λ_{akps}, with the ordered AKPS variable as default. This preprocessing step has the same characteristics as the two previous preprocessing steps (i.e. no parameters, applied during training and prediction).

Note that for the preprocessing steps estimating parameters from the available observations (i.e. correction of costs, with $\theta_{correct}$, and removal of cost outliers, with $\theta_{outlier}$), their position in the preprocessing pipeline in relation to the steps where observations are removed (i.e. removal of outliers and handling of "cannot assess" values) is of relevance since a different set of observations might yield a different parameter estimate. Accordingly, performing the preprocessing steps in a different order could lead to (slightly) different results.
Moreover, during the execution of the illustration as described in Section 5.2.1, in some resampling iterations performed during model generation and evaluation (particularly for nested CV), it occasionally happens that certain ordinal or categorical features in the data subset for which predictions are being made contain new values that were not encountered during training. This issue occurs exclusively with the highest and/or lowest values of these features, which are less frequent in the original COMPANION data set and thus more likely to be absent in the training set. Specifically, this affects the highest value of (cognitive) agitation, the highest and lowest values of AKPS (if AKPS is not collapsed into three unordered categories), the lowest value of age (if age is transformed into a categorical feature), and the highest values of "Pain" and IPOS-"Shortness of Breath" (if the fourth option for aggregating the IPOS score is selected). In these cases, we collapse the highest and second highest and/or lowest and second lowest values when making predictions.

B.3 Performance measures

In the illustration, two performance measures are considered: RMSE and R^2. The RMSE is obtained by taking the square root of the MSE (see Section 3.1) and is expressed in the same units as the outcome variable (i.e. costs in €). It ranges from 0 to ∞, where RMSE = 0 indicates perfect prediction. The R^2 performance measure is calculated by dividing the squared error of the prediction model by the squared error of a naive model that predicts the mean and then subtracting this ratio from 1. It is a relative measure that can be interpreted as the proportion of variance in the outcome variable explained by the prediction model. The range of R^2 is $(-\infty, 1]$, with $R^2 = 1$ indicating perfect prediction and a R^2 value of 0 or less indicating that a model performs no better or worse than the naive model, respectively. In this context, a lower prediction error corresponds to a higher R^2 value. See, e.g., Kuhn and Johnson, 2013 for more details on both performance measures.

B.4 Absolute prediction error estimates and selected HPs

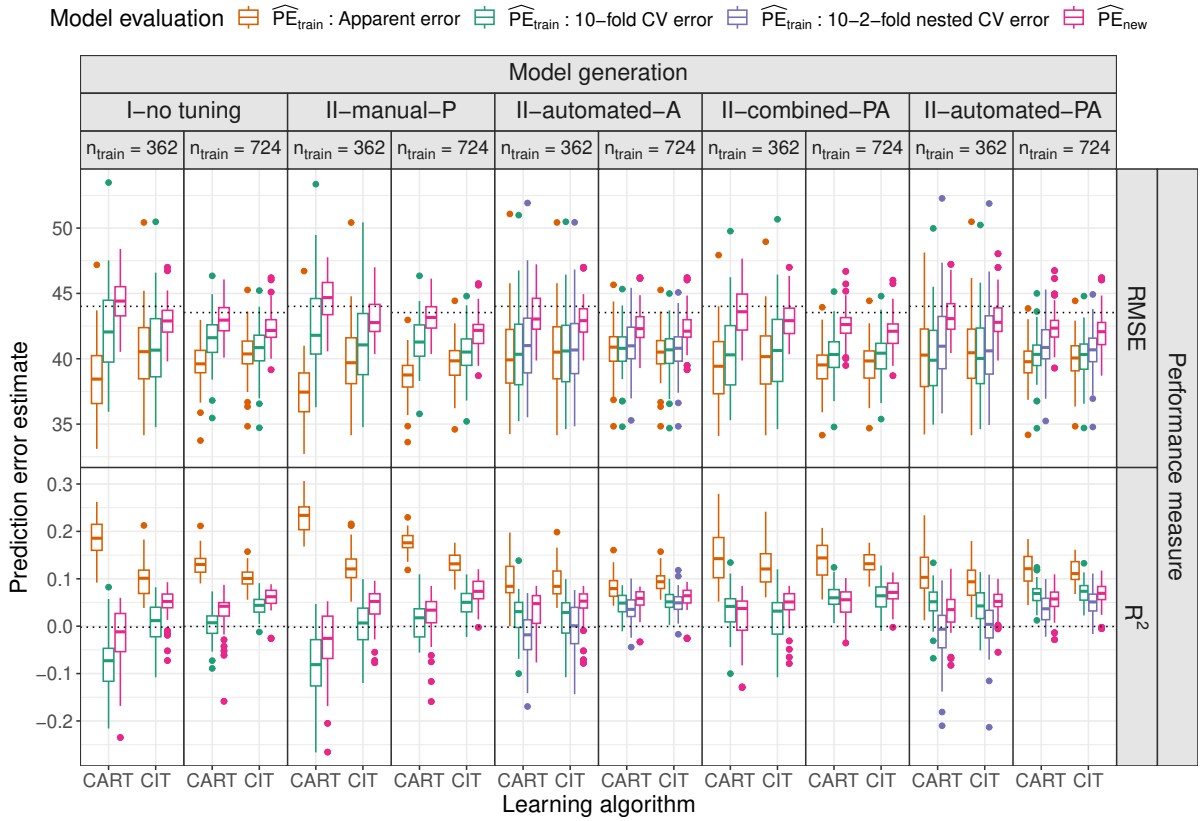

Figure S2: Absolute prediction error estimates $\widehat{\text{PE}}_{\text{train}}$ across 96 analysis settings, with each boxplot summarizing 50 repetitions of a specific setting. Additionally, absolute prediction error estimates $\widehat{\text{PE}}_{\text{new}}$ are shown. Importantly, $\widehat{\text{PE}}_{\text{new}}$ is independent of the model evaluation procedure performed on $\mathcal{D}_{\text{train}}$ and is therefore shown only for the 40 settings formed by all possible combinations of model generation procedures, performance measures, sample sizes, and learning algorithms ($5 \times 2 \times 2 \times 2 = 40$), where each boxplot again represents 50 repetitions. For reference, the dotted line represents the median prediction error estimate on $\mathcal{D}_{\text{new}}$ (averaged over the 50 repetitions) for a featureless learning algorithm, which naively predicts the mean. Taking the difference between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$ for each repetition results in Figure 5 in the main text.

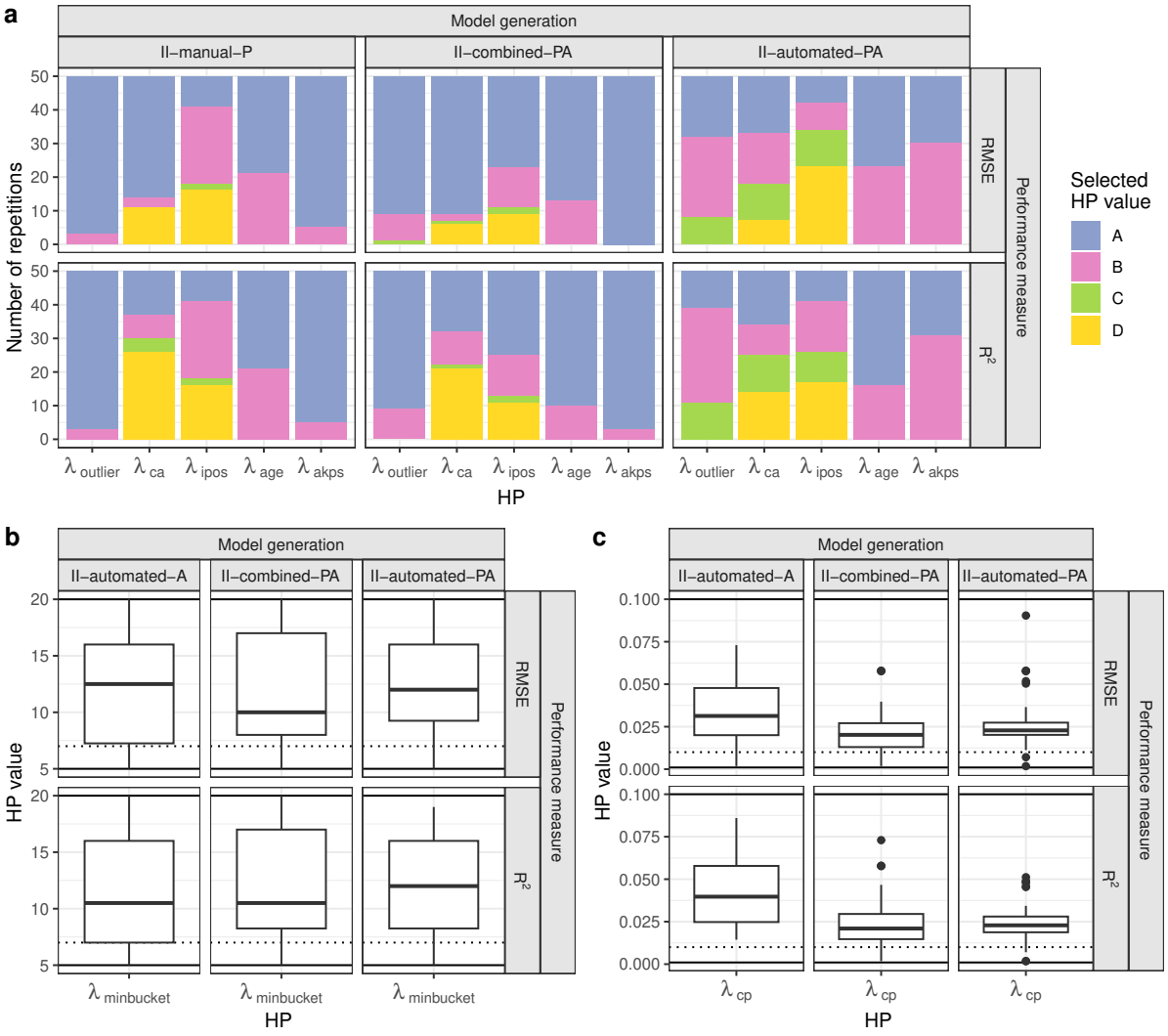

Figure S3: Selected HPs for the analysis settings where CART is used as the learning algorithm and $n_{\text{train}} = 362$. Only model generation procedures that involve tuning the corresponding HP type are shown. a: Preprocessing HPs. The labels A, B, C, and D correspond to the first, second, and, if present, subsequent values in the corresponding search space (with A being the default value). b and c: Algorithm HPs. Each boxplot represents 50 repetitions. The solid and dashed lines indicate the range of the considered search space and the default value, respectively.

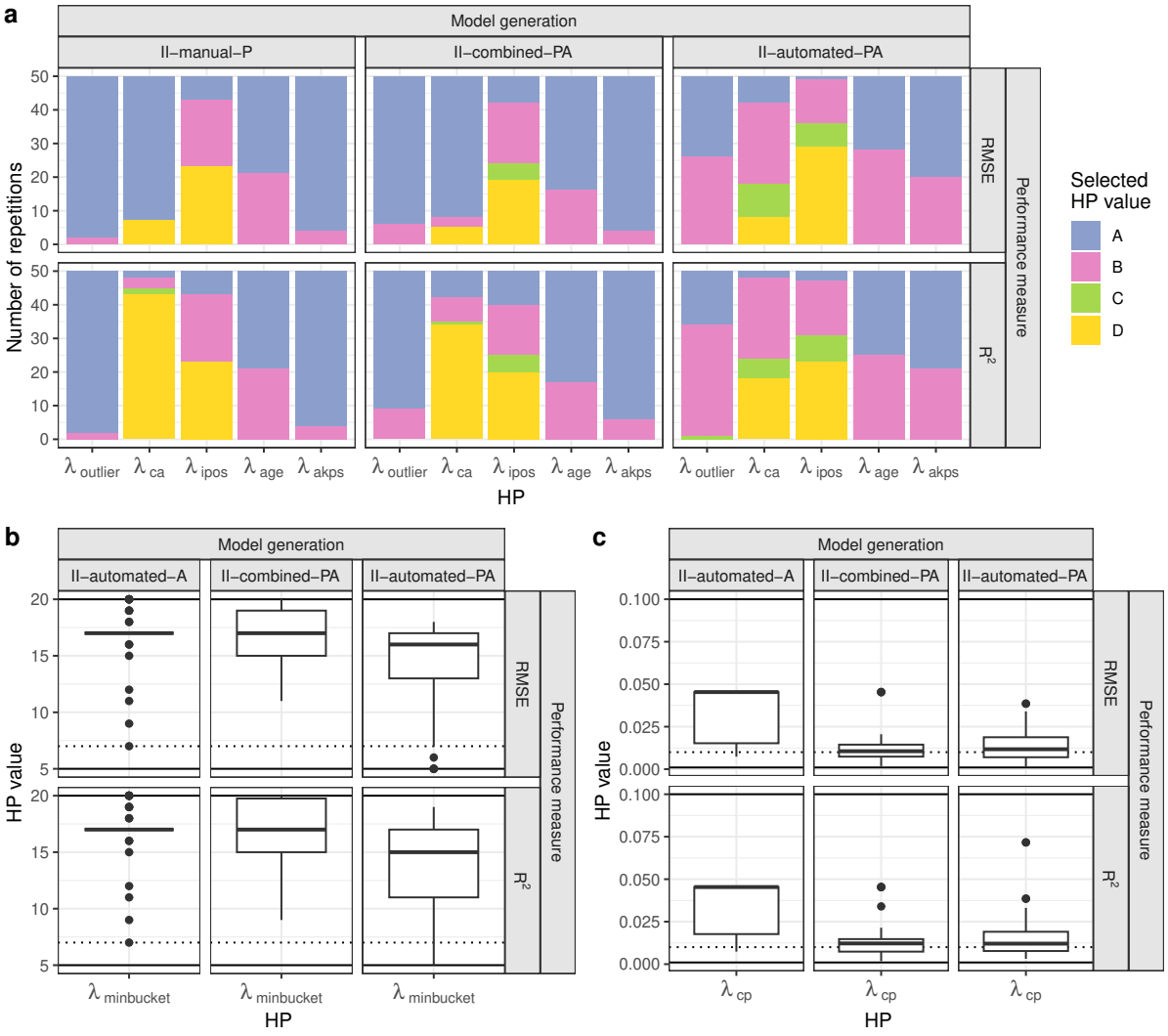

Figure S4: Selected HPs for the analysis settings where CART is used as the learning algorithm and $n_{\text{train}} = 724$. Only model generation procedures that involve tuning the corresponding HP type are shown. a: Preprocessing HPs. The labels A, B, C, and D correspond to the first, second, and, if present, subsequent values in the corresponding search space (with A being the default value). b and c: Algorithm HPs. Each boxplot represents 50 repetitions. The solid and dashed lines indicate the range of the considered search space and the default value, respectively.

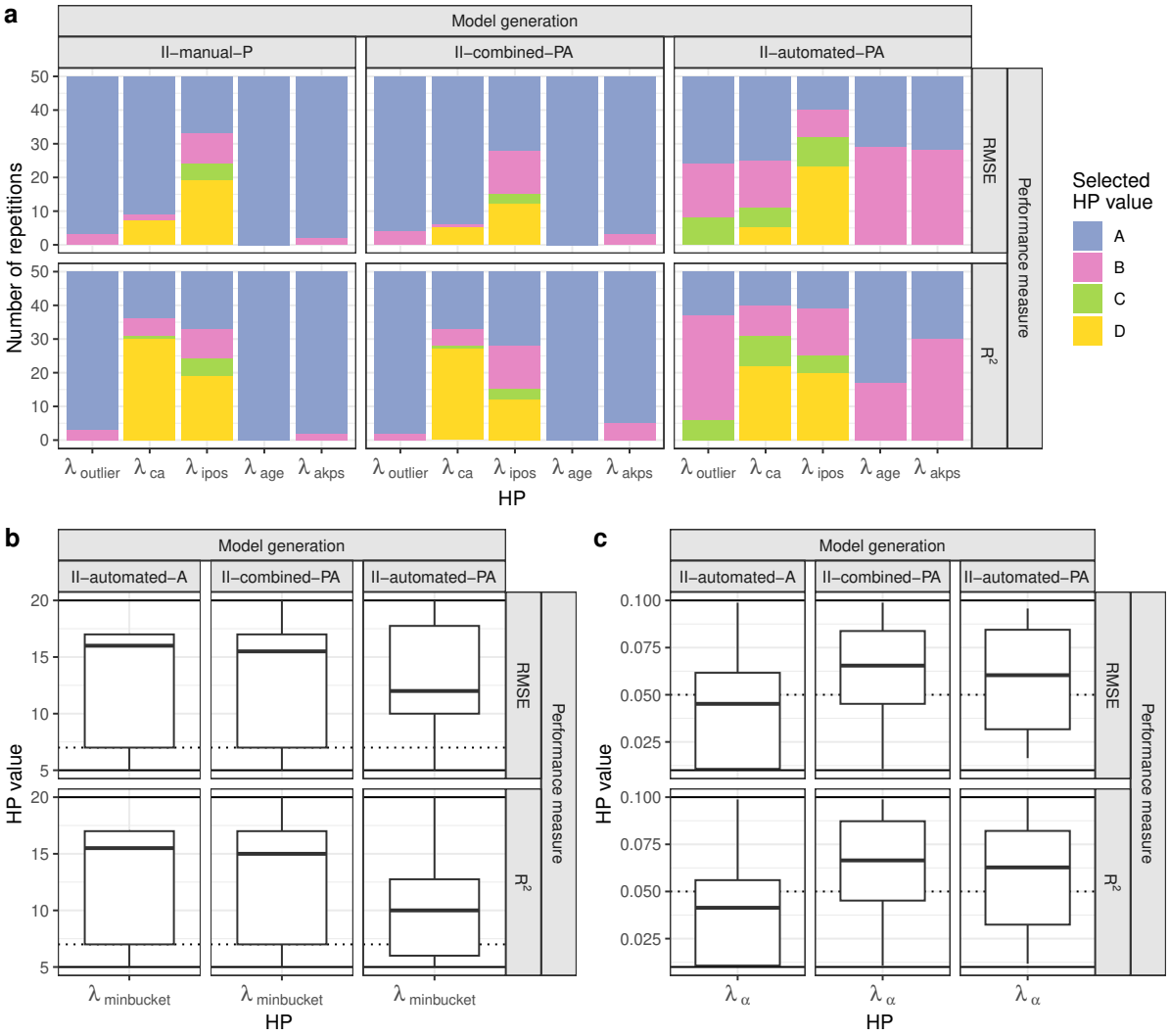

Figure S5: Selected HPs for the analysis settings where CIT is used as the learning algorithm and $n_{\text{train}} = 362$. Only model generation procedures that involve tuning the corresponding HP type are shown. a: Preprocessing HPs. The labels A, B, C, and D correspond to the first, second, and, if present, subsequent values in the corresponding search space (with A being the default value). b and c: Algorithm HPs. Each boxplot represents 50 repetitions. The solid and dashed lines indicate the range of the considered search space and the default value, respectively.

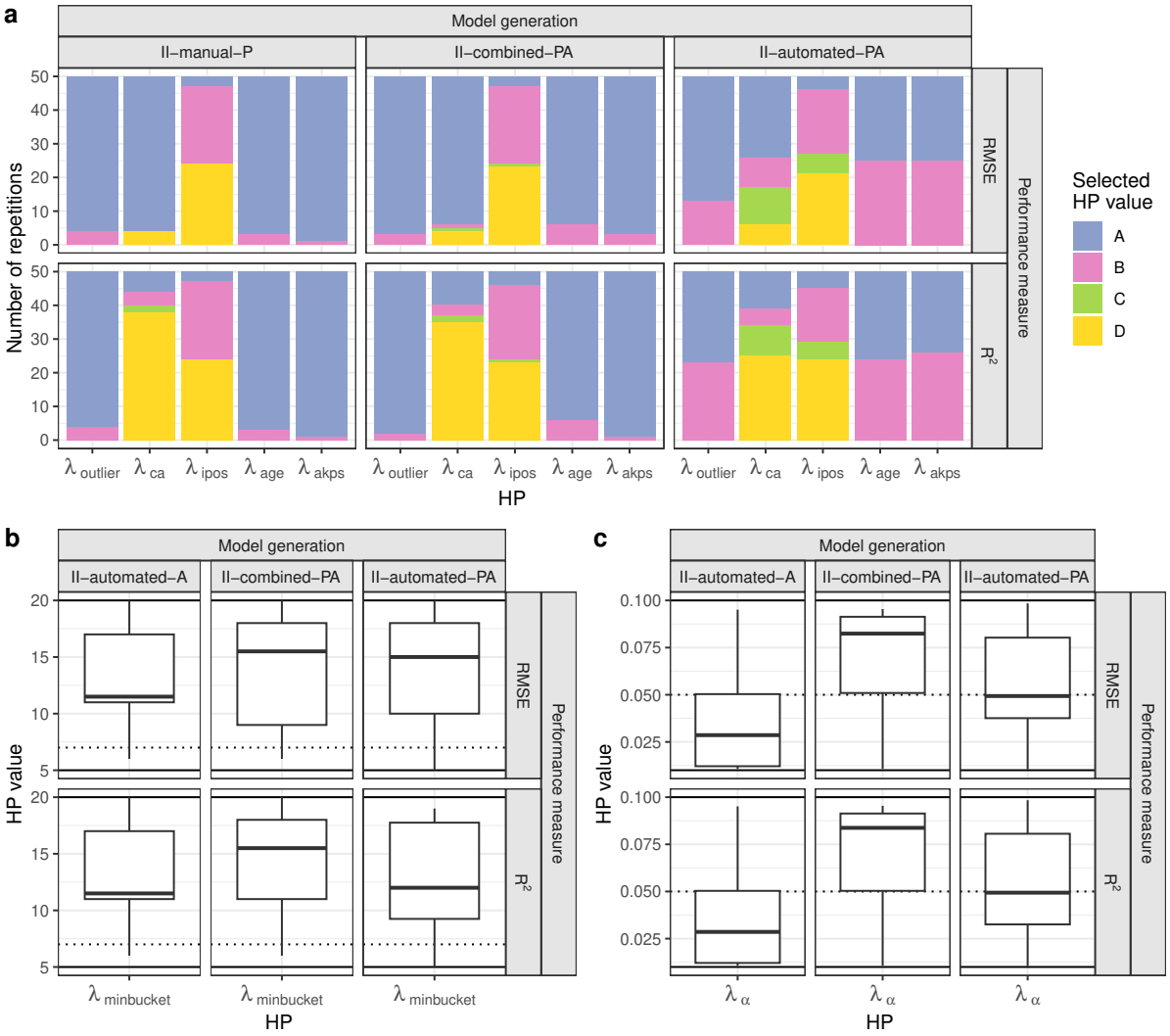

Figure S6: Selected HPs for the analysis settings where CIT is used as the learning algorithm and $n_{\text{train}} = 724$. Only model generation procedures that involve tuning the corresponding HP type are shown. a: Preprocessing HPs. The labels A, B, C, and D correspond to the first, second, and, if present, subsequent values in the corresponding search space (with A being the default value). b and c: Algorithm HPs. Each boxplot represents 50 repetitions. The solid and dashed lines indicate the range of the considered search space and the default value, respectively.

B.5 Clustering structure

In Figure 5 (Section 5.3), which presents the prediction error differences for 96 analysis settings, it can be seen that the CV error unexpectedly exhibits an optimistic bias in settings without HP tuning. The same observation applies to the nested CV error in analysis settings with HP tuning. These results can be attributed to the clustering structure of the COMPANION data set, and we will explain this in more detail below. Specifically, we will describe the clustering structure (Supplementary Section B.5.1), explain how it impacts the estimated prediction errors (Supplementary Section B.5.2), discuss why the experimental setup was not adapted to account for this clustering (Supplementary Section B.5.3), and present an additional extension of the experimental setup with respect to clustering (Supplementary Section B.5.4).

B.5.1 Clustering in the COMPANION data set

The COMPANION data set exhibits a nested clustering structure. At the first level, clustering arises because several palliative care phases may originate from the same episode of care of a patient. An episode of care is defined as the period between admission to a specific specialist palliative care setting and the termination of care in that same setting. At the second level, clustering occurs because the episodes of care in the data were collected from different palliative care teams. Episodes within the same team are typically more similar to one another than to episodes from different teams. Since no episode of care is associated with more than one palliative care team, the clustering follows a nested structure.

As a result, the 1,449 palliative care phases reported for the COMPANION data set in Section 5.1 originate from 705 episodes of care, which in turn are collected from 9 specialist palliative home care teams. A more detailed depiction of this nested clustering structure is provided in Figure S7.

B.5.2 Impact on prediction error estimates

While our empirical illustration and the paper as a whole focus on overlap-induced data leakage, the clustering structure of the COMPANION data set introduces another form of leakage that generally occurs when the assumption of independent and identically distributed (i.i.d.) observations is violated and the violation is not accounted for during model evaluation. This type of leakage is briefly mentioned in Section 2.4.2 of the main paper and described in more detail in Supplementary Section A.1. As a result, the prediction error estimates can be optimistically biased, even in the absence of overlap-induced data leakage. We now explain where the clustering is not accounted for in the experimental setup and how this affects the estimated prediction errors and their differences.

First, the clustering structure is ignored when splitting the COMPANION data set into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{new}}$, as the split is performed at the phase level rather than at the episode or team level. Consequently, if the prediction model is intended to be applied to new episodes and teams not present in the COMPANION data set, $\widehat{\text{PE}}_{\text{new}}$ is optimistically biased, as it has an unfair advantage compared to other data sets with new episodes and teams. A more precise statement in

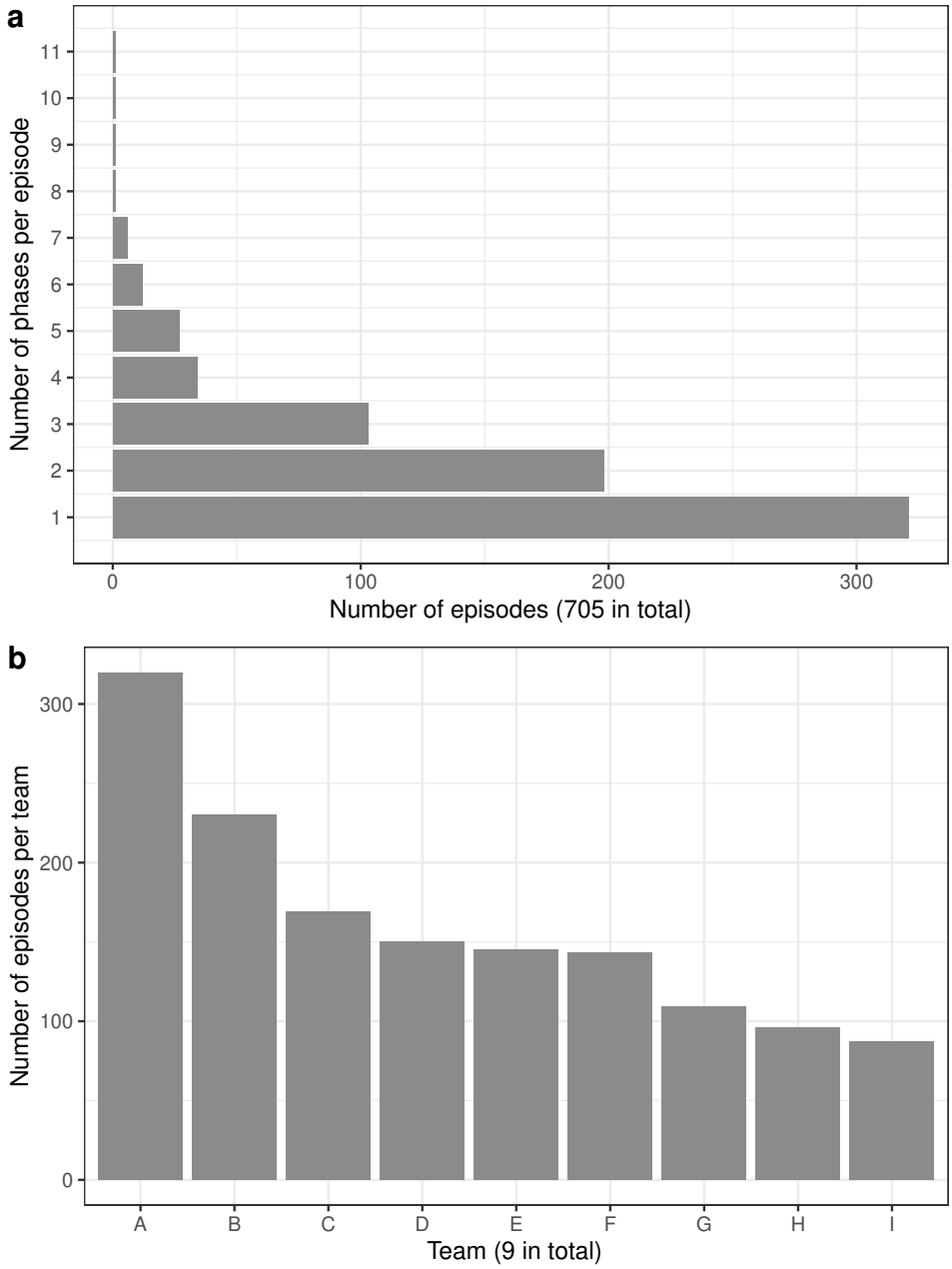

Figure S7: Overview of the nested clustering structure in the COMPANION data set. The x-axis represents the clusters, and the y-axis indicates the cluster size. a: Phases within episodes (first-level clustering). b: Episodes within teams (second-level clustering). The labeling of the teams (A, B, C, etc.) is specific to this plot and reflects the teams' ordering based on the number of episodes, with 'A' representing the team with the most episodes.

step (iii) in Section 5.2.1 would thus be that $\widehat{\text{PE}}_{\text{new}}$ is unbiased except for a potential optimistic bias caused by clustering-induced data leakage. Second, if $\widehat{\text{PE}}_{\text{train}}$ is estimated via simple or nested CV, the clustering structure is also ignored when creating the CV splits. Accordingly, as with $\widehat{\text{PE}}_{\text{new}}$, this leads to an optimistic bias in $\widehat{\text{PE}}_{\text{train}}$ due to data leakage induced by clustering (although in contrast to $\widehat{\text{PE}}_{\text{new}}$, $\widehat{\text{PE}}_{\text{train}}$ may also be affected by other biases). Note that for

nested CV, it is only the ignoring of the clustering in the outer CV loop that results in the optimistic bias, as the inner splits are only used for tuning.

For the difference between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$, which is the focus of our illustration, this has two key implications: If $\widehat{\text{PE}}_{\text{train}}$ results from an analysis setting where the apparent error was used to evaluate the final prediction model, the difference between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$ may underestimate the optimistic bias that would arise if $\mathcal{D}_{\text{new}}$ contained exclusively observations from new episodes and teams not present in $\mathcal{D}_{\text{train}}$. If $\widehat{\text{PE}}_{\text{train}}$ corresponds to the simple or nested CV error, the clustering-induced optimistic bias would, under the assumption that $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$ are subject to the same level of bias, effectively cancel out when considering the difference between $\widehat{\text{PE}}_{\text{train}}$ and $\widehat{\text{PE}}_{\text{new}}$. However, as shown in Figure 5, this is not the case. Further analysis (not shown) reveals that the observed differences arise from the slightly higher proportion of patient episodes present in both $\mathcal{D}'_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ during resampling, compared to the proportion of episodes present in both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{new}}$ during the initial split. As a result, $\widehat{\text{PE}}_{\text{train}}$ is affected by a larger optimistic bias than $\widehat{\text{PE}}_{\text{new}}$, which manifests in Figure 5, where their difference is examined.

B.5.3 Splits on cluster level

To prevent data leakage due to clustering, both the initial split into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{new}}$, as well as any resampling method applied to $\mathcal{D}_{\text{train}}$, must be performed at the team level. With a total of 9 teams, this means that in each repetition of every analysis setting, $\mathcal{D}_{\text{train}}$ consists of either 4 or 5 teams. Furthermore, when performing CV on $\mathcal{D}_{\text{train}}$ at the team level, it is not possible to create 10 folds. Instead, each team forms a fold, and CV is carried out in a leave-one-out manner. Figure S8 presents the resulting prediction error differences for all analysis settings where no HPs are tuned, alongside the corresponding results from the original setup with naive splits (i.e. splits that ignore clustering) for comparison. First, it can be observed that if $\widehat{\text{PE}}_{\text{train}}$ corresponds to the CV error, the differences are smaller than or equal to zero for RMSE. This confirms that the optimistic bias found for the CV error in the corresponding naive setup is caused by the clustering structure of the data. However, Figure S8b also reveals that performing CV at the team level leads to highly variable prediction error differences, which is not surprising given the limited number of teams, each varying in the number of episodes and phases they contain. Since we argue that, under these circumstances, it is not reasonable to perform HP tuning, we decided to ignore the clustering structure in the setup of our main analysis. Additionally, in the interest of computational resources, we did not conduct the team-level analysis for the remaining analysis settings involving tuning. However, this should clearly not be taken as a standard for applications beyond illustrative purposes.

B.5.4 Learning algorithms for clustered data

In addition to performing splits at the cluster level, we also extended the main experimental setup by including additional learning algorithms specifically designed for clustered data. These are the Random Effects/Expectation-Maximization Tree algorithm (REEMT; `R` package `REEMtree`; Sela and Simonoff, 2011), and the Linear Mixed-Effects Model Tree algorithm

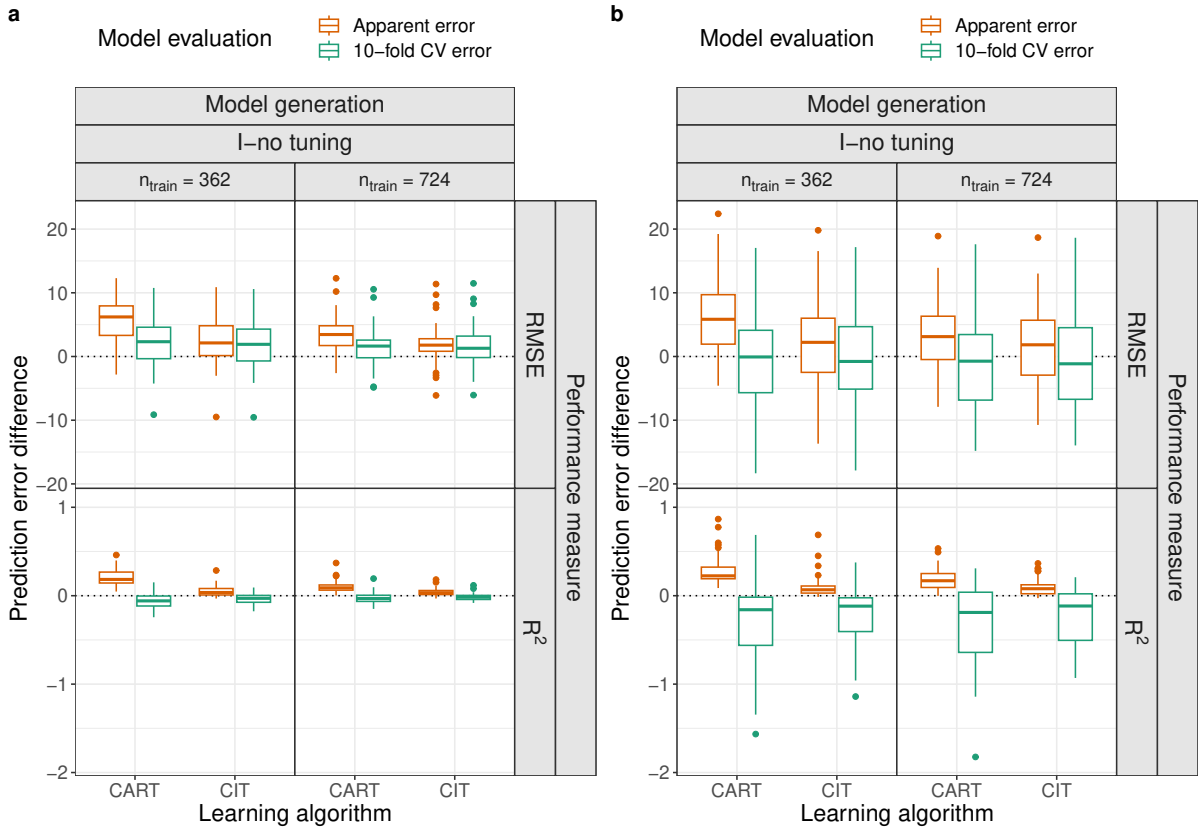

Figure S8: Comparison of prediction error differences when clustering is ignored vs. accounted for. Both subfigures present the prediction error differences for all considered analysis settings without HP tuning, with each boxplot summarizing 50 repetitions of a specific setting. The prediction error differences are calculated as $\widehat{\text{PE}}_{\text{new}} - \widehat{\text{PE}}_{\text{train}}$ for RMSE and $\widehat{\text{PE}}_{\text{train}} - \widehat{\text{PE}}_{\text{new}}$ for R^2. a: Naive setup, where clustering is ignored during splitting. Results are adapted from Figure 5, with extended y-axis limits. b: Cluster setup, where clustering is accounted for by performing splits at the team level.

(LMMT; `R` package `glmertree`; Fokkema et al., 2018). In the implementation used for our illustration, both algorithms take into account the clustering structure by iterating between two steps: (i) fitting a decision tree using the CART algorithm for REEMT or the CIT algorithm for LMMT and (ii) estimating random intercepts via a linear mixed model, which are subtracted from the outcome variable in the subsequent tree-fitting iteration. To ensure model stability, random effects are only included for each palliative care team, rather than for each individual episode, as more than 300 episodes consist of only a single palliative care phase (Figure S7a). Including REEMT and LMMT in the analysis, however, does not yield new insights. Their results closely resemble those of CART and CIT, as demonstrated in Figure S9, which compares the prediction error differences of the algorithms.

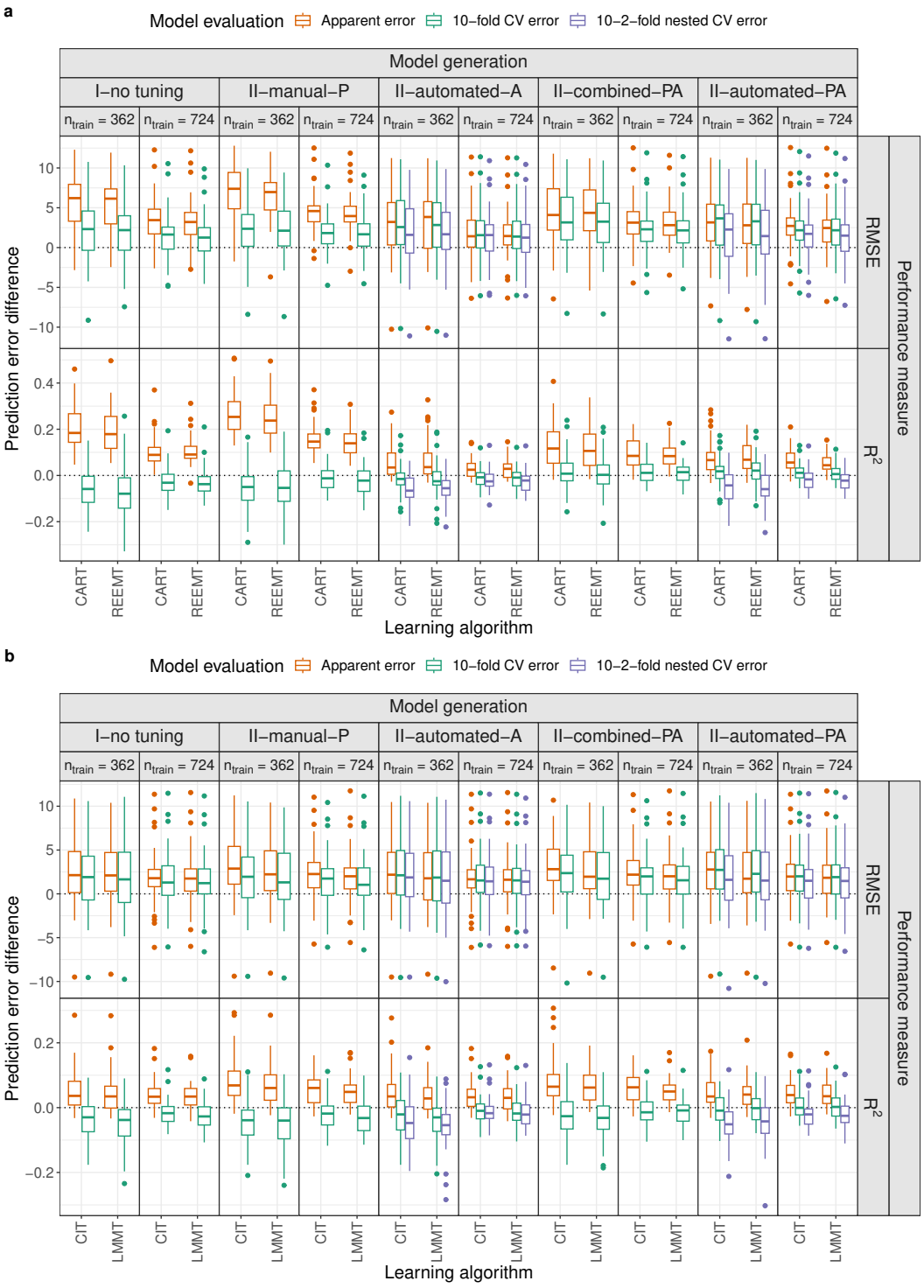

Figure S9: Comparison of prediction error differences between CART and CIT and their counterparts that include random intercepts, REEMT and LMMT, respectively. The same model generation and evaluation procedures, performance measures, and sample sizes as in the main setup are included. Each boxplot summarizes results from 50 repetitions of a specific setting. The prediction error differences are calculated as $\widehat{\text{PE}}_{\text{new}} - \widehat{\text{PE}}_{\text{train}}$ for RMSE and $\widehat{\text{PE}}_{\text{train}} - \widehat{\text{PE}}_{\text{new}}$ for R^2. a: CART vs. REEMT. b: CIT vs. LMMT.

B Contribution 2: "Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results"

This section is a reprint of:

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(2), e1441. https://doi.org/10.1002/widm.1441

Copyright:

Author contributions:

C. Nießl, M. Herrmann, and A.-L. Boulesteix conceptualized the paper, motivated by insights gained in a previous benchmark study led by M. Herrmann. C. Nießl led the development of the methodology, building on a pilot study by C. Wiedemann and incorporating expertise and insights on benchmark studies from M. Herrmann, G. Casalicchio, and A.-L. Boulesteix. C. Nießl wrote the R code for generating and analyzing the results, reusing selected elements of code from the pilot study by C. Wiedemann. The original draft of the manuscript was written by C. Nießl, with subsequent review and editing carried out by all authors.

Received: 3 June 2021 | Revised: 30 September 2021 | Accepted: 6 November 2021

DOI: 10.1002/widm.1441

FOCUS ARTICLE

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results

Christina Nießl[1] | **Moritz Herrmann**[2] | **Chiara Wiedemann**[1] | **Giuseppe Casalicchio**[2] | **Anne-Laure Boulesteix**[1]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig Maximilians University Munich, Munich, Germany

[2]Department of Statistics, Ludwig Maximilians University Munich, Munich, Germany

Correspondence
Christina Nießl, Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig Maximilians University Munich, Marchioninistr. 15, D-81377 Munich, Germany.
Email: cniessl@ibe.med.uni-muenchen.de

Funding information
This work was supported by the German Federal Ministry of Education and Research (01IS18036A) and by the German Research Foundation (BO3139/4-3, BO3139/7-1, BO3139/6-2) to ALB. The authors of this work take full responsibilities for its content.

Edited by: Mehmed Kantardzic, Associate Editor and Witold Pedrycz, Editor-in-Chief

Abstract

In recent years, the need for neutral benchmark studies that focus on the comparison of methods coming from computational sciences has been increasingly recognized by the scientific community. While general advice on the design and analysis of neutral benchmark studies can be found in recent literature, a certain flexibility always exists. This includes the choice of data sets and performance measures, the handling of missing performance values, and the way the performance values are aggregated over the data sets. As a consequence of this flexibility, researchers may be concerned about how their choices affect the results or, in the worst case, may be tempted to engage in questionable research practices (e.g., the selective reporting of results or the post hoc modification of design or analysis components) to fit their expectations. To raise awareness for this issue, we use an example benchmark study to illustrate how variable benchmark results can be when all possible combinations of a range of design and analysis options are considered. We then demonstrate how the impact of each choice on the results can be assessed using multidimensional unfolding. In conclusion, based on previous literature and on our illustrative example, we claim that the multiplicity of design and analysis options combined with questionable research practices lead to biased interpretations of benchmark results and to over-optimistic conclusions. This issue should be considered by computational researchers when designing and analyzing their benchmark studies and by the scientific community in general in an effort towards more reliable benchmark results.

This article is categorized under:
Technologies > Visualization
Technologies > Data Preprocessing
Technologies > Structure Discovery and Clustering

KEYWORDS

benchmarking, method comparison, over-optimistic results, questionable research practices, variability of results

1 | INTRODUCTION AND RELATED WORK

With the constant development of new methods in computational sciences (e.g., machine learning and bioinformatics), it is becoming increasingly difficult for data analysts to keep pace with scientific progress and to select the most appropriate method for their data and research question out of the many existing approaches. This problem is addressed by benchmark studies, which systematically analyze and compare the performance of several methods in different conditions using simulated or real data sets.

In many cases, benchmark studies are performed as part of a paper introducing a new method, usually with the intention to demonstrate the superiority of the new method over existing ones. Accordingly, they can be considered as biased in favor of the newly proposed method and should be seen as an informal method comparison rather than a real benchmark study (Boulesteix et al., 2013; Buchka et al., 2021; Norel et al., 2011). In contrast, so-called *neutral* benchmark studies are defined as benchmark studies that focus on the comparison itself and are ideally performed by reasonably neutral authors, that is, authors who (1) are equally experienced with all considered methods and (2) design and analyze the study in a rational way (Boulesteix et al., 2017). These characteristics make neutral benchmark studies essentially unbiased. Therefore, recommendations resulting from such studies are especially relevant both for method users and developers (Boulesteix et al., 2018).

Regarding the appropriate design and analysis of benchmark studies, the available literature ranges from general guidelines (Boulesteix, 2015; Weber et al., 2019) and statistical frameworks (Boulesteix et al., 2015; Demšar, 2006; Eugster et al., 2012; Hothorn et al., 2005, all with focus on supervised learning), to recommendations for context-specific benchmarks (e.g., Bokulich et al., 2020; Kreutz, 2019; Mangul et al., 2019; Zimmermann, 2020). However, for many issues relevant in practice (e.g., the selection of data sets and performance measures), no concrete guidance or methodology can be found. This means that researchers are usually faced with a high amount of flexibility when conducting their benchmark study.

As a consequence, researchers who are aware of these issues, although making well-considered design and analysis choices prior to conducting the benchmark study, might be concerned about how their choices affect the results. On the other hand, the high amount of flexibility could tempt less aware researchers to engage in questionable research practices (see John et al., 2012, in the context of applied research) when conducting their benchmark study. This includes the selective reporting of results (e.g., reporting the results of only one performance measure although performance was originally assessed by two measures) and the modification of specific design and/or analysis components of the benchmark study after seeing the results (e.g., using performance measures other than those originally selected). Of course, these practices are not questionable on their own. For example, it is fine to use an alternative performance measure if the current one does not produce meaningful results as long as the change of performance measure is adequately justified and documented. However, practices such as the selective reporting of results or the post hoc modification of benchmark components do become questionable if they are applied to fit the researchers' expectations or hopes. For example, researchers might seek an "exciting" result (e.g., a clear-cut result suggesting a univocal winner as opposed to vague tendencies) or have a specific presumption in mind that they want to be confirmed by the results (e.g., the superiority of a certain method or class of methods that they are more familiar with or that has performed well in previous benchmark studies).

The problem with such research practices is that they are likely to produce over-optimistic results, that is, results with an optimistic bias towards the researchers' expectations and hopes. While we are convinced that very few researchers have the actual intention to cheat (Ioannidis et al., 2014), it should not be understated that "even an honest person is a master of self-deception" (Nuzzo, 2015), meaning that every researcher is at risk of engaging in questionable research practices. Moreover, the non-neutrality that leads to such practices in the first place is difficult to avoid completely and is likely to arise in a subconscious manner even in studies intended as neutral. Note also that the actual neutrality of neutral benchmark studies can only be checked to a certain extent. For example, one may review the authors' publication lists to identify the methods they are most familiar with, but this gives only a partial picture of someone's (non-)neutrality.

In application fields of statistics (e.g., medicine and psychology), the multiplicity of analysis strategies and the associated risk of over-optimistic results are well-known issues (Hoffmann et al., 2021; Ioannidis, 2005; Simmons et al., 2011) and terms such as "p-hacking" or "fishing expeditions" have been discussed by many (Head et al., 2015; Wagenmakers et al., 2012). However, in methodological research including benchmark studies, this topic is covered rather sparsely. Existing literature on the risk and prevention of over-optimism in benchmark studies is either limited to general considerations in benchmarking guidelines (Boulesteix et al., 2017; Weber et al., 2019) or to benchmark studies that are performed as part of a paper introducing a new method (Boulesteix, 2015; Norel et al., 2011), which can be transferred to neutral benchmark studies only to a limited extent. Similarly, the scarce literature that *empirically* investigates the effects of over-optimism in benchmark studies in a quantitative manner is either also devoted to the bias affecting evaluations of a newly proposed method to other existing methods (Buchka et al., 2021; Jelizarow et al., 2010), or focusing on the selection of data sets (MacIà et al., 2013; Yousefi et al., 2010).

In this paper, we illustrate and discuss the multiplicity of options regarding the design and analysis of neutral benchmark studies based on real data sets, and examine its effect on the results. Note that although we focus on neutral benchmark studies based on real data, our results are also relevant to benchmarks comparing new to existing methods and, to some extent, benchmarks based on simulated data. We will empirically address the multiplicity of options and its effects in a twofold approach. In the first step, in order to raise awareness of the multiplicity of possible results and the over-optimism that may arise from questionable research practices, we use the results of a recently published benchmark study to illustrate how variable the resulting method rankings are when different options for design and analysis are considered. In the second step, we propose a framework based on multidimensional unfolding (Borg & Groenen, 2005) that enables researchers to assess the impact of each choice on the method rankings. More precisely, the framework allows to analyze when and how using alternative options for a specific choice affects the results and can thus be an effective strategy to prevent biased interpretations and over-optimistic conclusions.

The exemplary study we will use throughout the paper to illustrate our proposed framework and the multiplicity of possible options and results is a benchmark experiment by Herrmann et al. (2021) comparing the performance of 13 survival prediction methods based on 18 real so-called multi-omics" data sets. Note that our paper does not intend to question the results of this study. Instead, it should be seen as extended analysis of the benchmark study, which by assessing the multiplicity of results and examining the impact of each choice, makes the results of Herrmann et al. (2021) even more reliable and meaningful.

While the framework proposed in this paper can be utilized by all researchers who conduct benchmark studies of computational methods (e.g., in the fields of machine learning, data mining, statistics, etc.), the illustrated multiplicity of results should ideally also raise awareness among the readers of such studies. The concepts and results presented in this paper may therefore be useful for method developers and methodological researchers as well as applied researchers and data analysts.

The remainder of this paper is structured as follows: we review and discuss a selection of design and analysis choices in the context of benchmark studies in Section 2, and describe the design of the study as well as the principle of multidimensional unfolding in Section 3. The results are presented in Section 4, which is followed by a discussion in Section 5 and concluding remarks in Section 6.

2 | EXAMPLES OF DESIGN AND ANALYSIS CHOICES IN BENCHMARK STUDIES

2.1 | Setting

In this section, we discuss some of the choices that researchers are faced with when conducting a benchmark study based on real data sets. In general, most choices that have to be made to conduct a benchmark study relate to (1) the general aim of the study, (2) the design of the study, or (3) the analysis of the performance results; see the left part of Figure 1. Choices that belong to the first category are, for instance, the choice of methods to be compared or the type of outcome variable to be considered. However, in this paper, we focus on choices regarding the design of the study (i.e., how the aim of the study is addressed) and the analysis of performance results (i.e., how the $L \times M$ matrix of results generated by each considered performance measure is analyzed, where L and M are the numbers of data sets and methods, respectively). It is important to note that these choices should ideally be made prior to conducting the benchmark study. However, we conjecture that they are in practice often made post hoc, that is, after seeing the

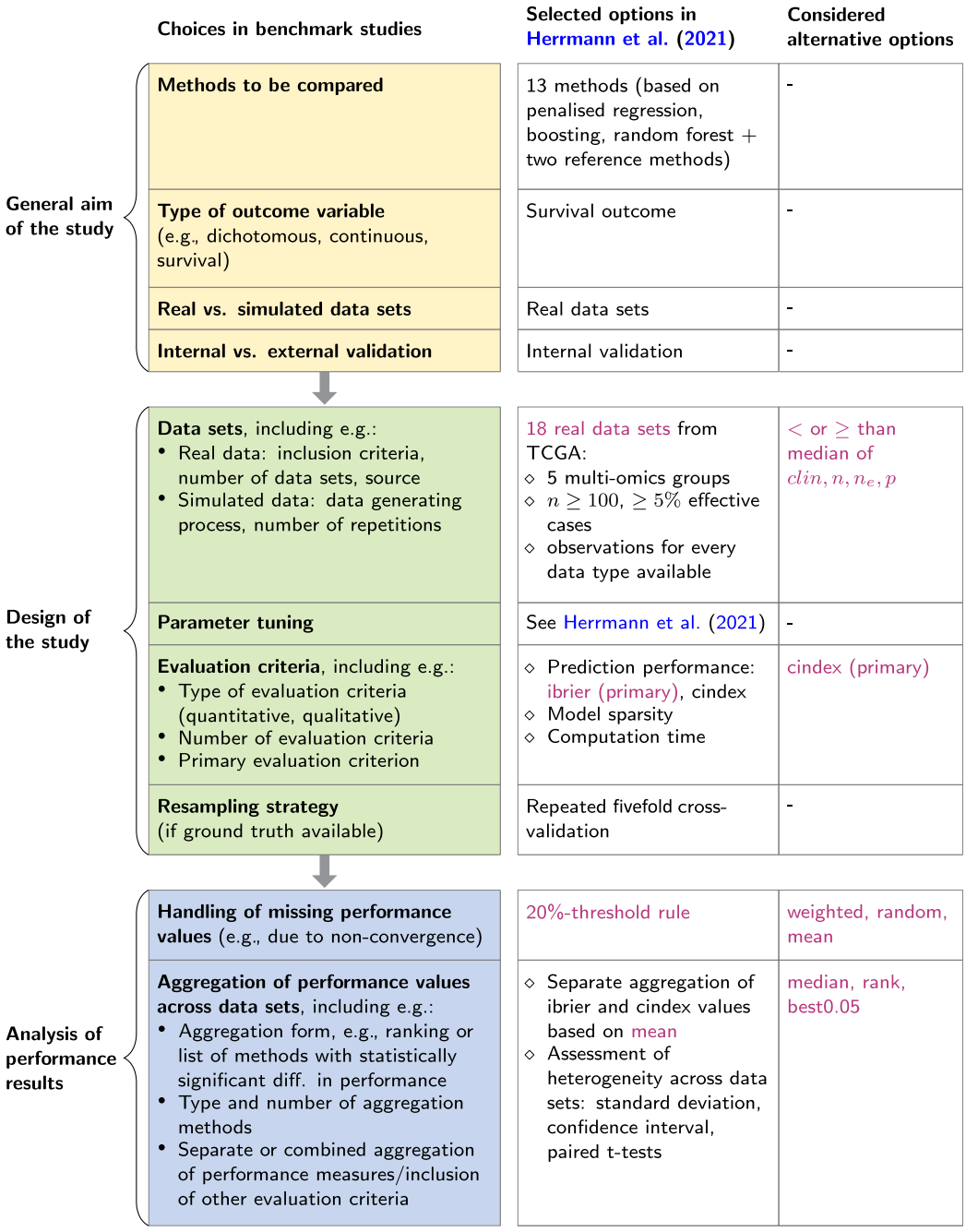

FIGURE 1 Examples of choices that researchers are usually faced with when conducting a benchmark study including options used in the example benchmark study by Herrmann et al. (2021) (second column) and alternative options (third column). Options that are considered in our illustration are colored in pink

results—which can amount to questionable research practices. When reading a benchmark study, there is no way to check when the choices were made.

For each choice, we will give concrete examples of possible options that will later be analyzed with regard to their effect on the results; see the right part of Figure 1. For this purpose, we consider the benchmark study by Herrmann et al. (2021) mentioned above. The authors compare the performance of $M = 13$ survival prediction methods (here denoted as *BlockForest*, *Clinical Only*, *CoxBoost*, *CoxBoost Favoring*, *Glmboost*, *Grridge*, *Ipflasso*, *Kaplan–Meier*, *Lasso*, *Prioritylasso*, *Prioritylasso Favoring*, *Ranger* and *Rfsrc*) on $L = 18$ real multi-omics data sets. See the original paper (Herrmann et al., 2021) for details on the methods, the benchmark experiment, and the results. We selected this study as an example because some of the authors of the present paper were also involved in conducting the benchmark study

by Herrmann et al. (2021). We therefore had first hand insight about the issues Herrmann et al. (2021) faced while designing and analyzing the benchmark study, which we believe to be reasonably representative of the important challenges encountered in most benchmark studies, as we will discuss in the remainder of this section.

2.2 | Design choices

2.2.1 | Data sets

The selection of data sets is an important design choice in every benchmark study, as the performances are usually highly variable across data sets (Novianti et al., 2015; Weber et al., 2019). To make meaningful statements and prevent the study from being underpowered, it is recommended to consider an adequate number of data sets (Boulesteix et al., 2017). Although there are suggestions on how to calculate the minimum required number (Boulesteix et al., 2015), it seems that the number of included data sets is usually based on practical criteria (such as availability or computational cost) rather than statistical considerations (MacIà et al., 2013). Moreover, if the benchmark study aims at external validation, the number of data sets that can be included in the benchmark study is usually limited, as for many data sets there is often no comparable data set available that could be used for external validation.

Concerning the type of data sets, researchers should include data sets that are representative for the domain of interest and diverse enough to make sure the methods can be evaluated under a wide range of conditions (Gatto et al., 2016; Weber et al., 2019). Corresponding inclusion criteria for the data sets should be defined before conducting the benchmark study (Boulesteix et al., 2017). However, the decision on how the inclusion criteria are defined lies with the researcher. In many benchmark studies, the exact search strategy or inclusion criteria are not reported transparently, suggesting that in these cases, there might be no clearly defined inclusion criteria at all.

In the benchmark study by Herrmann et al. (2021), the authors selected all cancer data sets with five different multi-omics groups and more than 100 samples from the TCGA research network (http://cancergenome.nih.gov). Additionally, they excluded data sets that did not have observations for every data type or less than 5% effective cases (i.e., patients with event), resulting in a total of $L = 18$ data sets. However, depending on their research interest, Herrmann et al. (2021) could have set additional constraints. For example, if the authors had been interested in the performance of the methods on data sets with a small number of effective cases, they could have adjusted the inclusion criteria accordingly (e.g., set $n_e < 30$). The other way around, one may decide to ignore data sets with a small number of events (e.g., set $n_e \geq 30$) because it is questionable if it makes sense to fit models in this case at all.

In this paper, we will address the multiplicity of possible options regarding the selection of data sets and its impact on the results by considering subgroups of the original $L = 18$ data sets defined based on some of the data sets' characteristics. The considered characteristics are the number of clinical variables (*clin*), the number of observations (n), the number of effective observations (n_e), and the number of variables (p). For each data set characteristic, we will only consider data sets that are smaller ($<$) or greater or equal ($\geq$) than the median value of the respective data set characteristic over the 18 considered data sets. This results in eight groups with 8–10 data sets.

2.2.2 | Quantitative performance measure

Another important aspect of benchmarking is the choice of evaluation criteria, which usually includes both quantitative performance measures and other measures such as runtime or qualitative features such as user-friendliness. Although all these evaluation criteria are important, we will focus on quantitative performance measures in this paper.

The choice of performance measure is usually context-specific, that is, it depends on the type of methods and data addressed in the benchmark study, as well as on the aspects of performance that are considered the most important by the researcher (Morris et al., 2019; Weber et al., 2019). It is also often a nontrivial choice. For some tasks such as classification, researchers are spoilt for choice considering the variety of measures they can choose from (e.g., accuracy, sensitivity/specificity, area under the curve or F1-score), which makes decisions difficult (Mangul et al., 2019; Robinson & Vitek, 2019). In contrast, for more complex situations they might have to design their own performance measures, which can also be challenging (Weber et al., 2019). To provide a more complete picture of the methods' behavior and avoid over-optimism, it can be useful to consider more than one performance measure (Norel et al., 2011). However,

there is no way to objectively determine the adequate number of performance measures as this is highly context dependent.

In the benchmark study by Herrmann et al. (2021), the primary performance measure is the integrated Brier score (Graf et al., 1999; denoted as *ibrier*). Additionally, they consider Uno's *C*-index (Uno et al., 2011; denoted as *cindex*). The authors justify their decision to use the ibrier as primary measure by the fact that cindex only measures the discriminatory power and is not a strictly proper scoring rule (Blanche et al., 2019), while the ibrier additionally measures calibration. However, they argue that if the main interest lies in *ranking* patients according to their risk, then the cindex would also be a valid measure. Furthermore, they reason that it makes sense to include the cindex for the purpose of comparability with other studies, since it is a widely used performance measure. Accordingly, depending on which aspect of performance they would have considered more important, Herrmann et al. (2021) could have also used the cindex as primary performance measure or only selected one of the two performance measures. In this paper, we will thus compare the results of ibrier and cindex.

2.3 | Analysis choices

2.3.1 | Handling of missing performance values

Because of non-convergence or other computational issues, methods sometimes fail to output a result for a specific data set. In the context of resampling procedures such as cross-validation or bootstrapping, the consequence is that performance values may be missing for all or part of the resampling iterations for some data sets. This problem seems to be common especially in benchmarks of larger scale (Bischl et al., 2013). While there is at least some literature devoted to the selection of data sets and performance measures, the issue of missing performance values in some combinations of data sets and methods is almost completely ignored. Many authors of benchmark studies do not report how they handled missing performance values, and there is to our knowledge no corresponding guidance available.

Bischl et al. (2013) mention several possible ad hoc options that could be applied if the missing values occur only on a subset of resampling iterations, namely that missing values could be imputed by the worst possible value or by the mean of the remaining performance values obtained for this combination of data set and method—although both options are not ideal in their opinion. Another ad hoc option they actually use for their benchmark study is a mixed strategy, where the imputed value is sampled from an estimated normal distribution of the remaining values if the method fails in less than 20% of the resampling iterations. If the method fails in more than 20% of the resampling iterations, the worst possible value is used for imputation. Herrmann et al. (2021), who use cross-validation as resampling procedure and also face the problem of failing iterations, use a similar 20%-threshold rule as Bischl et al. (2013). However, instead of sampling from a normal distribution, they use the mean performance value of the remaining iterations and instead of the worst possible value, they assign values of the performance measures corresponding to random prediction (i.e., 0.25 for ibrier and 0.5 for cindex).

Since there seems to be no common agreement on how to handle missing values in this context, other sensible options would also be justifiable. For example, missing values could be imputed by a formula that weights the mean performance value and the random performance value used by Herrmann et al. (2021) according to the proportion of missing values, thus avoiding the choice of an arbitrary threshold. For the ibrier, where 0 corresponds to the best possible value and 0.25 to random prediction, the imputed value for the considered combination of data set and method could be defined as

$$x_{\text{impute}} = 0.25 - \left(0.25 - \frac{\sum_{i \in \mathcal{I}} x_i}{|\mathcal{I}|}\right) \cdot (1 - r), \tag{1}$$

where $\mathcal{I}$ is the set of indices of the non-failed iterations, x_i is the ibrier value for iteration $i \in \mathcal{I}$, and r is the proportion of missing values. For two methods with the same mean value for non-failed iterations, the method with more missing values obtains a worse performance value. Moreover, the imputed value is equal to 0.25 if a method has 100% failures for a data set, or a mean value greater or equal than 0.25 (which makes sense since fluctuations above the value 0.25 corresponding to random prediction are not relevant). Another advantage of this weighted imputation procedure is that

it reduces to the mean when the proportion of missing values r tends to 0—as intuitively expected. The corresponding formula for the cindex can be found in the Supplementary material.

In this paper, we will consider four imputation methods that can be used to handle the issue of missing performance values: the 20%-threshold rule used by Herrmann et al. (2021), the weighted method in Equation (1), imputation using values that correspond to random prediction, and imputation using the average of the non-failed iterations.

2.3.2 | Aggregation of performance values across data sets

Although it is common to analyze the methods' individual performances across data sets (e.g., using graphical tools), most benchmark studies ultimately aggregate the performance values over the data sets to generate an overall method evaluation. This is done, for example, in the form of a ranking (often taking not only the rank order into account, but also the aggregated performance values that generate these ranks) or a list of methods that show statistically significant differences in performance. While there is much literature addressing statistical testing procedures in benchmark experiments based on a single data set (Dietterich, 1998; Hothorn et al., 2005) or several data sets (Demšar, 2006; Eisinga et al., 2017), there seems to be no consensus on how to generate an overall method *ranking* from several data sets, which we will focus on in this section.

For example, the performance values can be aggregated using standard summary measures such as the mean, median, minimum, maximum, or standard deviation (Mersmann et al., 2015). Since the distribution of performance values can be considerably skewed, some authors advise against using the mean or median as aggregation method. Instead, they recommend assigning ranks to the methods for each data set such that the best method in the corresponding data set obtains rank 1 and the worst method rank M, where M is the number of considered methods (Demšar, 2006; Hornik & Meyer, 2007). The resulting ranks are then usually aggregated using the mean (e.g., Kibekbaev & Duman, 2016; Verenich et al., 2019) or, less often, the median (e.g., Orzechowski et al., 2018).

Other possible aggregation methods include counting the number of times a method performs best, often divided by the number of data sets to obtain a value between 0 and 1 (e.g., De Cnudde et al., 2020; Fernández-Delgado et al., 2014; Wu et al., 2020). Some of these authors suggest to not only consider the best performing method for each data set but also the set of methods performing similarly to the best method. Accordingly, Fernández-Delgado et al. (2014) consider the number of data sets in which a method achieves 95% or more of the maximum accuracy (i.e., the accuracy achieved by the best performing method in that data set) divided by the total number of data sets. In the same vein, Wu et al. (2020) estimate the probability of achieving good performance as the number of data sets for which the method is among the top three methods divided by the total number of data sets.

Note that all aggregation methods presented so far are based on point estimates of the methods' performances. Although less frequently used in practice, it is also possible to generate method rankings based on the results of statistical tests (i.e., pairwise comparisons indicating if Method 1 performs significantly better than Method 2) using consensus rankings (Hornik & Meyer, 2007).

If more than one performance measure and/or other evaluation criteria (e.g., runtime) are considered, researchers also have to decide if rankings arising from multiple criteria should be combined in some form (e.g., Eugster et al., 2012) or should be considered separately, as suggested by Weber et al. (2019). Specifically, Weber et al. (2019) recommend to identify a set of consistently high performing methods based on the individual rankings and then highlight the different strengths of each method.

Herrmann et al. (2021) aggregate the performance values based on ibrier and cindex using the mean and consider each ranking separately. To assess the heterogeneity of performances across data sets, they also calculate the resulting standard deviations and confidence intervals and perform paired t-tests. In our illustration, we will consider four aggregation methods that can be used to generate method rankings: mean (as used by Herrmann et al., 2021), median, mean rank, and number of times a method performs best. If two methods obtain the same rank according to the number of times they perform best, they are additionally ranked by the number of times their performance lies within the 5% environment of the best performing method. This applies if $\frac{|\bar{x}_m - \bar{x}_{best}|}{\bar{x}_{best}} < 0.05$, where $\bar{x}_m$ denotes the performance (cindex or ibrier) of method m and $\bar{x}_{best}$ the performance of the best performing method in the corresponding data set. We denote this aggregation method (i.e., counting the number of times a method performs best and the number of times it lies within the 5% environment as secondary ranking method) as *best0.05*.

Note that we will focus on the ranks resulting from each aggregation method instead of the aggregated performance values that generate these ranks since the four aggregation methods have different scales (cindex/ibrier for mean and

median, mean ranks for mean rank and counts for best0.05), which would require appropriate normalization to compare them in a meaningful way. While this normalization would be specific to the type of considered evaluation criteria and aggregation methods, ranks can be generated in almost every benchmark study, which is why they are used in this illustrative example. Moreover, since we only evaluate the results of one performance measure at a time (ibrier or cindex), we are not considering different options for combining rankings that result from more than one performance measure.

3 | METHODS

3.1 | Design of the study

To illustrate the variability of benchmark results with respect to design and analysis choices, we use the benchmark results from Herrmann et al. (2021) and systematically examine different combinations of design and analysis options. Specifically, we consider all combinations of options regarding the choice of data sets (9 options), performance measure (2 options), imputation method (4 options), and aggregation method (4 options) described in Section 2 and Figure 1. This results in $9 \times 2 \times 4 \times 4 = 288$ combinations. We then compare the 288 resulting rankings of the 13 survival prediction methods, where a rank of 1 corresponds to the best performing method and a rank of 13 to the worst performing method (average ranks are assigned in case of ties).

3.2 | Multidimensional unfolding

The impact of each choice on the method rankings is assessed using multidimensional unfolding (Borg & Groenen, 2005; Coombs, 1964), which we will briefly introduce in the remainder of this section. Multidimensional unfolding is a technique that represents preference data as distances in a low-dimensional space. It locates K ideal points representing the subjects (in our case, $K = 288$ combinations) and M object points representing the objects (in our case, $M = 13$ methods) such that the distances from each ideal point to the object points correspond to the observed preference values. The closer an object point lies to a subject's ideal point, the stronger the subject's preference for that object. Accordingly, the ideal point itself corresponds to maximal preference (Borg et al., 2013). Note that this intuitive representation of preferences is the main reason why multidimensional unfolding is preferred over other, more widely used methods for dimension reduction, such as principal component analysis, that could alternatively be used to analyze the method rankings (for details on the differences see Chapter 16.2 in Borg & Groenen, 2005).

Multidimensional unfolding takes non-negative dissimilarities δ_{km} ($k = 1,...,K$; $m = 1,...,M$) as input, which are the preference values possibly converted in a way that small values correspond to high preferences. In our case, where the preference values are ranks, this is not necessary since a small rank already indicates high preference. Moreover, the number of dimensions dim must be specified, which we set to $dim = 2$ as it is done in most applications of multidimensional unfolding. To find the coordinates for the points representing the K subjects and M objects, a loss function (*stress*) is minimized. It is defined as

$$\sigma^2\left(\hat{\boldsymbol{D}}, \boldsymbol{Z}_1, \boldsymbol{Z}_2\right) = \sum_{k=1}^{K}\sum_{m=1}^{M} w_{km}\left(\hat{d}_{km} - d_{km}(\boldsymbol{Z}_1, \boldsymbol{Z}_2)\right)^2, \tag{2}$$

where w_{km} denotes a non-negative a priori weight (which is set to $w_{km} = 1$ by default), and $\boldsymbol{Z}_1 \in \mathbb{R}^{K \times dim}$ and $\boldsymbol{Z}_2 \in \mathbb{R}^{M \times dim}$ are the coordinates for the points representing the subjects and objects, respectively. Moreover, $d_{km}(\boldsymbol{Z}_1, \boldsymbol{Z}_2)$ denotes the fitted Euclidean distances

$$d_{km}(\boldsymbol{Z}_1, \boldsymbol{Z}_2) = \sqrt{\sum_{s=1}^{dim} (z_{1ks} - z_{2ms})^2}. \tag{3}$$

The matrix $\hat{\boldsymbol{D}} \in \mathbb{R}_0^{+K \times M}$ contains the disparities $\hat{d}_{km} = f(\delta_{km})$, which are the optimally scaled dissimilarities. This means that the loss function in Equation (2) is not only minimized with respect to $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ but also with respect to a

function $f(\cdot)$ that transforms the dissimilarities δ_{km} into disparities $\hat{d}_{km}$ (the function class depends on the assumed scale level). If, as in our example, the preference data are available in the form of ranks, $f(\delta_{km})$ reflects a monotone step function that is found through monotonic regression on the dissimilarities. This type of multidimensional unfolding is referred to as ordinal or non-metric unfolding. However, multidimensional unfolding can also be easily applied if the preference data are on a metric scale level by simply employing a different function class. In our example benchmark study, such metric preference data could be aggregated ibrier or cindex values, for instance.

To avoid degenerate solutions due to equal disparities which occur particularly often in non-metric unfolding, it is recommended to use a penalized version of the stress function in (2) that involves the coefficient of variation $v(\hat{\boldsymbol{D}})$. The penalized stress function is minimized through numerical optimization using a strategy called SMACOF (Stress Majorization of a Complicated Function) and is implemented in an R package of the same name (de Leeuw & Mair, 2009). For details on multidimensional unfolding and its implementation see Mair et al. (2021), Borg and Groenen (2005), and Busing et al. (2005).

4 | RESULTS

For full reproducibility, the entire analysis and the results presented in this section are publicly available in the GitHub repository https://github.com/NiesslC/overoptimism_benchmark.

4.1 | Overall variability and step-wise optimization

As a first step, we compare the method rankings resulting from all 288 combinations of design and analysis options. Figure 2 shows the corresponding rank distribution for each method. Importantly, it reveals that any method can achieve almost any rank. On one hand, all methods but one achieve rank 1 (8 methods) or 2 (4 methods) for at least

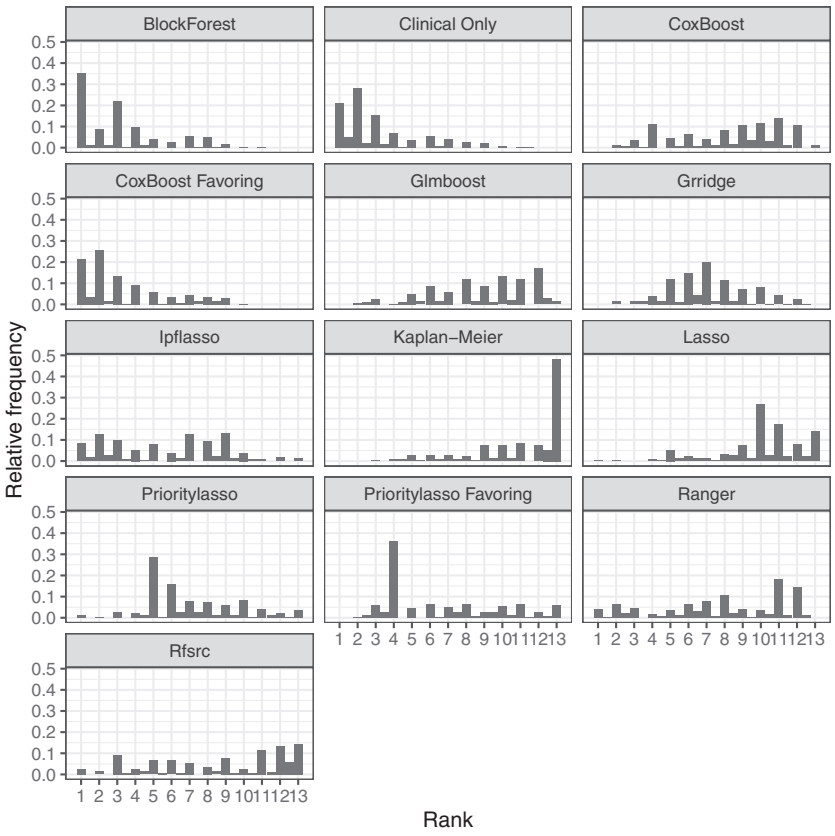

FIGURE 2 Rank distribution of 13 methods generated by 288 combinations of design and analysis options

one combination. The exception is Kaplan–Meier, which does not use any feature information and can achieve ranks as small as 3. On the other hand, 10 methods are found to be the worst or one of the two worst methods (i.e., have rank 13 or 12.5, respectively) for at least one combination. The highest rank obtained by the remaining methods (Clinical Only, BlockForest, and CoxBoost Favoring) ranges from 10 to 11.5. Figure 2 also reveals that the ranks are distributed differently for each method. For example, while Clinical Only obtains rank 1 or 2 in approximately 50% of the combinations, the ranks of Ranger are more evenly distributed.

While considering all combinations of options provides valuable information on the overall variability of results, it is not a realistic scenario concerning over-optimism in the sense that no researcher conducting a benchmark study would try all possible combinations to obtain a favorable result (unless they are actively cheating, which we do not assume here). Therefore, we additionally illustrate how easy it is to modify the method rankings if the design and analysis options are selected in a step-wise optimization process, which might represent a more realistic scenario. In our illustration, the step-wise optimization for each method is performed as follows: In each step (i.e., for each choice), the option that yields the best rank for the considered method (or the best performance value in case of equal ranks) is selected. If all options yield the same result, the default option is used. As default options, we use all 18 data sets, ibrier as primary performance measure, the 20%-threshold rule as imputation method, and the mean as aggregation method. This corresponds to the setting of Herrmann et al. (2021). Moreover, we assume that a favorable result is a small rank for a specific method. Note that this may not always be the case, for example, if one expects a reference method such as Kaplan–Meier to obtain a high rank or considers a group of several methods as target.

Figure 3 displays the optimization process if the ranks are optimized in the order: (1) imputation method, (2) aggregation method, (3) performance measure, and (4) data sets. It shows that for 8 of 13 methods, the best rank achieved by step-wise optimization corresponds to the smallest possible rank for the corresponding method (i.e., the smallest rank that can be achieved when all 288 combinations are considered) and for another three methods, the step-wise optimization achieves one rank higher than the smallest possible rank. Only two methods (Prioritylasso and Grridge) show a larger discrepancy between step-wise optimization and considering all possible combinations. However, this is not too surprising considering the few cases and thus very specific combinations where they achieve small ranks (see Figure 2). If a step is missing in the optimization process of a certain method, this indicates that the corresponding step did not

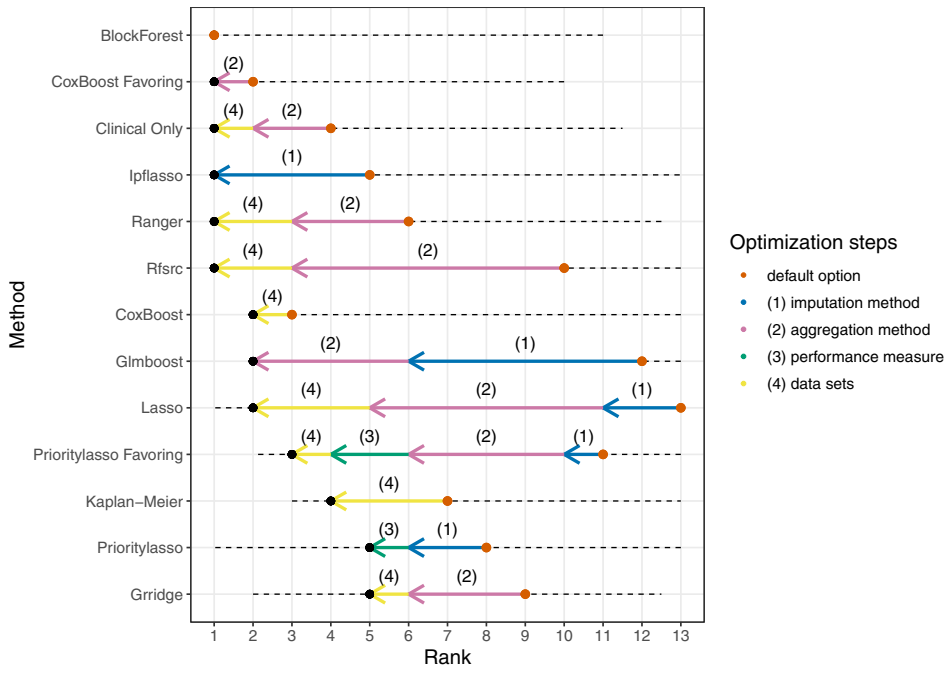

FIGURE 3 Step-wise optimization of method ranks by (1) imputation method (blue), (2) aggregation method (pink), (3) performance measure (green), and (4) data sets (yellow). The dotted line corresponds to the smallest and highest possible ranks when all 288 combinations are considered. Missing steps indicate that they did not lead to an improved rank. Default options correspond to Herrmann et al. (2021)

improve the rank of that method. In fact, all methods except Lasso and Prioritylasso Favoring require no more than two optimization steps.

Note that the results of the step-wise optimization depend on the default options. For example, when cindex instead of ibrier is used as default option, the resulting ranks are higher (see Figure S1). Moreover, the results depend on the order in which the ranks are optimized. The order shown in Figure 3 is realistic in the sense that researchers might find it more problematic to modify components of the benchmark study that are generally considered as important (i.e., performance measure or data sets) and thus only resort to them if the previous optimization steps (i.e., imputation method or aggregation method) do not yield a favorable result. However, other orders in which the ranks are optimized would also be conceivable. For example, the selection of data sets could be optimized first since it offers many options and can be easily modified by eliminating specific data sets. In this case, the selection of data sets remains the only optimization step for many methods since the subsequent steps do not lead to an improvement (see Figure S2), which already indicates the large impact of data set selection, discussed in more detail in the next section.

4.2 | Impact of individual design and analysis choices

To gain additional insight concerning the impact of each design and analysis choice, the method rankings are analyzed using multidimensional unfolding. Figure 4 displays the resulting unfolding solution that represents the rankings of all 288 combinations regarding the 13 methods. Before looking at the different colorings of the ideal points in Figure 4a–d, we can make some general observations on how the combinations and methods are scaled in the plot (which is identical for each figure). First, the unfolding solution clearly shows that the method rankings can differ widely depending on which combination of design and analysis options is considered, which is consistent with the results presented in Section 4.1. Second, similar to the rank distribution in Figure 2, the unfolding solution indicates that some methods tend to achieve smaller ranks than other methods. This applies specifically to Clinical Only, CoxBoost Favoring, and BlockForest, which are scaled close to the origin and thus have a small distance to most ideal points. In contrast, other methods such as Lasso and Kaplan–Meier can be found in the periphery of the plot, indicating that they obtain rather high ranks by most combinations.

Of course, the degree to which the presented unfolding solution reflects the actual rankings depends on its goodness-of-fit (a perfect fit usually requires as many dimensions as there are methods, i.e., $dim = M = 13$). However, following Mair et al. (2016), the unfolding solution in Figure 4 fits the ranking data reasonably well (see the Supplementary material for diagnostic figures and measures).

An important feature of the unfolding solution in Figure 4 is that not only the distances between ideal and object points can be interpreted, but also the distances within ideal and object points. This means that, in contrast to the rank distribution in Figure 2, the unfolding solution also provides information about which methods are ranked similarly and which combinations of design and analysis options yield similar rankings. We make use of the latter (i.e., the fact that the unfolding solution indicates which combinations yield similar rankings) to assess the impact of each design and analysis choice on the method rankings. For this purpose, the unfolding solution is supplemented with additional information, which results in Figure 4a–d: For each choice, the ideal points are colored according to the option that was used in the respective combination, with the default option (i.e., the option used in Herrmann et al., 2021) colored in gray. Moreover, we connect each ideal point representing the default option to the ideal points representing the alternative options given that the other three choices remain the same. Although this makes the representation dependent on which option is used as the default, for reasons of clarity, we refrain from additionally connecting the alternative options with each other.

The resulting plot for the choice of performance measure is displayed in Figure 4a. The gray lines indicate that the distances between most ideal points corresponding to pairs of ibrier and cindex within one specific setting (i.e., combinations where the other three choices remain the same) are large. Accordingly, the choice of performance measure strongly impacts the resulting method ranking for most settings. Figure 4a also reveals that the ideal points corresponding to ibrier and cindex form two clearly separated clusters. Accordingly, the variability in the method rankings is reduced if the performance measure is fixed. This applies in particular to the cindex, whose corresponding ideal points show considerably less variation than the ideal points corresponding to the ibrier. With regard to the remaining

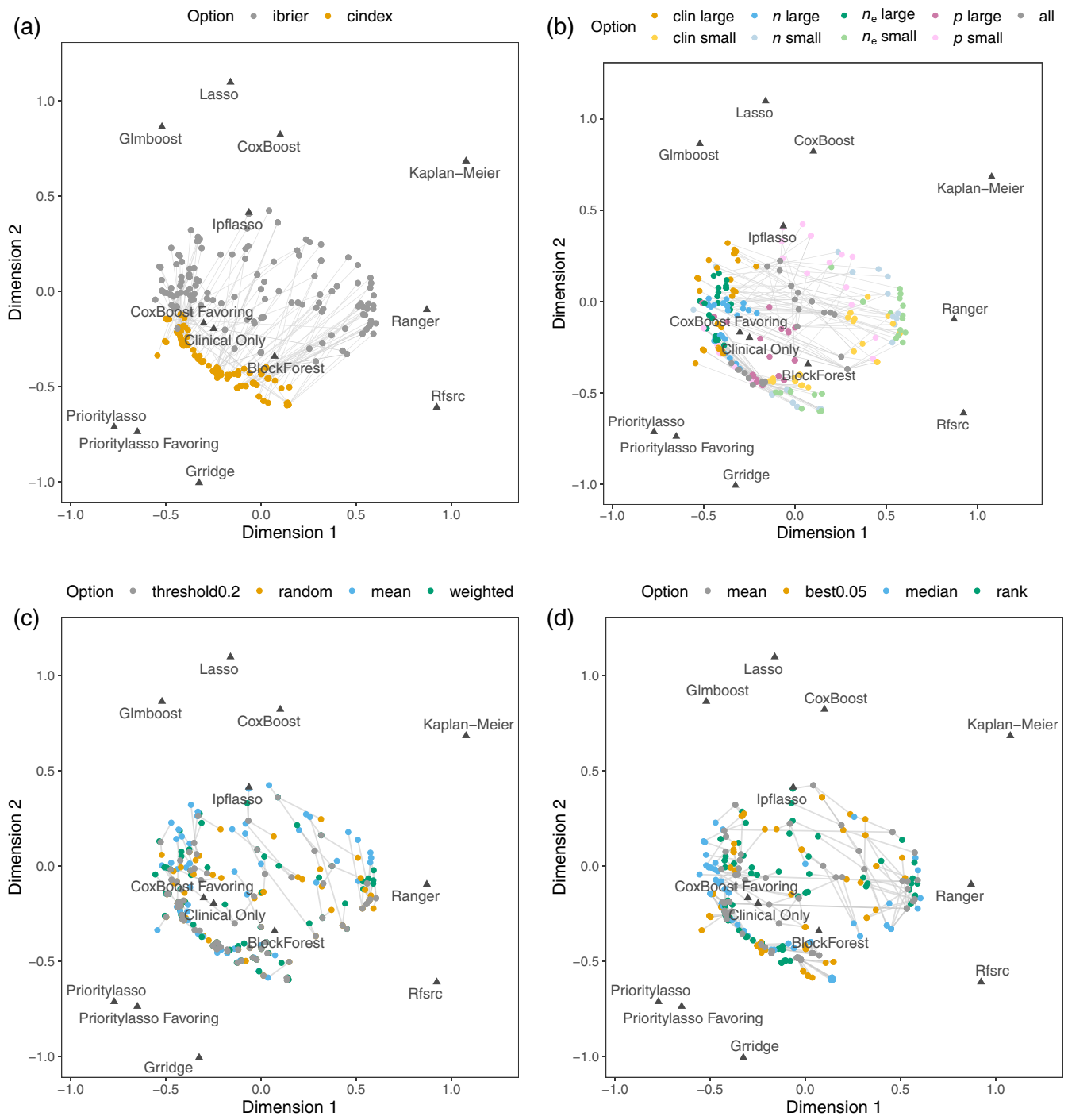

FIGURE 4 Unfolding solution representing the rankings of 288 combinations of design and analysis options (*ideal points*; circles) regarding 13 methods (triangles). For each choice, the ideal points are colored according to the option that was used in the respective combination (default options corresponding to Herrmann et al., 2021 are gray). Each ideal point representing a default option is connected to the ideal points representing alternative options, given that the other three choices remain the same. (a) Performance measure, (b) data sets, (c) imputation method, and (d) aggregation method

three choices (data sets, imputation method, and aggregation method), this means that their impact is smaller if the cindex is used as performance measure. This finding might be explained by the fact that the cindex only measures discriminatory power (see Section 2) and might thus be more robust to changes in the remaining design and analysis choices than the ibrier.

As can be seen from Figure 4b, another important choice that accounts for a large part of the variability in the method rankings is the selection of data sets, especially if the ibrier is used as performance measure (compare with Figure 4a). Figure 4b also reveals that within the two clusters corresponding to cindex and ibrier, the ideal points are roughly clustered according to the group of data sets that was used in the respective combination. This indicates that keeping the data sets fixed in addition to the performance measure again reduces the variability in the method rankings. Regarding the type of data sets used in each combination, Figure 4b shows that within both clusters of performance measure, the ideal points corresponding to small and large values of each data set characteristic lie approximately opposite to each other while the ideal points representing all 18 data sets are located between them. With regard to the choice of data sets, the largest discrepancy between two rankings can thus be expected when comparing the results of two groups that correspond to small and large values of one of the considered data set characteristics. Using all 18 data sets, on the other hand, results in a compromise between the two extremes.

As already stated above, the variability in the method rankings is considerably reduced if performance measure and data sets are fixed, which in turn means that the variations caused by using different imputation or aggregation methods are expected to be small. This finding is confirmed by Figure 4c,d. The gray lines indicate that variations in the method rankings caused by deviations from the default imputation or aggregation method mainly arise for ibrier as the performance measure and all groups of data sets except those with many clinical variables or large values of n or n_e (compare with Figure 4a,b). In some of the other settings, the impact of the choice of imputation and aggregation method is so small that the ideal points corresponding to different imputation/aggregation methods have the same coordinates (i.e., yield the same ranking). This applies in particular to the choice of imputation method, which generally has less impact on the method rankings than the choice of aggregation method, as can be seen from comparing Figure 4c and Figure 4d.

The distances between ideal points of default and alternative options that are represented as gray lines in Figure 4a–d can also be summarized as boxplots, which are displayed in Figure 5. This representation provides information that is technically also included in Figure 4a–d, but is presented more clearly in Figure 5. For example, it shows for each choice which alternative option used instead of the default option tends to yield the highest variations in the method rankings (e.g., for the choice of imputation method, it is the option that uses the mean of the non-failed iterations as imputation value). Moreover, Figure 5 reveals that according to the unfolding solution, the largest discrepancy between two rankings generated by only varying one design or analysis option is achieved by using the median instead of the mean as aggregation method. This is an unexpected finding since it has already been stated above and can also be seen from Figure 5 that in most settings (i.e., combinations where the other three choices remain the same), the choice of aggregation method tends to have a smaller impact on the method rankings than the choice of performance

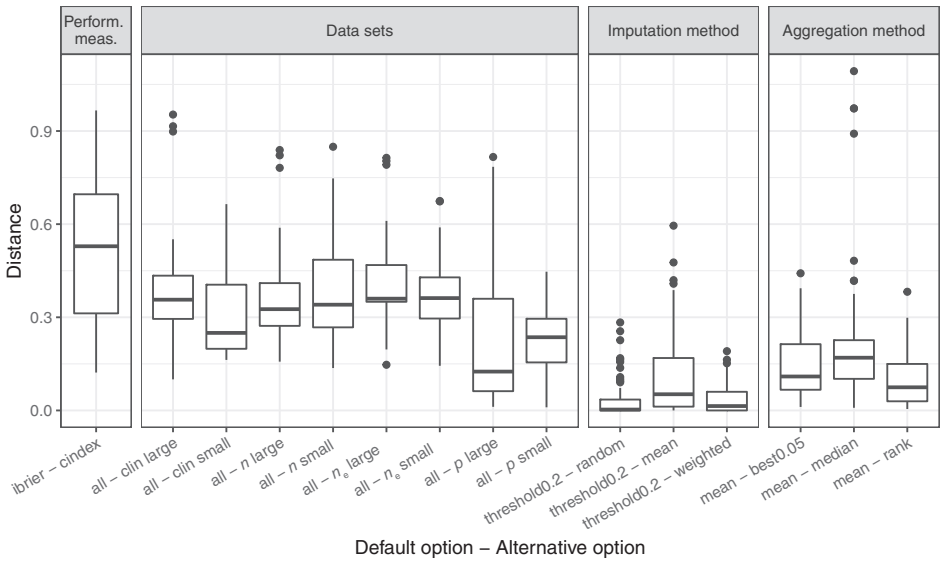

FIGURE 5 Distances between ideal points of combinations that represent default and alternative options of one specific choice (given that the other three choices remain the same), derived from the unfolding solution in Figure 4. The larger the distance, the larger the discrepancy between the two rankings generated by using the alternative option instead of the default option

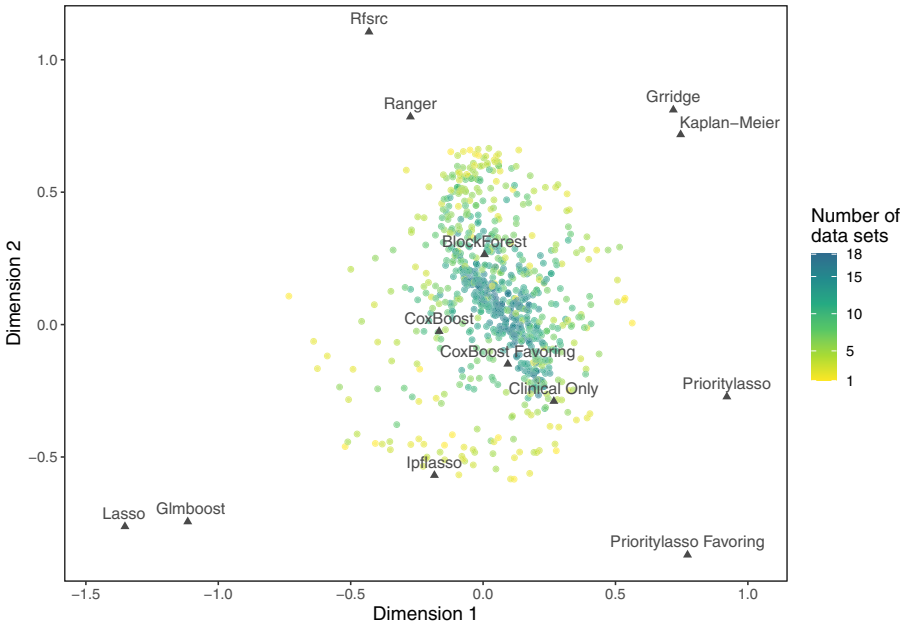

FIGURE 6 Unfolding solution representing 774 rankings (circles) of 13 methods (triangles) generated by randomly sampling different groups of data sets while performance measure, imputation method, and aggregation method are fixed to their respective default option

measure and data sets. A major drawback of Figure 5 is that in contrast to Figure 4a–d, it does not provide any information about how similar the rankings generated by the alternative options are, nor about how the ranks of the individual methods change.

Of course, all findings concerning the impact of the individual design and analysis choices depend on the number and type of options considered for each choice. Specifically, for the choice of data sets, we only consider a small subset of possible options and we focus, in addition to the 18 original data sets, on groups of approximately equal size (8–10 data sets) generated by specific data set characteristics. We thus complement our analysis by illustrating the impact of the choice of data sets if more options are considered, especially with regard to the number of data sets. For this purpose, we keep performance measure, imputation method, and aggregation method fixed to their respective default option and randomly draw 50 permutations of the 18 original data sets. For each of these permutations we store the method rankings generated by only considering the first l data sets with $l = 1, ..., 17$, and remove duplicate groups of data sets (which mainly occur for groups with 1, 2, or 17 data sets). This results in 774 rankings including one ranking generated by the 18 original data sets, which are all represented in the unfolding solution in Figure 6. The widely distributed ideal points clearly indicate that the choice of data sets is even more essential if the number of data sets is not restricted and the groups of data sets are not defined based on specific data set characteristics (as it was the case above in Figure 4). As one might have expected, we also observe that the variability in the method rankings increases if the number of data sets decreases. Accordingly, the most extreme rankings (i.e., rankings that differ the most from the ranking generated using all 18 data sets) occur for groups with only a few data sets. Since Figure 4a revealed that the impact of the choice of data sets strongly depends on the choice of performance measure, we repeat the analysis using cindex as performance measure (see Figure S3). Similar to Figure 4b, the impact of the choice of data sets is considerably reduced. However, as in Figure 6, the variability in the method rankings increases with decreasing number of data sets.

5 | DISCUSSION

5.1 | Summary

In this paper, we addressed the multiplicity of design and analysis options in the context of benchmark studies and the associated risk of over-optimistic results. As a preliminary step, we reviewed literature related to the choice of four design and analysis choices that researchers are usually faced with when conducting a benchmark study based on real

data sets, namely the choice of data sets, the choice of quantitative performance measure, the choice of imputation method for missing performance values, and the choice of aggregation method to generate an overall method ranking.

We then used the benchmark study by Herrmann et al. (2021) to illustrate how variable the resulting method rankings of a benchmark study can be when all possible combinations of a range of design and analysis options are considered. In fact, in this example, the results were so variable that any method could achieve almost any rank, that is, each method could almost be presented as best or worst method for at least one combination of design and analysis options. For the more realistic scenario where the design and analysis options are not systematically examined for each combination but selected in a step-wise optimization process, we observed that the variability in the method rankings is smaller but still remarkable.

In addition to examining the overall variability in the method rankings, we also investigated the individual impact of each choice on the results using multidimensional unfolding. As might be expected, the choice of performance measure and data sets accounts for a large part of the variability in the method rankings. The impact of the choice of imputation and aggregation method, on the other hand, tends to be considerably smaller but still non-negligible in many settings. In general, the impact of each choice depends on the options used for the other three choices, with the choice of performance measure affecting the impact of the remaining choices most strongly. In an additional analysis, we increased the number of considered options for the choice of data sets, which clearly showed that the variability in the method rankings increases if the number of data sets decreases and once again emphasized the importance of the choice of data sets.

5.2 | Limitations

Of course, the specific results obtained for the example study by Herrmann et al. (2021) should only be seen as an illustration that cannot be generalized to other benchmark studies. Moreover, one possible reason why the method rankings are so variable is that in our example benchmark study, many performance differences are small and the performance values differ widely across data sets, as discussed in the original study by Herrmann et al. (2021). The focus of our study was on ranks, which do not reflect the size of the differences between the methods' performances or the heterogeneity across data sets. On the one hand, taking these aspects into account rather than focusing on ranks may lead to much less variable results, particularly if one relies on statistical tests. On the other hand, the multiplicity of possible analysis options is not limited to the analysis of ranks: there are also plenty of possibly ways to analyze performance differences and the heterogeneity across data sets, even if statistical tests are performed (e.g., paired *t*-test or Wilcoxon signed-rank test with or without correction for multiple testing, or global tests such as the Friedman test).

5.3 | Negative consequences and possible solutions

Despite these limitations, our illustration suggests that, as a consequence of the multiplicity of design and analysis options, the results of benchmark studies could be much more variable than many researchers realize. Combined with questionable research practices (e.g., the selective reporting of results or the targeted modification of specific design and analysis components), this potentially high variability of benchmark results can lead to biased interpretations and over-optimistic conclusions regarding the performance of some of the considered methods. Given the high level of evidence that is attributed to neutral benchmark studies (Boulesteix et al., 2017), a "neutral" benchmark study that is in fact biased could thus negatively affect both methodological and applied research by misleading method users and developers (Weber et al., 2019).

Fortunately, there are several strategies to prevent over-optimistic benchmark results that arise from the multiplicity of design and analysis options, some of which are already applied by many researchers, including Herrmann et al. (2021). For example, strategies inspired from blinding in clinical trials can help to reduce non-neutrality and/or the potential to exploit the multiplicity of possible options. Specifically, blinding could be realized by labeling the methods with non-informative names (e.g., Method A, Method B, etc.) such that researchers have no information about the performance of each method until the end of the study (Boulesteix et al., 2017). If the benchmark study is based on simulated data, researchers could also be blinded to the data generation process, which prohibits the possibility to tune the parameters of selected methods according to the known ground truth (e.g., Kreutz et al., 2020).

The remaining strategies to prevent over-optimistic results can be summarized using the work of Hoffmann et al. (2021), who formalize the effect of both random sources of uncertainty (including sampling uncertainty) and

epistemic sources of uncertainty (resulting in a multiplicity of possible analysis strategies and thus opening the door to questionable research practices) on the replicability of research findings. They outline six steps researchers from all empirical research fields can take to make their own research more replicable and credible. In brief, researchers should (1) be aware of the multiplicity of possible analysis strategies, (2) reduce uncertainty, (3) integrate uncertainty, (4) report uncertainty, (5) acknowledge uncertainty, and (6) publish all research code, data and material. Although Hoffmann et al. (2021) focus on applied rather than methodological research, we argue that their recommended steps can also be applied to address the sources of uncertainty that arise from the design and analysis of benchmark studies.

Step 1. In the context of benchmark studies, the first step to reduce the risk of over-optimistic results is to simply be aware of the multiplicity of possible design and analysis options and the potential for questionable research practices. We can only speculate about how much awareness for this issue is already present in methodological research but hope that this paper contributes to raising it.

Step 2. The second step suggested by Hoffmann et al. (2021) is to reduce sources of uncertainty. In the context of benchmark studies, this could be realized by consulting existing benchmarking guidelines found in literature. However, as discussed in this paper, guidelines for many issues relevant in practice are still lacking. We claim that more guidance and standardized approaches are needed in this context. Regarding the choice of data sets, uncertainty could be reduced if the number of data sets to include in the study would be consequently based on statistical considerations such as power calculation (e.g., Boulesteix et al., 2015) and if data sets would be selected according to strict and well-considered inclusion criteria. Both aspects are facilitated if structured and well-documented databases exist for the type of data to be studied.

Step 3. As a third step, Hoffmann et al. (2021) recommend to integrate remaining sources of uncertainty that could not be reduced in the second step. Analysis approaches such as confidence intervals, statistical tests, or boxplots that take the heterogeneity of performance values across data sets into account can be seen as first steps towards integrating the uncertainty regarding the choice of data sets. However, they do not provide much information about how the benchmark results would change if only certain subgroups of data sets would be considered. A more advanced but less common way to integrate uncertainty regarding the choice of data sets is to analyze the relationship between method performance and data set characteristics (e.g., Eugster et al., 2014; Kreutz et al., 2020; Oreski et al., 2017). Concerning the choice of evaluation criteria (including quantitative performance measures), the aggregation of method rankings resulting from different criteria into an overall ranking can be seen as an attempt towards integrating uncertainty. However, to our knowledge, currently existing approaches such as consensus rankings (Hornik & Meyer, 2007) do not provide any measure of uncertainty.

Step 4. For all sources that cannot be adequately integrated, Hoffmann et al. (2021) suggest to systematically report the results of alternative analysis strategies, which, in the context of benchmark studies, would be alternative design and analysis options. While reporting the results of alternative analysis strategies, for example, in the form of a sensitivity analysis, is a common procedure in applied research (Hoffmann et al., 2021), to our knowledge it is rarely performed in benchmark studies (especially if they are based on real data sets). However, considering the lack of ways to reduce and integrate uncertainty when designing and analyzing benchmark studies, adequately reporting the results of alternative options seems to be all the more important. One reason for the lack of uncertainty reporting in benchmark studies could be that, to our knowledge, no suitable framework has been available so far. This gap could be filled by the framework based on multidimensional unfolding that we used in this paper. It can be seen as a systematic version of standard sensitivity analysis that allows to graphically assess the variability of the method rankings with respect to a large number of different combinations of design and analysis options. It also provides information about the individual impact of each choice on the method ranking and thus enables researchers to analyze when and how using alternative options for a specific choice affects the results. In this way, the risk of misleading readers is reduced and the benchmark results become even more reliable and valuable. Moreover, using the framework allows to identify critical choices that substantially affect the results and should therefore be particularly well justified in future benchmark studies and be given more consideration in benchmarking guidelines.

Step 5. The next important step suggested by Hoffmann et al. (2021) is to accept the inherent uncertainty of scientific findings. In the context of benchmark studies, this implies that researchers should clearly state that the benchmark results are conditional on the selected design and analysis options (Boulesteix et al., 2013; Hornik & Meyer, 2007). In this vein, researchers should also acknowledge that just as in applied research, generalizations from a single study are usually not appropriate (Amrhein et al., 2019; Hoffmann et al., 2021). This emphasizes the need for more high-quality benchmark studies and for meta-analyses of benchmark studies (e.g., Gardner et al., 2019), which, however, are still rare and unfortunately sometimes not considered as full-fledged research by the scientific community (Boulesteix

et al., 2020). Another aspect also related to the acceptance of uncertainty is to recognize that statistical inference within exploratory analyses should be treated with great caution (Amrhein et al., 2019; Hoffmann et al., 2021). Similar to applied research, strictly confirmatory benchmark studies could be realized by pre-registration of design- and analysis plans, as recently implemented in the context of the so-called pre-registration experiment (see https://preregister.science) or through the registered report" publication format (Chambers, 2013), which has meanwhile been adopted by several interdisciplinary journals that also accept computational papers. It is also important to recall that there is usually no best method for all scenarios and data sets (the well-known "no free lunch" theorem; Wolpert, 2002). Especially for data sets and evaluation criteria, it might thus be advisable to accept the uncertainty that is associated with their choice by putting more focus on the analysis of the individual strengths and weaknesses of each method than on an aggregated overall ranking. This can for example be realized by individually analyzing the rankings generated by each evaluation criterion and by investigating the relationship between method performance and data set characteristics (see Step 3).

Step 6. As a final step, the publication of codes and (if possible) data sets that ideally allow the extension to alternative options and additional methods can reduce the impact of over-optimism since it enables readers to run alternative analyses and to reveal potentially biased results.

The strategies provided in this section are also summarized in a checklist (Table S1), which can assist researchers when designing and analyzing benchmark studies.

6 | CONCLUSION

In conclusion, our illustration suggests that benchmark results can be highly variable with respect to design and analysis choices, which can lead to biased interpretations and over-optimistic conclusions. However, there is a wide range of strategies that can help to avoid these pitfalls. We hope that our proposed framework makes a useful contribution towards this objective. While a certain amount of over-optimism can probably never be completely avoided, addressing this problem will lead to more reliable and valuable benchmark results.

ACKNOWLEDGMENT

The authors thank Anna Jacob for language correction.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in GitHub at https://github.com/NiesslC/overoptimism_benchmark.

AUTHOR CONTRIBUTIONS

Christina Nießl: Conceptualization (equal); formal analysis (lead); methodology (lead); visualization (lead). **Moritz Herrmann:** Conceptualization (equal); data curation (equal); methodology (supporting). **Chiara Wiedemann:** Conceptualization (equal); data curation (equal); methodology (supporting). **Giuseppe Casalicchio:** Conceptualization (equal); methodology (supporting). **Anne-Laure Boulesteix:** Conceptualization (equal); funding acquisition (equal); methodology (supporting); supervision (equal).

ORCID

Christina Nießl https://orcid.org/0000-0003-2425-7858
Moritz Herrmann https://orcid.org/0000-0002-4893-5812
Giuseppe Casalicchio https://orcid.org/0000-0001-5324-5966
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947

RELATED WIREs ARTICLE

Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey

REFERENCES

Amrhein, V., Trafimow, D., & Greenland, S. (2019). Inferential statistics as descriptive statistics: There is no replication crisis if we don't expect replication. *The American Statistician*, *73*, 262–270.

Bischl, B., Schiffner, J., & Weihs, C. (2013). Benchmarking local classification methods. *Computational Statistics*, *28*, 2599–2619.

Blanche, P., Kattan, M. W., & Gerds, T. A. (2019). The *c*-index is not proper for the evaluation of *t*-year predicted risks. *Biostatistics*, *20*, 347–357.

Bokulich, N. A., Ziemski, M., Robeson, M. S., & Kaehler, B. D. (2020). Measuring the microbiome: Best practices for developing and benchmarking microbiomics methods. *Computational and Structural Biotechnology Journal*, *18*, 4048–4062.

Borg, I., & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications* (2nd ed.). Springer.

Borg, I., Groenen, P. J. F., & Mair, P. (2013). *Applied multidimensional scaling*. Springer.

Boulesteix, A.-L. (2015). Ten simple rules for reducing overoptimistic reporting in methodological computational research. *PLoS Computational Biology*, *11*, e1004191.

Boulesteix, A.-L., Binder, H., Abrahamowicz, M., & Sauerbrei, W. (2018). On the necessity and design of studies comparing statistical methods. *Biometrical Journal*, *60*, 216–218.

Boulesteix, A.-L., Hable, R., Lauer, S., & Eugster, M. J. A. (2015). A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, *69*, 201–212.

Boulesteix, A.-L., Hoffmann, S., Charlton, A., & Seibold, H. (2020). A replication crisis in methodological research? *Significance*, *17*, 18–21.

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. A. (2013). A plea for neutral comparison studies in computational sciences. *PLoS One*, *8*, e61562.

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*, 138.

Buchka, S., Hapfelmeier, A., Gardner, P. P., Wilson, R., & Boulesteix, A.-L. (2021). On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biology*, *22*, 152.

Busing, F. M. T. A., Groenen, P. J. K., & Heiser, W. J. (2005). Avoiding degeneracy in multidimensional unfolding by penalizing on the coefficient of variation. *Psychometrika*, *70*, 71–98.

Chambers, C. D. (2013). Registered reports: A new publishing initiative at Cortex. *Cortex*, *49*, 609–610.

Coombs, C. H. (1964). *A theory of data*. Wiley.

De Cnudde, S., Martens, D., Evgeniou, T., & Provost, F. (2020). A benchmarking study of classification techniques for behavioral data. *International Journal of Data Science and Analytics*, *9*, 131–173.

de Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software*, *31*, 1–30.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, *10*, 1895–1923.

Eisinga, R., Heskes, T., Pelzer, B., & Grotenhuis, M. (2017). Exact *p*-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers. *BMC Bioinformatics*, *18*, 68.

Eugster, M. J. A., Hothorn, T., & Leisch, F. (2012). Domain-based benchmark experiments: Exploratory and inferential analysis. *Austrian Journal of Statistics*, *41*, 5–26.

Eugster, M. J. A., Leisch, F., & Strobl, C. (2014). (Psycho-)analysis of benchmark experiments: A formal framework for investigating the relationship between data sets and learning algorithms. *Computational Statistics and Data Analysis*, *71*, 986–1000.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*, 3133–3181.

Gardner, P. P., Watson, R. J., Morgan, X. C., Draper, J. L., Finn, R. D., Morales, S. E., & Stott, M. B. (2019). Identifying accurate metagenome and amplicon software via a metaanalysis of sequence to taxonomy benchmarking studies. *PeerJ*, *7*, e6160.

Gatto, L., Hansen, K. D., Hoopmann, M. R., Hermjakob, H., Kohlbacher, O., & Beyer, A. (2016). Testing and validation of computational methods for mass spectrometry. *Journal of Proteome Research*, *15*, 809–814.

Graf, E., Schmoor, C., Sauerbrei, W., & Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, *18*, 2529–2545.

Head, M. L., Holman, L., Lanfear, R., Kahn, A. T., & Jennions, M. D. (2015). The extent and consequences of p-hacking in science. *PLoS Biology*, *13*, e1002106.

Herrmann, M., Probst, P., Hornung, R., Jurinovic, V., & Boulesteix, A.-L. (2021). Large scale benchmark study of survival prediction methods using multi-omics data. *Briefings in Bioinformatics*, *22*, bbaa167. https://doi.org/10.1093/bib/bbaa167

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., & Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, *8*, 201925.

Hornik, K., & Meyer, D. (2007). Deriving consensus rankings from benchmarking experiments. In R. Decker & H.-J. Lenz (Eds.), *Advances in data analysis* (pp. 163–170). Springer.

Hothorn, T., Leisch, F., Zeileis, A., & Hornik, K. (2005). The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, *14*, 675–699.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLoS Medicine*, *2*, e124.

Ioannidis, J. P. A., Munafò, M. R., Fusar-Poli, P., Nosek, B. A., & David, S. P. (2014). Publication and other reporting biases in cognitive sciences: Detection, prevalence, and prevention. *Trends in Cognitive Sciences*, *18*, 235–241.

Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., & Boulesteix, A.-L. (2010). Overoptimism in bioinformatics: An illustration. *Bioinformatics*, *26*, 1990–1998.

John, L. K., Loewenstein, G., & Prelec, D. (2012). Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, *23*, 524–532.

Kibekbaev, A., & Duman, E. (2016). Benchmarking regression algorithms for income prediction modeling. *Information Systems*, *61*, 40–52.

Kreutz, C. (2019). Guidelines for benchmarking of optimization approaches for fitting mathematical models. *Genome Biology*, *20*, 281.

Kreutz, C., Can, N. S., Bruening, R. S., Meyberg, R., Mérai, Z., Fernandez-Pozo, N., & Rensing, S. A. (2020). A blind and independent benchmark study for detecting differentially methylated regions in plants. *Bioinformatics*, *36*, 3314–3321.

MacIà, N., Bernadó-Mansilla, E., Orriols-Puig, A., & Kam Ho, T. (2013). Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition*, *46*, 1054–1066.

Mair, P., Borg, I., & Rusch, T. (2016). Goodness-of-fit assessment in multidimensional scaling and unfolding. *Multivariate Behavioral Research*, *51*, 772–789.

Mair, P., Groenen, P. J. F., & de Leeuw, J. (2021). More on multidimensional scaling and unfolding in R: smacof version 2. *Journal of Statistical Software*. https://cran.r-project.org/web/packages/smacof/vignettes/smacof.pdf

Mangul, S., Martin, L. S., Hill, B. L., Lam, A. K. M., Distler, M. G., Zelikovsky, A., Eskin, E., & Flint, J. (2019). Systematic benchmarking of omics computational tools. *Nature Communications*, *10*, 1393.

Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., & Weihs, C. (2015). Analyzing the BBOB results by means of benchmarking concepts. *Evolutionary Computation*, *23*, 161–185.

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*, 2074–2102.

Norel, R., Rice, J. J., & Stolovitzky, G. (2011). The self-assessment trap: Can we all be better than average? *Molecular Systems Biology*, *7*, 537.

Novianti, P. W., Jong, V. L., Roes, K. C., & Eijkemans, M. J. (2015). Factors affecting the accuracy of a class prediction model in gene expression data. *BMC Bioinformatics*, *16*, 199.

Nuzzo, R. (2015). How scientists fool themselves —And how they can stop. *Nature*, *526*, 182–185.

Oreski, D., Oreski, S., & Klicek, B. (2017). Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, *52*, 109–119.

Orzechowski, P., La Cava, W., & Moore, J. H. (2018). Where are we now? A large benchmark study of recent symbolic regression methods. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18*, Association for Computing Machinery, New York, NY, USA (pp. 1183–1190).

Robinson, M. D., & Vitek, O. (2019). Benchmarking comes of age. *Genome Biology*, *20*, 205.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*, 1359–1366.

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., & Wei, L. J. (2011). On the *C*-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in Medicine*, *30*, 1105–1117.

Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Teinemaa, I. (2019). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology*, *10*, 34.

Wagenmakers, E. J., Wetzels, R., Borsboom, D., van der Maas, H. L. J., & Kievit, R. A. (2012). An agenda for purely confirmatory research. *Perspectives on Psychological Science*, *7*, 632–638.

Weber, L. M., Saelens, W., Cannoodt, R., Soneson, C., Hapfelmeier, A., Gardner, P. P., Boulesteix, A.-L., Saeys, Y., & Robinson, M. D. (2019). Essential guidelines for computational method benchmarking. *Genome Biology*, *20*, 125.

Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, & F. Hoffmann (Eds.), *Soft computing and industry: Recent applications* (pp. 25–42). Springer.

Wu, Z., Zhu, M., Kang, Y., Leung, E. L.-h., Lei, T., Shen, C., Jiang, D., Wang, Z., Cao, D., & Hou, T. (2020). Do we need different machine learning algorithms for QSAR modeling? A comprehensive assessment of 16 machine learning algorithms on 14 QSAR data sets. *Briefings in Bioinformatics*, *22*, bbaa321. https://doi.org/10.1093/bib/bbaa321

Yousefi, M. R., Hua, J., Sima, C., & Dougherty, E. R. (2010). Reporting bias when using real data sets to analyze classification performance. *Bioinformatics*, *26*, 68–76.

Zimmermann, A. (2020). Method evaluation, parameterization, and result validation in unsupervised data mining: A critical survey. *WIREs Data Mining and Knowledge Discovery*, *10*, e1330.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(2), e1441. https://doi.org/10.1002/widm.1441

Supplementary Materials for "Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results"

Christina Nießl * [1], Moritz Herrmann[2], Chiara Wiedemann[1], Giuseppe Casalicchio[2], and Anne-Laure Boulesteix[1]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig Maximilians University Munich (Germany)
[2]Department of Statistics, Ludwig Maximilians University Munich (Germany)

*Corresponding author, e-mail: cniessl@ibe.med.uni-muenchen.de, Institute for Medical Information Processing, Biometry and Epidemiology, Ludwig Maximilians University Munich, Marchioninistr. 15, D-81377, Munich, Germany.

A Weighted imputation method for missing cindex performance values

For the cindex, where 1 corresponds to the best possible value and 0.5 to random prediction, the imputed value for the considered combination of data set and method that corresponds to the proposed "weighted imputation method" is

$$x_{impute} = 0.5 + (\frac{\sum_{i \in \mathcal{I}} x_i}{|\mathcal{I}|} - 0.5)_+ \cdot (1 - r), \tag{1}$$

where $\mathcal{I}$ is the set of indices of the non-failed iterations, x_i is the cindex value for iteration $i \in \mathcal{I}$ and r is the proportion of missing values.

B Additional figures step-wise optimisation

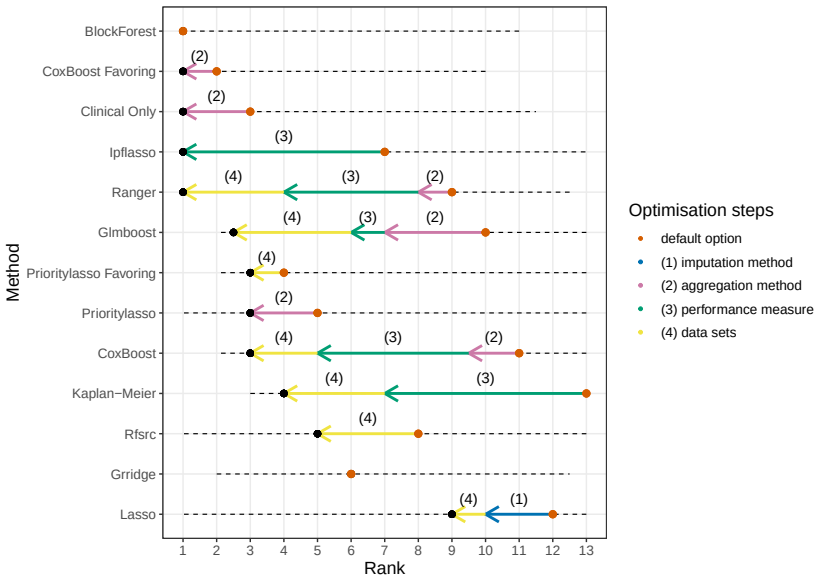

Figure S1: Step-wise optimisation of method ranks by (1) imputation method (blue), (2) aggregation method (pink), (3) performance measure (green), and (4) data sets (yellow). The dotted line corresponds to the smallest and highest possible ranks when all 288 combinations are considered. Missing steps indicate that they did not lead to an improved rank. Default options correspond to Herrmann et al. (2021) except performance measure, which is set to cindex.

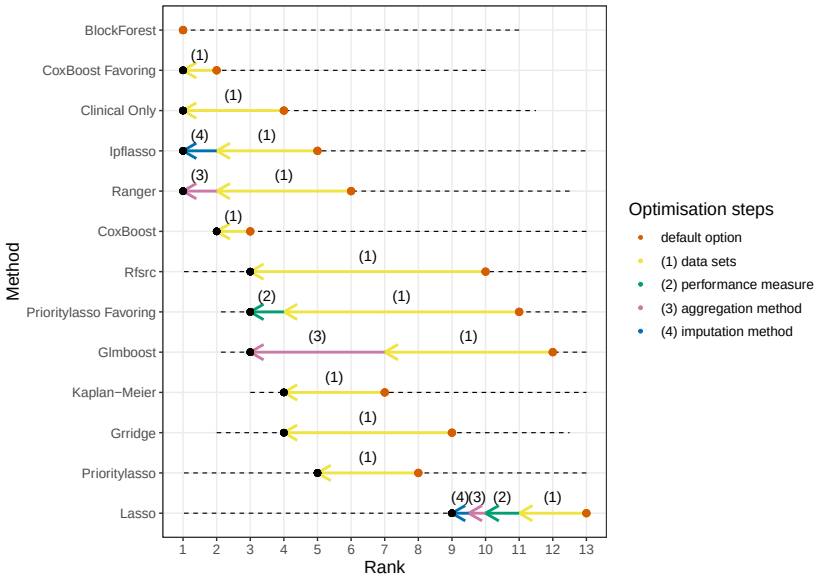

Figure S2: Step-wise optimisation of method ranks by (1) data sets (yellow), (2) performance measure (green), (3) aggregation method (pink), and (4) imputation method (blue). The dotted line corresponds to the smallest and highest possible ranks when all 288 combinations are considered. Missing steps indicate that they did not lead to an improved rank. Default options correspond to Herrmann et al. (2021).

C Additional figures unfolding

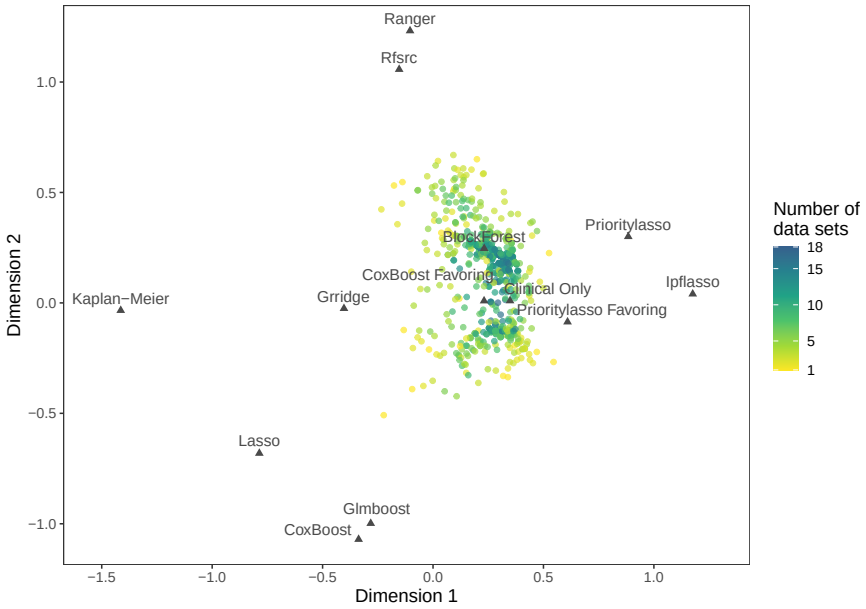

Figure S3: Unfolding solution representing 774 rankings (circles) of 13 methods (triangles) generated by randomly sampling different groups of data sets while imputation method and aggregation method are fixed to their respective default option and performance measure is set to cindex.

D Goodness-of-fit unfolding solutions

The assessment of the goodness-of-fit is based on Mair et al. (2016). We assess the fit of three unfolding models presented in this paper:

- Model 1: Unfolding solution representing the rankings of 288 combinations of design and analysis options regarding 13 methods
- Model 2: Unfolding solution representing 774 rankings of 13 methods generated by randomly sampling different groups of data sets while performance measure, imputation method, and aggregation method are fixed to their respective default option
- Model 3: Unfolding solution representing 774 rankings of 13 methods generated by randomly sampling different groups of data sets while imputation method and aggregation method are fixed to their respective default option and performance measure is set to cindex

Permutation test We test the null hypothesis that the unfolding solution is obtained from a random permutation of dissimilarities. Rejecting the null hypothesis provides some evidence that the unfolding solution captures a structural signal. For all three unfolding models, the resulting p-value is < 0.001.

Scree plots We generate scree plots with varying number of dimensions (i.e. $dim = 1, \ldots, 12$, since $dim = 13$ results in a stress value of 0). Ideally, we would see an elbow at $dim = 2$ (the dimension chosen for the unfolding models in this paper), which would indicate that additional dimensions represent only random components of the data (Borg et al., 2013). Although no clear elbow is visible in Figure S4-S6, the scree plots indicate that the stress is already considerably low for $dim = 2$. Note that in Figure S6, the iteration limit was reached when running the unfolding models for $dim \geq 6$ and the stress is close to 0, which might indicate degenerate solutions (i.e. solutions that yield extremely small stress values but are uninformative representations of the data since the distances between subject and object points are all practically equal; Borg et al., 2013).

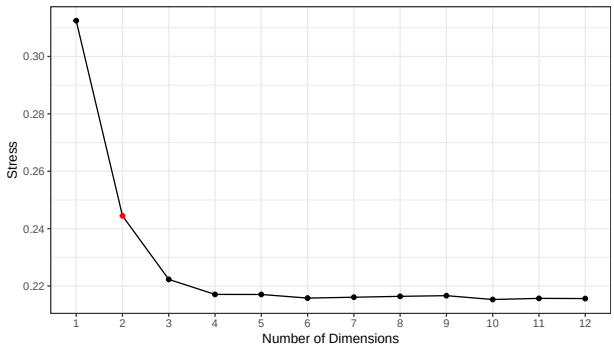

Figure S4: Scree plot for unfolding model 1. The stress value for $dim = 2$, which was used in our application, is coloured in red.

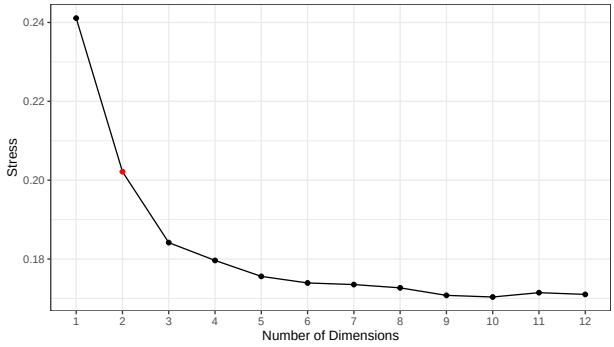

Figure S5: Scree plot for unfolding model 2. The stress value for $dim = 2$, which was used in our application, is coloured in red.

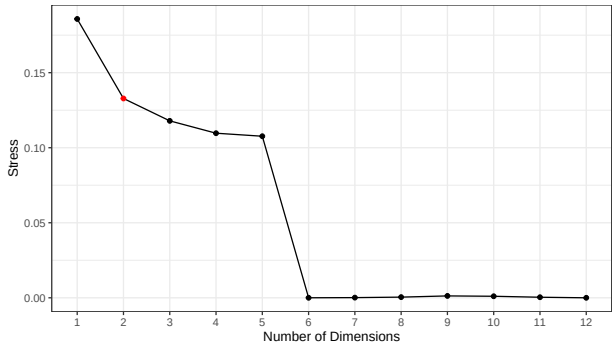

Figure S6: Scree plot for unfolding model 3. The stress value for $dim = 2$, which was used in our application, is coloured in red.

Stress-per-point To check for influential points that should be subject to special consideration, we can look at the stress-per-point values (SPP). The SPP values are assessed separately for subjects (here: combinations of design and analysis options) and objects (here: methods). As can be seen from Figure S7-S9, there are no extreme outliers for any of the three unfolding models presented in this paper. On the method side, all stress proportions are smaller than 14%, and on the combination side, most stress proportions are smaller than 1%.

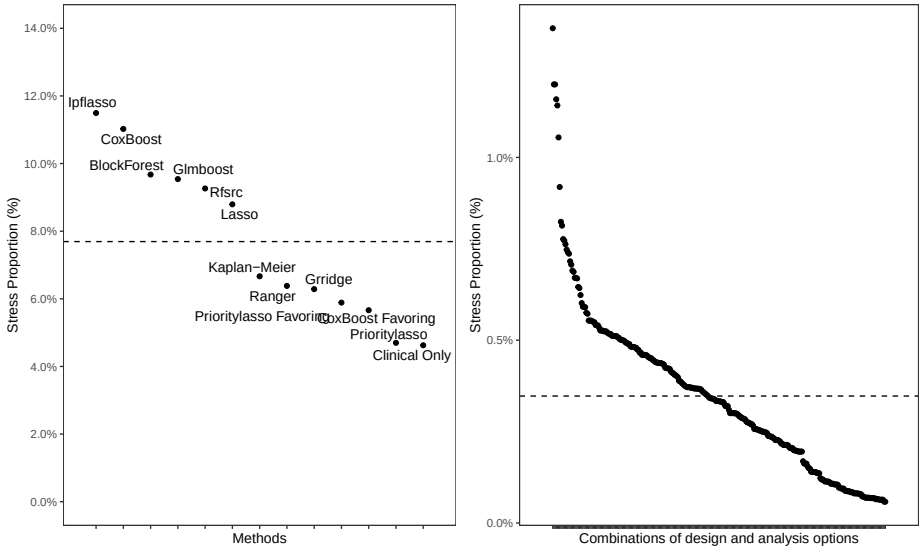

Figure S7: Stress-per-point for methods (left) and combinations of design and analysis options (right) for unfolding model 1. The greater the stress proportion, the more the point contributes to the misfit of the unfolding solution. The dotted line represents the stress proportion if every method/combination contributed equally to the misfit.

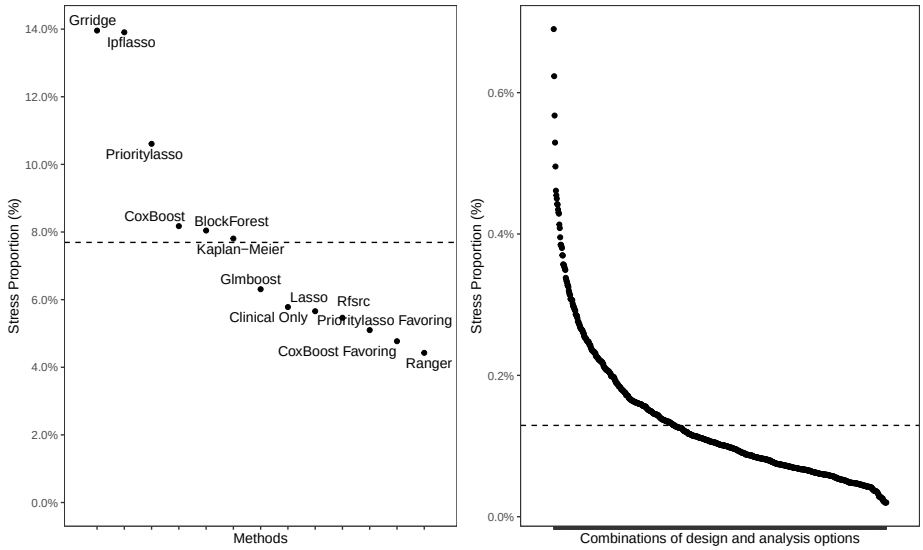

Figure S8: Stress-per-point for methods (left) and combinations of design and analysis options (right) for unfolding model 2. The greater the stress proportion, the more the point contributes to the misfit of the unfolding solution. The dotted line represents the stress proportion if every method/combination contributed equally to the misfit.

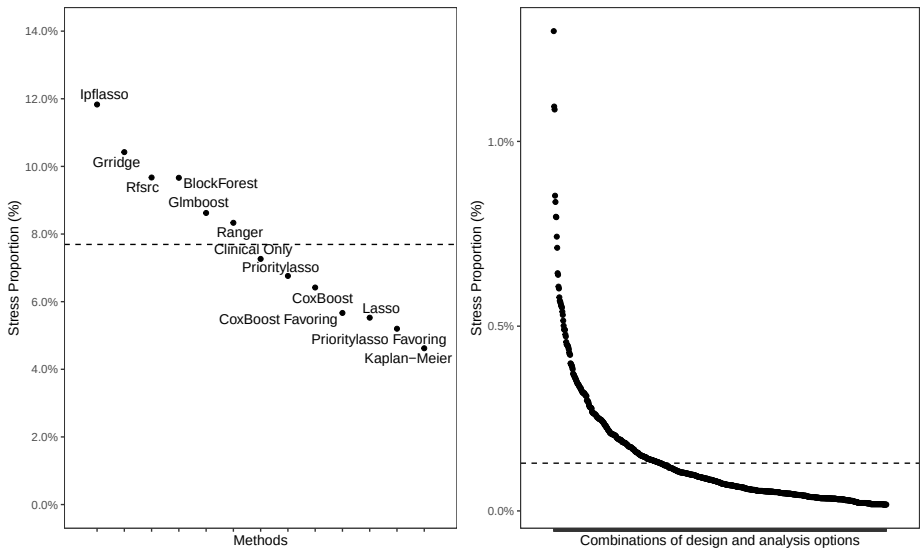

Figure S9: Stress-per-point for methods (left) and combinations of design and analysis options (right) for unfolding model 3. The greater the stress proportion, the more the point contributes to the misfit of the unfolding solution. The dotted line represents the stress proportion if every method/combination contributed equally to the misfit.

E Overview of strategies to prevent over-optimistic results in benchmark studies

Table S1: Overview of strategies to prevent over-optimistic results in benchmark studies. Most strategies can be summarised using the work of Hoffmann et al. (2021), who outline six steps researchers from all empirical fields can take to make their research more replicable and credible.

Strategy	**Implementation in the context of benchmark studies**
Step 1: Awareness	• Be aware of the multiplicity of design and analysis options in benchmark studies and the potential for questionable research practices
Step 2: Reduce uncertainty	• Consult existing benchmarking guidelines • Base the number of data sets on statistical considerations (e.g. power calculation) • Select data sets according to strict and well-considered inclusion criteria • If possible, use structured and well-documented data bases
Step 3: Integrate uncertainty	• Use analysis approaches such as confidence intervals, statistical tests or boxplots to assess the heterogeneity of performance values across data sets • More advanced: analyse the relationship between method performance and data set characteristics
Step 4: Report uncertainty	• Report the results of alternative design and analysis options, e.g. using the framework based on multidimensional unfolding
Step 5: Accept uncertainty	• Clearly state that the benchmark results are conditional on the selected design and analysis options • Treat statistical inference within exploratory analysis with caution $\rightarrow$ confirmatory benchmark studies can be realised by pre-registration or registered reports • Recall that there is usually no best method for all scenarios and data sets $\rightarrow$ more focus on analysing the individual strengths and weaknesses of each method
Step 6: Data/code availability	• Publish all code and (if possible) data sets that ideally allow the extension to alternative options and additional methods
Blinding	• Label the methods with non-informative names (e.g. Method A, Method B, etc.) • For simulated data: blinding to the data generation process

References

Borg, I., Groenen, P. J. F., and Mair, P. (2013). *Applied multidimensional scaling*. Springer, Berlin Heidelberg.

Herrmann, M., Probst, P., Hornung, R., Jurinovic, V., and Boulesteix, A.-L. (2021). Large-scale benchmark study of survival prediction methods using multi-omics data. *Briefings in Bioinformatics*. 10.1093/bib/bbaa167.

Hoffmann, S., Schönbrodt, F., Elsas, R., Wilson, R., Strasser, U., and Boulesteix, A.-L. (2021). The multiplicity of analysis strategies jeopardizes replicability: Lessons learned across disciplines. *Royal Society Open Science*, 8:201925.

Mair, P., Borg, I., and Rusch, T. (2016). Goodness-of-fit assessment in multidimensional scaling and unfolding. *Multivariate Behavioral Research*, 51:772–789.

C Contribution 3: "Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment"

This section is a reprint of:

Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2024). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *Biometrical Journal*, *66*(1), 2200238. https://doi.org/10.1002/bimj.202200238

Copyright:

Author contributions:

C. Nießl, S. Hoffmann, and A.-L. Boulesteix conceptualized the paper. C. Nießl designed the methodology and conducted the search for suitable benchmark studies to be included in the experiment, with support from S. Hoffmann and A.-L. Boulesteix. The code for the analysis and the original draft of the manuscript were written by C. Nießl. All authors contributed to the review and editing of the manuscript.

Received: 31 August 2022 | Revised: 9 January 2023 | Accepted: 10 January 2023

DOI: 10.1002/bimj.202200238

Biometrical Journal

RESEARCH ARTICLE

Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment

Christina Nießl[1,2] | **Sabine Hoffmann**[1,3] | **Theresa Ullmann**[1] | **Anne-Laure Boulesteix**[1]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, LMU Munich, Munich, Germany

[2]Munich Center for Machine Learning (MCML), Munich, Germany

[3]Department of Statistics, LMU Munich, Munich, Germany

Correspondence
Christina Nießl, Institute for Medical Information Processing, Biometry and Epidemiology, LMU Munich, Marchioninistr. 15, 81377, Munich, Germany.
Email: cniessl@ibe.med.uni-muenchen.de

Funding information
German Federal Ministry of Education and Research, Grant/Award Number: 01IS18036A; Deutsche Forschungsgemeinschaft, Grant/Award Numbers: BO3139/4-3, BO3139/7-1

This article has earned an open data badge "**Reproducible Research**" for making publicly available the code necessary to reproduce the reported results. The results reported in this article could fully be reproduced.

Abstract

The constant development of new data analysis methods in many fields of research is accompanied by an increasing awareness that these new methods often perform better in their introductory paper than in subsequent comparison studies conducted by other researchers. We attempt to explain this discrepancy by conducting a systematic experiment that we call "cross-design validation of methods". In the experiment, we select two methods designed for the same data analysis task, reproduce the results shown in each paper, and then reevaluate each method based on the study design (i.e., datasets, competing methods, and evaluation criteria) that was used to show the abilities of the other method. We conduct the experiment for two data analysis tasks, namely cancer subtyping using multiomic data and differential gene expression analysis. Three of the four methods included in the experiment indeed perform worse when they are evaluated on the new study design, which is mainly caused by the different datasets. Apart from illustrating the many degrees of freedom existing in the assessment of a method and their effect on its performance, our experiment suggests that the performance discrepancies between original and subsequent papers may not only be caused by the nonneutrality of the authors proposing the new method but also by differences regarding the level of expertise and field of application. Authors of new methods should thus focus not only on a transparent and extensive evaluation but also on comprehensive method documentation that enables the correct use of their methods in subsequent studies.

KEYWORDS
benchmarking, overoptimism, performance, reproducibility, validation

1 | INTRODUCTION

In the literature on data analysis methods, including statistical journals, machine learning journals, and conference proceedings, most articles describe *new* methods, thus contributing to an increasing multitude of potential methods addressing various data analysis problems. It is commonly claimed by the authors proposing these new methods that they perform better than existing ones in some sense. For anecdotal evidence in the context of supervised classification, see Boulesteix et al. (2013). The fact that new methods are typically claimed to be better than existing ones does not necessarily imply that these statements are wrong. In fact, this is what we would expect if we assume continuous scientific progress. However, the recurrent character of these claims, combined with the requirement of journals and reviewers to make these sorts of statements regarding the superiority of the proposed methods, make them somewhat suspicious (Boulesteix et al., 2013; Norel et al., 2011). A recent survey of papers that compare preprocessing methods for a special type of high-throughput molecular data indicates that, at least in this specific context, the paper introducing a method is indeed more optimistic regarding its performance than subsequent papers that are more neutral towards the method (Buchka et al., 2021).

In a different but related approach, several studies demonstrate that it is relatively easy to make a method appear better than it actually is (Jelizarow et al., 2010; Nießl et al., 2022; Pawel et al., 2022; Sonabend et al., 2022; Ullmann et al., 2023). These studies suggest that overoptimistic statements regarding a method's performance may be partly attributed to the nonneutral attitude of the authors, who are naturally interested to present their method in a positive light. More precisely, it is argued that the nonneutrality may translate into a conscious or subconscious optimization of the method and the study design in which it is evaluated (e.g., by selectively reporting the considered datasets or simulation parameters) such that the proposed method shows good performance.

Imagine there are two methods, method 1 and method 2, available to address a specific data analysis task. We set the number of methods to two for the sake of simplicity, but the following arguments can be extended to a setting with more than two methods. Further, imagine the typical situation in which the authors of method 1 and the authors of method 2 both claim that their method performs well. The study designs they use to support their claims are different. We will call them design 1 and design 2, respectively. Following the conjecture discussed in the previous paragraph that study designs used by authors of methods may overfit their methods and vice versa, a natural question is how method 1 would perform when reevaluated using design 2 and how method 2 would perform when reevaluated using design 1. In the present paper, we put this idea into practice by conducting a systematic experiment, which we call "cross-design validation of methods". More precisely, we consider two exemplary data analysis tasks, namely multiomic data integration for cancer subtyping and differential gene expression analysis, and for each exemplary task we select two papers that propose a new method. For each of these two methods, we reproduce the evaluation shown in the paper that introduced it and then reevaluate it on the design used by the authors of the other paper. In this context of *methodological* research, where data analysis methods are considered as research objects, the study design includes datasets (with a focus on real data for the first task and simulated data for the second task), competing methods, and evaluation criteria.

The goal of this cross-design validation experiment is twofold. First, it allows dissection of the variability of study designs and its impact on the results in the context of methodological research—in a similar way as so-called multianalyst experiments (Silberzahn et al., 2018) do in application fields of statistics. Second, the cross-design validation experiment provides insights into the mechanisms leading to performance discrepancies such as those observed by Buchka et al. (2021) between the original paper (i.e., the paper that introduces the method of interest) and subsequent papers (i.e., papers that propose another method and include the method of interest as competitor or papers that are dedicated to method comparison itself). Importantly, these are *real-world* observations—as opposed to the previous experiments by Jelizarow et al. (2010), Ullmann et al. (2023), and Pawel et al. (2022), mimicking the behavior of fictional researchers who wish to present their method in a favorable light. Finally, our experiment also provides insights regarding the reproducibility of results and the difficulty of performing fair method comparisons as a by-product.

Because the authors of our selected papers made code and data available for the purpose of reproducibility, our experiment can be performed without involving them personally, which considerably simplifies its organization and execution. Moreover, while we could also gain insights from reevaluating the methods of a study design selected by ourselves, the cross-design character of the experiment guarantees a certain degree of neutrality of our comparisons.

The remainder of this paper is structured as follows. The general structure of our cross-design validation experiment is outlined in Section 2. Sections 3 and 4 present the two data analysis tasks, while a discussion of the mechanisms leading to the observed performance differences along with a summary can be found in Section 5. We conclude the paper in Section 6.

2 | PRELIMINARY REMARKS AND DESIGN OF THE EXPERIMENT

2.1 | Terminology

Before describing the experiment in more detail, we briefly clarify the terminology used throughout the paper. Similar to Klau et al. (2020) and Buchka et al. (2021), we define the term *method* not just as the statistical testing or modeling approach, but as the full analysis pipeline potentially including steps such as data normalization. All methods considered in the experiment have several parameters that can be set by the method user (e.g., the maximum number of clusters or the type of multiple testing correction), which we refer to as *method parameters*.

Moreover, we define the *study design* as the combination of all components that contribute to the performance assessment of the method of interest. The study design consists of three main components, namely datasets (real or simulated), competing methods (including their respective method parameters), and evaluation criteria (in our exemplary analysis tasks referring to the evaluation metric and the way the results are aggregated across the real datasets or simulation repetitions). Note that data preprocessing can be seen both as part of the method or part of the data component. In our experiment, we consider preprocessing steps as belonging to the data component if performed for *all* methods and belonging to the method (i.e., the method of interest or the competing methods) otherwise.

2.2 | Selection of the papers

As a preliminary step, we first have to select appropriate papers for both exemplary data analysis tasks that we consider in our experiment, namely cancer subtyping using multiomic data and differential gene expression analysis. Both are applications from the field of biostatistics at the interface with bioinformatics. Apart from the requirement that the paper must introduce a new method, there are two eligibility criteria related to reproducibility: (i) the code to reproduce the results presented in the paper is *publicly* available and can be run without errors, and (ii) the code is written in `R`, the programming language we are most familiar with. Note that the number of papers to be included in the experiment is not specified in advance and the search for eligible papers is not conducted in a formal or systematic way.

While the restriction to `R` as a programming language (ii) excludes some papers, the majority of papers fail criterion (i). In many cases, authors only provide the code to use their method (e.g., an `R` package) but not to reproduce the results shown in the paper. In other cases, the link to the code is broken, the code supposedly included in the supplement cannot be found, or some of the files needed to reproduce the code are missing (e.g., the file containing the empirical data the simulation shown in the paper is based on). Note that we purposely refrain from contacting the authors if the code is not publicly available to make the selection of papers independent of the authors' willingness to respond and provide the code. Although we do not restrict the number of selected papers in advance, the above-mentioned difficulties lead us to stop the search after finding two eligible papers per data analysis task, resulting in $2 \times 2 = 4$ papers included in our experiment.

The conclusion of this search process, although being limited to specific analysis tasks and conducted informally, is that the practice of making code and data openly available is far from being the standard in the methodological literature beyond positive exceptions such as the *Biometrical Journal* (Hofner et al., 2016). The four papers included in our experiment (Nguyen et al., 2019; Osabe et al., 2021; Rappoport & Shamir, 2019; Zhou et al., 2021) should thus be seen as rare positive examples of open research practices in methodological research.

2.3 | Design of the experiment

While all four papers evaluate their respective method extensively in various settings, our experiment includes only the results that (i) are presented as figures or tables and appear in the main paper, that is, excluding the supplement (to keep the experiment feasible) and (ii) compare the method of interest to competing methods (since we can only compare the *relative* performance of a method if the considered papers use different evaluation metrics that do not allow a direct comparison). If the results are based on both real and simulated data, we only consider the results of the data type that is predominantly employed in the paper. In some cases, we exclude more results, which are reported and justified in Sections 3.1 and 4.1.

For each of the four papers, we first compare the results obtained by running the available code to the results presented in the corresponding paper. For this purpose, we use the same `R` and `R` package versions that were used by the authors,

TABLE 1 Illustration of the cross-design validation experiment.

	Performance of method 1	Performance of method 2
Study design by authors of method 1	**A**: Shown in the original paper	**B**: ?
Study design by authors of method 2	**C**: ?	**D**: Shown in the original paper

as far as this information is provided (see Tables S1 and S6 in the Supporting Information). Moreover, we do not modify the code in a way that would change the results, even in cases where we notice discrepancies between the code and the procedure described in the paper (referred to as "design-implementation-gap" by Lohmann et al., 2022, in the context of simulation studies). Exceptions to these rules are explicitly reported in Sections 3.2 and 4.2.

For both data analysis tasks, we then reevaluate each method on the study design used by the authors of the other paper and compare the resulting performances. Our experiment can thus be seen as a "cross-design validation of methods" (see Table 1). As stated above, the study design consists of three main components, namely datasets, competing methods, and evaluation criteria. We also vary these components individually, which allows us to assess their individual impact on the performance of the selected methods. Some challenges arise when reevaluating the methods on the new study design, in particular the choice of method parameters, which we set before viewing the performance results to avoid the risk of favoring one of the methods. Moreover, while we generally adhere to the code used to reproduce the results when "crossing" the designs, some modifications are necessary. Details on how we address these challenges for each data analysis task can be found in Sections 3.2 and 4.2.

The `R` code and data to reproduce the experiment are openly available at https://doi.org/10.6084/m9.figshare.20754028

3 | DATA ANALYSIS TASK I: CANCER SUBTYPING USING MULTIOMIC DATA

The first exemplary data task we consider in our experiment is cancer subtyping through clustering of patients based on multiomic data, an active research field with many newly proposed methods in recent years (see Duan et al., 2021, for an overview). The aim of these methods is to identify clusters (in this context referred to as *subtypes*) with common biological characteristics or clinical phenotypes (e.g., survival time or drug response). This process helps to understand the etiology of the disease and to develop better diagnostic tools and personalized treatment strategies (Duan et al., 2021; Subramanian et al., 2020; Tepeli et al., 2020). Recently developed cancer subtyping methods are usually able to integrate multiple types of high-dimensional molecular data such as genomics, epigenomics, transcriptomics, or proteomics (hence the term *multiomic* data; Subramanian et al., 2020). The two methods selected for our experiment are PINSPlus and NEMO, which were proposed by Nguyen et al. (2019) and Rappoport and Shamir (2019), respectively. Information on where to find the original codes provided by the authors is listed in our code documentation. We will abbreviate Nguyen et al. (2019) and Rappoport and Shamir (2019) by N19 and R19.

3.1 | Study design in the original papers

In the following, we outline and compare the study designs that are used to assess the performance of PINSPlus and NEMO in their respective original papers and that meet the inclusion criteria of our experiment (see Table 2 for an overview). We also report the authors' justifications for the design choices. For this purpose, we will also refer to T. Nguyen et al. (2017), which propose PINS, the predecessor method of PINSPlus, and to Rappoport and Shamir (2018), a benchmark study intended as neutral that has been previously conducted by the authors of NEMO. All results of NEMO's competing methods originate from this benchmark study, that is, the results of NEMO were simply added to the results of the previously published benchmark study. Since both R19 and N19 mainly use real datasets to evaluate their methods, we do not further consider the simulation results presented by R19.

Data Both R19 and N19 use datasets from The Cancer Genome Atlas Research Network (TCGA; https://www.cancer.gov/tcga), where each dataset corresponds to a different cancer type (e.g., kidney renal clear cell carcinoma or acute myeloid leukemia). The two author teams also consider the same three types of omic data (gene expression, methylation, miRNA expression) but use different numbers of datasets (34 in N19 vs. 10 in R19). Neither N19 nor R19 explicitly comments on the number of datasets and the selected cancer types, although 34 seems to be close to the maximum number of available datasets for the three considered types of omic data at the time of publication. Moreover, neither N19 nor R19 discusses

TABLE 2 Overview of the study design components used for performance assessment of PINSPlus and NEMO.

Study design component		PINSPlus (Nguyen et al., 2019)	NEMO (Rappoport & Shamir, 2019)
Datasets	Number and type	• 34 TCGA datasets (gene expression, methylation, miRNA expression) • *2 METABRIC datasets (gene expression, CNV)	• 10 TCGA datasets (gene expression, methylation and miRNA expression) • *Partial TCGA datasets
	Preprocessing (all methods)	See Table S2	See Table S2
Competing methods	Number and type	3: SNF, iCluster+, Consensus Clustering	9: PINS, SNF, iClusterBayes, *k*-means, spectral clustering, MCCA, LRAcluster, *rMKL-LPP, *MultiNMF
	Preprocessing (method-specific)	See Table S2	See Table S2
	Other method parameters	• SNF: *alpha* = 0.5, *no. iterations* = 10, *number of clusters* = estimated according to eigen-gaps, *maximum number of clusters* = 5, *number of neighbors* = 20 • Other methods: see the original paper	• SNF: *alpha* = 0.5, *number of iterations* = 30, *number of clusters* = estimated according to rotation cost, *maximum number of clusters* = 15, *number of neighbors* = number of samples/10 • Other methods: see the original paper
Evaluation criteria	Metric	• Survival: logrank test	• Survival: permutation-based logrank test • Clinical: permutation-based /Kruskal–Wallis test (discrete/continuous) for up to six clinical variables • *Runtime • *Number of clusters
	Aggregation	• Number of datasets with significant and most significant logrank *p*-value	• Number of datasets with significant logrank *p*-value • Number of datasets with at least one enriched clinical variable • Mean $-\log_{10}$ logrank *p*-value • Mean number of enriched clinical variables

Note: Included are only components (i) for which the corresponding results are presented as figures or tables in the main paper (i.e., not in the supplement), (ii) that compare the method of interest to other competing methods, and (iii) that correspond to the performance assessment based on real data. In addition, some components are not included in the experiment, which are indicated by asterisks (*). Competing methods and evaluation criteria for datasets not included in the experiment are not shown.

their choice of omic data types, which seems to be a general issue in papers proposing new cancer subtyping methods, as criticized by Duan et al. (2021).

Although the 10 cancer types included by R19 are also considered in N19, the corresponding datasets have different numbers of patients and omic variables. This is mainly caused by the different preprocessing steps applied by N19 and R19 (see Supporting Information Section A.2 for details). In addition, the two papers probably also use different dataset versions (it is not possible to identify the data version used by N19).

Note that N19 also considers two breast cancer datasets that do not originate from TCGA and exhibit different omic types. However, we exclude them from our experiment since some evaluation criteria of R19 require six clinical variables (see below), which are either not available or cannot be clearly identified for these two datasets. Moreover, we do not include the partial datasets (i.e., datasets where some patients do not have any measurements for one or more omic data types) used in R19 to demonstrate NEMO's ability to analyze this type of data. This is because PINSPlus assumes complete data and would require potentially suboptimal solutions such as imputation.

Competing methods R19 and N19 use different numbers and types of competing methods to assess the relative performance of their proposed methods. While R19 uses nine competing methods, N19 only considers three methods. The only method that is included in both papers is similarity network fusion (SNF; Wang et al., 2014). The difference in the number of competing methods is not surprising given that the performance evaluation of NEMO is, in contrast to PINSPlus, based on a benchmark study with a focus on method comparison itself (Rappoport & Shamir, 2018). Such studies typically aim to compare as many methods as possible to generate comprehensive guidelines for method users. Interestingly, R19 includes PINS, the predecessor method of PINSPlus, as a competing method. PINSPlus itself is not included since it did not exist yet when Rappoport and Shamir (2018) conducted their benchmark study. Concerning the choice of competing methods, Rappoport and Shamir (2018) report that they aim to represent diverse multiomic clustering approaches, and that within each approach they choose widely used methods with available software and clear usage guidelines. N19 refer to their selected competing methods as established subtyping methods.

Regarding the parameter selection of the competing methods, NEMO's authors state in Rappoport and Shamir (2018) that they choose the method parameters following the guidelines given by the authors of the respective method (which involves performing a parameter search if suggested) and construct parameter selection methods by themselves if there are no available guidelines. N19 does not have a comparable statement except for the number of clusters for the method consensus clustering (Monti et al., 2003), which, as stated in T. Nguyen et al. (2017), is determined as suggested by Monti et al. (2003). For SNF, the only method that is considered as a competing method for both PINSPlus and NEMO, N19 and R19 both normalize the omic variables to have a mean of 0 and a standard deviation of 1 (which, as stated in Section 2.1, we consider as a method parameter since it is not applied for all methods in both papers). However, they choose different values for the number of neighbors (20 vs. number of samples/10), the number of iterations (10 vs. 30), the number of clusters (estimate according to eigen-gaps vs. rotation cost), and the maximum number of considered clusters (5 vs. 15). See Table S2 for the method-specific preprocessing steps as well as N19 and R19 for all other parameters of the remaining methods.

Note that we have to exclude two competing methods (rMKL-LPP and MultiNMF) considered by R19 from the experiment since we are not able to run them (see Supporting Information Section A.3 for details).

Evaluation criteria With regard to the evaluation criteria, N19 focuses on the methods' ability to identify clusters with significant survival differences using the logrank test. Note that in this context, the logrank test is equivalent to performing a Cox regression (which is the term used by N19), but we will refer to it as the logrank test since this seems to be the more commonly used term in cancer subtyping methodology. T. Nguyen et al. (2017) note that the same logrank test was also used by the authors proposing SNF (Wang et al., 2014), which can be seen as a justification for their choice. For each method, N19 highlights the datasets with significant (i.e., $p < 0.05$), and most significant (i.e., the smallest significant p-value across all methods) p-values by color.

In R19, the assessment of significant survival differences is also based on the logrank test. In addition, the authors assess "clinical enrichment" by testing the association between the identified clusters and six clinical variables (gender, progression of the tumor, cancer in lymph nodes, metastases, total progression, and age at initial diagnosis), although not all variables are available in each clinical dataset. R19 employs the χ^2-values using a permutation procedure, arguing that in the cancer subtyping context, the χ^2-values, the number of datasets with at least one enriched clinical variable, the mean $-\log_{10}$ logrank p-value, and the mean number of enriched clinical variables per dataset. R19 thus considers four evaluation criteria regarding survival and clinical enrichment. Note that one of these criteria (the number of datasets with significant logrank p-values) is very similar to the criterion used by N19 (the number of datasets with [most] significant logrank p-values), the only difference being the estimation of the p-value (approximation-based vs. permutation-based) and the inclusion of the number of datasets with the most significant p-values as a second-order ranking criterion in N19.

In addition to analyzing survival differences and clinical enrichment, R19 also reports the number of clusters and the runtime of each method. However, we do not consider these criteria in our experiment since the number of clusters has no clear optimal value and runtime is not comparable due to different computational resources.

3.2 | Challenges when conducting the experiment

Reproducibility The results presented in N19 are fully reproducible, except for one p-value of iCluster+. In contrast, the results presented in R19 cannot be exactly reproduced. Besides the two methods that cannot be run at all, the performance results of the remaining methods are slightly different compared to the original paper, especially for the clinical enrichment criteria (the difference between original and reproduced results with regard to NEMO's performance is reported

in Section 3.3). Interestingly, 76 of the 80 clustering solutions (8 methods × 10 datasets) are equal to the clustering solutions provided by Rappoport and Shamir (2018), with two of the remaining four solutions only differing in one and three individuals, respectively. This means that the reproducibility problems (also observed for some of the 76 settings yielding identical clustering solutions) might be caused by the permutation tests. Moreover, the provided code is probably not the exact code used by R19, as indicated by the fact that R19 refers to Rappoport and Shamir (2018) for the code to reproduce the results, but also mention that the implementations for MCCA (sparse multiple canonical correlation analysis), LRAcluster, and *k*-means were slightly changed compared to Rappoport and Shamir (2018).

When reproducing the results, we do not modify the code provided by the authors in a way that would change the results and attempt to use the same `R` and `R` package versions as in the original papers (see Table S1 in the Supporting Information). However, we have to set a different number of cores in some settings and use a different `R` version for running the permutation tests by R19 due to different computational resources (see our code documentation for details), which may contribute to the reproducibility issues.

Crossing the designs Evaluating the performance of PINSPlus and NEMO using each other's datasets, competing methods, and evaluation criteria poses a number of challenges, the most important one being the choice of parameters both for the two methods of interest, PINSPlus and NEMO, and the competing methods. Whenever a method is applied to a new (set of) dataset(s), the method user needs to carefully select its parameters or a corresponding parameter selection method, which of course also applies to our experiment. Since both N19 and R19 use the same three types of omic data from the same source (TCGA), we set the parameters of PINSPlus and NEMO as in their respective original paper, which corresponds to their default parameter setting. Note that we also do not change the range of possible values for the number of clusters, a parameter that can be specified for both methods and is set to $\{2, 3, 4, 5\}$ for PINSPlus and to $\{2, 3, \dots, 14, 15\}$ for NEMO. We also attempt to use the same parameters for the competing methods when applying them to the new datasets. However, for two competing methods of N19 (iCluster+ and Consensus Clustering), the optimal number of clusters has to be selected by the user according to plots generated by the method when run on a specific dataset. When applying these two methods on the datasets by R19, we thus have to manually choose the optimal number of clusters for every dataset, and although we try to imitate the decisions of N19 on their datasets, a clear determination is not always possible (an issue that is also noted by Duan et al., 2021). Moreover, some refinements regarding the method-specific preprocessing steps are necessary for two competing methods of R19 (see Section A.2 in the Supporting Information).

In addition to the choice of method parameters, some challenges arise when applying the evaluation criteria by R19 to the datasets by N19. Specifically, the logrank permutation test by R19 does not converge for some methods on two datasets by N19, resulting in a p-value of 0 in 15 method-data combinations. In these cases, we use the approximation-based logrank p-values. Moreover, clustering solutions resulting from the dataset UCS (N19) are not tested for clinical enrichment (R19) since it only includes one of the six clinical variables ("gender") with only one value ("female").

3.3 | Results

Performance based on the original study design Figure 1A and 1D shows the reproduced performance results of PINSPlus and NEMO based on their original study design. Note that the representation in these figures slightly differs from the original papers to achieve a comprehensive and yet clear summary of the results. The most important difference is that the papers also report the individual performance results for each dataset (we provide the individual performance results in Tables S4 and S5 in the Supporting Information).

When evaluated based on its original design, PINSPlus seems to be clearly superior to the three competing methods. It has the most significant p-values ($p < 0.05$) regarding survival, with 21 of the 25 significant p-values being the smallest across all methods. NEMO also shows good performance in its original study design, although its performance is not as clearly superior to the competing methods as the performance of PINSPlus. It achieves the highest numbers of datasets with significantly different survival and at least one enriched clinical variable (although there are two competing methods that achieve the same number of datasets with clinical enrichment). Moreover, none of the competing methods achieves both a higher mean $-\log_{10}$ logrank p-value and a higher mean number of enriched clinical variables. Only MCCA obtains a higher mean $-\log_{10}$ logrank p-value than NEMO but has a lower mean number of enriched clinical variables. Note that despite the reproducibility issues, both the absolute (i.e., the values of the four evaluation criteria considered by R19) and the relative performance of NEMO (i.e., when comparing these values to the competing methods) correspond to the results shown in the original paper. The only difference affecting the relative performance of NEMO is that in the original paper,

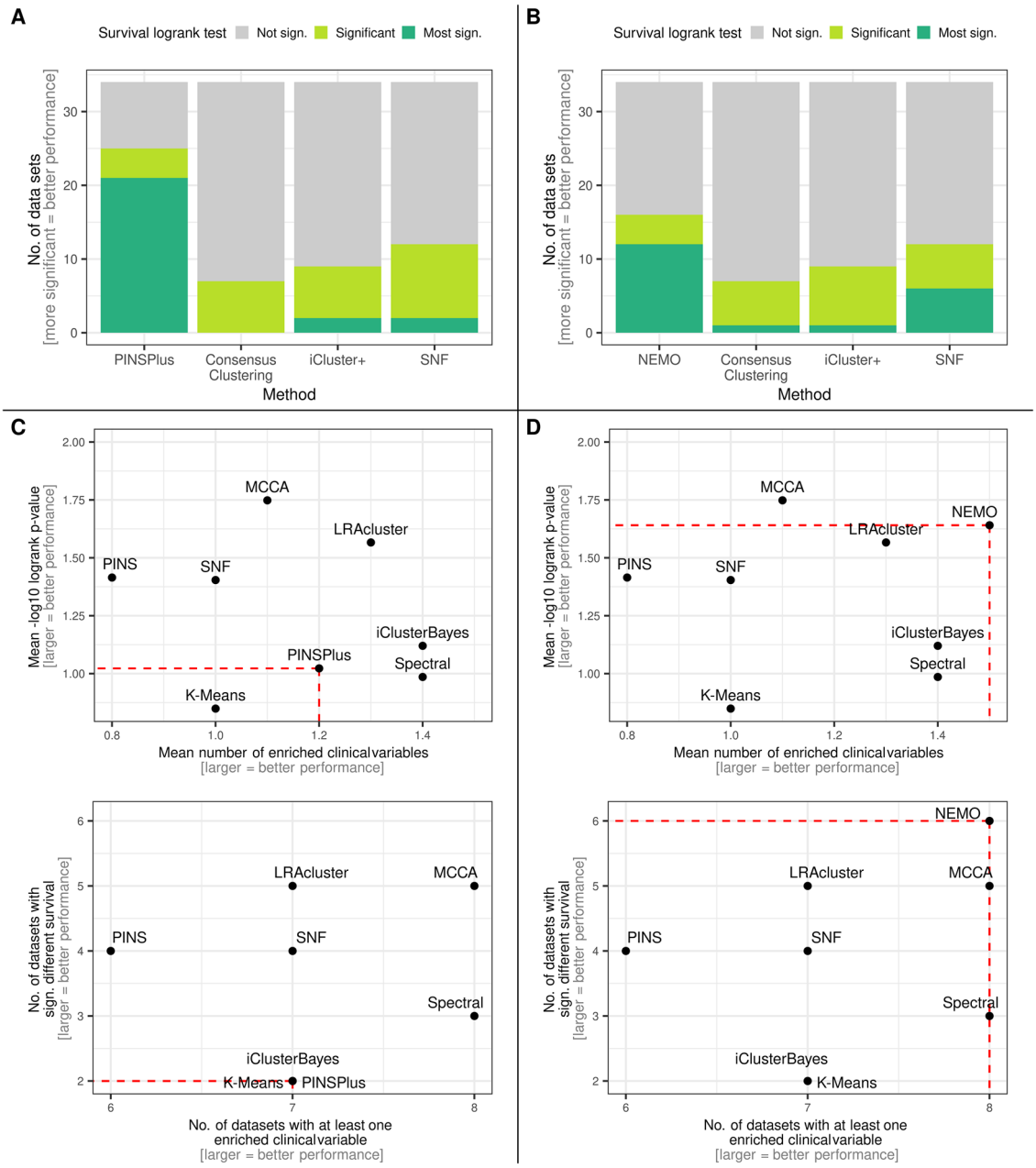

FIGURE 1 Results of the cross-design validation experiment for the cancer subtyping example, where the subfigures A–D correspond to the cells of Table 1. (A) Performance of PINSPlus based on the design by Nguyen et al. (2019) (= original design). (B) Performance of NEMO based on the design by Nguyen et al. (2019) (= crossed design). (C) Performance of PINSPlus based on the design by Rappoport and Shamir (2019) (= crossed design). (D) Performance of NEMO based on the design by Rappoport and Shamir (2019) (= original design).

one of the two methods that could not be reproduced (rMKL-LPP), achieves a higher mean number of enriched clinical variables than NEMO but a lower mean $-\log_{10}$ logrank p-value.

Performance based on the crossed design The performance results of NEMO and PINSPlus based on each others' study design (i.e., datasets, competing methods, and evaluation criteria) are presented in Figure 1B and 1C, respectively. In the study design of R19, PINSPlus does not outperform the competing methods. It is only the fourth and sixth best method with regard to the mean number of enriched clinical variables and mean $-\log_{10}$ logrank p-value, respectively. It belongs to the three worst methods with regard to the number of datasets with significantly different survival and only outperforms PINS, its predecessor method, with regard to the number of datasets with at least one enriched clinical variable. In contrast, NEMO still outperforms the competing methods in the design by N19, although its superiority is not as pronounced as

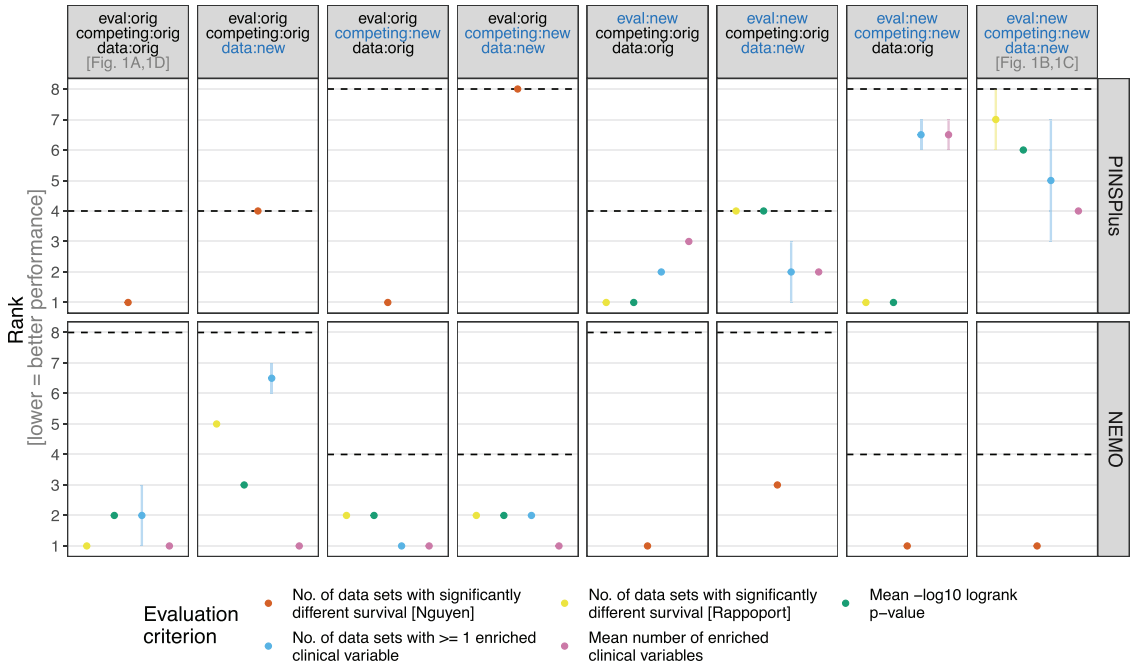

FIGURE 2 Performance ranks of the two cancer subtyping methods PINSPlus and NEMO based on datasets, competing methods, and evaluation criteria that either correspond to the original (PINSPlus: Nguyen et al., 2019; NEMO: Rappoport & Shamir, 2019) or crossed design (PINSPlus: Rappoport & Shamir, 2019; NEMO: Nguyen et al., 2019). Each panel represents the performance rank(s) of PINSPlus or NEMO for one combination of datasets, competing methods and evaluation criteria. If more than one method achieves the same value for a certain criterion, the point represents the average rank and the line indicates the rank range based on minimum and maximum rank. The dashed lines correspond to the number of compared methods, that is, the highest possible rank.

for PINSPlus in the same design (PINSPlus achieves 25 significant p-values while NEMO only achieves 16 for the same 34 datasets).

We also analyze the performance of PINSPlus and NEMO when datasets, competing methods, and evaluation criteria are varied individually. Figure 2 shows the resulting ranks of PINSPlus and NEMO for all eight combinations of the three components, where each component can either be set to the original or the crossed version ($2^3 = 8$). For each criterion (one by N19 and four by R19), a rank of 1 corresponds to the best method. If more than one method achieves the same value for a certain criterion, the minimum, maximum, and average ranks are reported. As can be seen from Figure 2, the ranks of PINSPlus and NEMO generally vary for each combination of datasets, competing methods, and evaluation criterion. Apart from its original design, PINSPlus achieves rank 1 for the evaluation criteria related to survival (i.e., number of [most] significant p-values and mean $-\log_{10}$ logrank p-value) in all combinations where the datasets by N19 are used. However, PINSPlus belongs to the worst performing methods according to survival when applied to the datasets by R19. As mentioned in Section 3.1, the 10 datasets corresponding to different cancer types that are used by R19 are also included in N19. Interestingly, PINSPlus achieves a significant p-value for nine of these 10 datasets in N19, indicating that the difference in performance for these datasets is mainly due to the different preprocessing steps. With regard to the clinical evaluation criteria, PINSPlus seems to have average performance, neither clearly performing better nor worse than the other methods.

In comparison to PINSPlus, the ranks of NEMO are more robust across the different study designs. For six of eight study designs, it achieves rank 1 or 2 for all evaluation criteria (if the minimum or average rank is considered). The only study design where NEMO's performance is considerably worse for two evaluation criteria is the design where only the datasets are taken from N19 while evaluation criteria and competing methods correspond to the original paper. Moreover, it can be noted that while the slightly different calculation of the number of datasets with significant logrank p-values in N19 and R19 does not have an impact on the ranks of PINSPlus, NEMO tends to achieve better ranks for the version of N19. For example, it achieves rank 1 instead of 2, for settings where data and competing methods are by N19. A comparison of approximation-based and permutation-based p-values for all methods and datasets can be found in the Supporting Information file (Figure S1), showing that the approximation-based p-values are indeed generally smaller. The Supporting

Information file also provides a comparison of the two different parameter settings of SNF that are specified by N19 and R19 (Figure S2), which reveals a considerable but nonsystematic performance difference between the two implementations.

4 | DATA ANALYSIS TASK II: DIFFERENTIAL GENE EXPRESSION ANALYSIS

The second data task we consider in our experiment is differential gene expression analysis, which aims at identifying genes that show differences in their expression levels between two or more conditions (Soneson & Delorenzi, 2013). Of the many methods that have been proposed for this task (Seyednasrollah et al., 2013), the more recent ones usually expect RNA-Seq data as input, which means that gene expression is measured as nonnegative counts (Rigaill et al., 2018). The two methods for differential expression analysis included in the experiment are SFMEB (scaling-free minimum enclosing ball) and MBCdeg (derived from MBCluster.Seq, a model-based clustering algorithm for RNA-Seq data), which have recently been proposed by Zhou et al. (2021) and Osabe et al. (2021) and require RNA-Seq data as input. As stated in Section 2, these papers are selected because they make the code to reproduce the results openly available (information on where the code can be found is reported in our code documentation). We will abbreviate them by Z21 and O21 in the following.

4.1 | Study design in the original papers

In this section, we review the datasets, competing methods, and evaluation criteria that are used to assess the performance of SFMEB and MBCdeg in their respective original paper and that meet the inclusion criteria of our experiment (see Table 3 for an overview). We also report the justifications for the design choices provided by the authors. Since Z21 and O21 primarily use simulated data to evaluate their methods, we do not further consider their real data analyses.

Data Both Z21 and O21 generate simulated count data representing RNA-Seq read counts of p genes in $2 \times n_{obs}$ samples from two groups. O21 also simulates count data from three groups, but we exclude these settings from the experiment because SFMEB does not seem to be intended for this type of data (all evaluations in the original paper by Z21 are based on two-group data). The simulation framework of Z21 and O21 is based on different code implementations (code by Robinson & Oshlack, 2010, and `compcodeR` `R` package, Soneson, 2014 vs. `TCC` `R` package, Sun et al., 2013) as well as different distributions to generate the count data (Poisson and negative binomial distribution vs. only negative binomial distribution). Moreover, the two papers choose different numbers of simulation repetitions (20 vs. {50,100}), different sample sizes per group ({1,2,5,8} vs. 3), and different numbers of genes ({15,000,…, 29,800} vs. 10,000).

The simulations also differ with respect to the characteristics of the differentially expressed (DE) genes. In contrast to O21, the DE genes in Z21 include uniquely expressed (UE) genes (u_1, u_2) that have zero counts in groups 1 or 2, respectively. Moreover, Z21 and O21 consider different proportions of DE genes ({0.3,…,0.7} excluding UE genes vs. {0.05,…,0.75}), different $\log_2$ fold-changes between the groups (i.e., the true $\log_2$ ratio of expression change; ≥ 2 vs. 2), and different proportions of upregulated genes (i.e., genes having higher expression) in group 1 ({0.6,…,1} vs. {0.5,…,1}).

In contrast to O21, Z21 applies prefiltering of the genes (e.g., filtering of genes with mean count ≤ 2) for all methods, although some of their considered methods additionally filter genes internally. Moreover, in some settings, Z21 considers heterogeneous data composed of two datasets with different simulation parameters ($\log_2$ fold-change, number of genes, etc.). In the results included in the experiment, O21 only varies the proportion of DE genes and the proportion of upregulated genes, but in a fully factorial manner which results in $6 \times 4 = 24$ simulation settings. However, it should be noted that O21 also varies other parameters (e.g., the $\log_2$ fold-change) in settings not considered in our experiment since they did not meet the inclusion criteria (e.g., because the corresponding figures are shown in the supplement). In the simulation settings by Z21 that are included in our experiment (15 settings in total), more parameters are varied, but not in a fully factorial manner. More specifically, the 15 included settings originate from five "studies" (each consisting of three settings) with different simulation parameters. Within each study, one simulation parameter is varied (see Table 3).

Understandably, neither Z21 nor O21 provides a justification for every single simulation parameter but often refers to similar parameter values observed in real data. Regarding the choice of the number of simulation repetitions, however, neither of the two papers provides a justification. As criticized by Morris et al. (2019), this seems to be a general issue in papers presenting simulation studies.

Competing methods Z21 compares SFMEB with five competing methods they consider as widely used. Two of these methods are referred to as edgeR and DESeq (Anders & Huber, 2010; Robinson et al., 2009; see below for more details), which closely corresponds to the methods selected by O21 (edgeR and DESeq2; Love et al., 2014). In addition to these

TABLE 3 Overview of the study design components used for performance assessment of SFMEB and MBCdeg.

Study design component		SFMEB (Zhou et al., 2021)					MBCdeg (Osabe et al., 2021)	
Datasets	Number of conditions	Two conditions					Two conditions, *Three conditions	
	Number of settings	15					24	
	Setting names	Study 1	Study 2	Study 3	Study 4	Study 5	Fig. 1 in O21	Fig. 3 in O21
	Distribution	Poisson				Negative binomial	Negative binomial	
	Code based on	Robinson and Oshlack (2010)				`R` package `compcodeR`	`R` package `TCC`	
	Number of repetitions	20					100	50
	Samples per group	1		{2,5,8}			3	
	Total number of genes	16,500	28,800	16,800	29,800	15,000	10,000	10,000
	UE genes	(1000,500)	(1000,500), (800,1500)	(1000,800)	(1000,800), (2000,1000)	–	–	–
	Properties of DE genes (excl. UE)	{0.3, 0.5, 0.7}	0.6 + {0.1, 0.3, 0.5}	0.6	0.6 + 0.4	0.6	{0.25, 0.05}	{0.45,0.55, 0.65,0.75}
	Log_2 fold-change	2	2 + 3	2	2 + 3	≥ 3	2	2
	Properties of upregulated genes	0.9	0.9+ 0.1	0.6	0.9+ 0.1	1	{0.5, 0.7, 0.9, 1}	{0.5, 0.7, 0.9, 1}
	Heterogeneous data	No	Yes	No	Yes	No	No	
	Prefiltering	Mean count ≤ 2				Total count < 1	–	
Competing methods	Type and number	5: edgeR, DESeq/DESeq2, HTN, Library Size, NOISeq					3: edgeR, DESeq2, TCC	
	Standard edgeR	No (binomial test)		No (based on Poisson)		Yes	Yes	
	Other method parameters	See the original paper					See the original paper	
Evaluation criteria	Metric	AUC				AUC (smoothed ROC)	AUC	
	Aggregation	Boxplots					Boxplots	

Note: Included are only components (i) for which the corresponding results are presented as figures or tables in the main paper (i.e., not in the supplement), (ii) that compare the method of interest to other competing methods, and (iii) that correspond to the performance evaluation based on simulated data. In addition, some components are not included in the experiment, which are indicated by asterisks (*). Competing methods and evaluation criteria for datasets not included in the experiment are not shown. In the case of design-implementation gaps, the table refers to the code for reproducing the results.

two methods, O21 also considers the less well-known method TCC (tag count comparison; Sun et al., 2013), arguing that it is not sufficient to compare a newly proposed method to the most commonly used methods (edgeR and DESeq2) as those might not be the ones best suited for the analysis. Moreover, they see TCC as the main alternative to their proposed method since the normalization algorithm used by TCC corresponds to the normalization algorithm used by one version of MBCdeg.

Interestingly, Z21 and O21 use different implementations of edgeR. While the implementation by O21 corresponds to one of the edgeR standard workflows, Z21 uses three different implementations of edgeR across their simulation settings of which only one would be typically considered as edgeR (but still with different parameters than O21), while the other two are only edgeR-like. One reason for this choice is that some simulation settings in Z21 do not have biological replicates (i.e., $n_{obs} = 1$ in each group), for which the standard edgeR implementation yields an error (see Supporting Information Section B.2 for details). Regarding the implementation of DESeq/DESeq2, Z21 actually use both DESeq and DESeq2, although they generally refer to the method as DESeq, the predecessor method of DESeq2. This might be explained by the fact that, similar to edgeR, DESeq2 is not intended for settings without biological replicates and thus yields an error, which is why Z21 uses DESeq in these settings. Note that it has been shown that DESeq and DESeq2 perform differently (Love et al., 2014). Both Z21 and O21 use the same parameters for DESeq2. For the parameters of the remaining methods see Z21 and O21 as well as the referenced code.

Evaluation criteria Both Z21 and O21 assess the methods' ability to correctly identify DE genes using the area under the receiver operating characteristic curve (AUC). They both justify this decision with the fact that the AUC, in contrast to other popular measures, does not require the choice of a threshold value. The AUC takes values from 0 to 1, where 1 corresponds to perfect discrimination of DE and non-DE (i.e., nondifferentially expressed) genes, and 0.5 corresponds to random assignment. However, due to an unfortunate default option in the `R` package used by Z21 to calculate the AUC, the resulting AUC values are 1 minus the correct AUC for some repetitions (see Supporting Information Section B.3 for details). Apart from the different `R` packages used to calculate the AUC, Z21 also employs a *smoothed* ROC curve (receiver operating characteristic curve) to estimate the AUC in some of their simulation settings (study 5), which can lead to slightly different results compared to the nonsmoothed ROC curve. Regarding the aggregation of AUC values across the simulation repetitions, both Z21 and O21 use boxplots.

4.2 | Challenges when conducting the experiment

Reproducibility When reproducing the results presented in O21 and Z21, we do not modify the original code in a way that would change the results, with one exception: We change the number of simulation repetitions from 10 to 20 (i.e., the number reported in the paper) in the code provided by Z21 since the results using 20 repetitions are more similar to the results shown in Z21 (note that we make this change before crossing the designs). As stated in Section 2.3, we also use the same `R` and `R` package versions as in the original papers (see Table S6 in the Supporting Information). However, Z21 does not provide this information, which is why we use the most recent package versions available when conducting the experiment (see our code documentation for the exact version information). The code by Z21 also does not include a random number seed, which we therefore set but which is most likely different from the seed used by Z21. Note that for reproducing the results of Z21, we use their AUC implementation potentially yielding incorrect results, but additionally calculate the correct version.

Based on these modifications, running the code of Z21 and O21 results in very similar but not exactly the same boxplots as shown in the original papers. More specifically, the relative performance of each method is the same in the original and reproduced versions, but some boxplots have, for example, different outliers. For Z21, this relatively high degree of reproducibility is noteworthy considering the fact that the provided code does not include a seed or version information. The only three settings that do not yield similar results are the settings from study 5 by Z21 (the differences between the original and reproduced results are described in Section 4.3). Apart from the aforementioned missing seed and version information, the different results in these settings could be due to the fact that the code might not have been provided in its final version.

Crossing the designs As already stated in the first example on cancer subtyping, conducting the cross-design experiment implies that all considered methods are applied to new datasets (new in the sense that these datasets have not been included in the original paper). It is thus necessary to carefully specify the method parameters of SFMEB, MBCdeg, and all competing methods. Although the simulation settings of Z21 and O21 are less comparable than the real datasets of N19 and R19 in the cancer subtyping example, we nevertheless adopt the parameter values from the original papers because we

consider the risk of running the methods with suboptimal parameter settings to be lower for the parameters used by Z21 and O21 than for parameters selected by ourselves (especially because we select the parameters *before* seeing the results to avoid the risk of favoring one of the methods, as stated in Section 2.3). However, both Z21 and O21 consider more than one parameter value for some methods, and Z21 even uses different methods across the simulation settings (i.e., DESeq and DESeq2). For all methods evaluated in Z21 (i.e., SFMEB and its competing methods), we adopt the parameters from study 5 since they are the most similar to the simulation settings considered in O21 (i.e., nonheterogeneous data, generated using the binomial distribution, with replicates). In all simulation settings of O21 included in our experiment, the authors evaluate two versions of MBCdeg, which are denoted as MBCdeg1 and MBCdeg2 and correspond to two different normalization options. Since MBCdeg1 and MBCdeg2 are also implemented separately in the code, we include both versions in the experiment but decide to focus on MBCdeg2, which was observed to be slightly more stable and accurate in O21, before seeing any results. Although O21 does not vary any other parameters of MBCdeg or the competing methods, we note that the main parameter of MBCdeg that is extensively discussed by O21 might not be ideal for some simulation settings of Z21. We thus conduct a sensitivity analysis using two different values for this parameter (see Section B.4 for details).

Since O21 and Z21 use the same evaluation criterion (i.e., boxplots representing the AUC values of all simulation repetitions), we only reevaluate the performance of SFMEB and MBCdeg on each other's competing methods and data. When crossing the designs, we do not consider the AUC that is based on the smoothed ROC curve used by Z21 in some simulation settings. Of course, we also do not use the incorrectly calculated version of the AUC.

Note that not all design components of Z21 and O21 are compatible. More specifically, the DESeq2 and edgeR implementation in O21 results in an error when applied to the simulation settings without biological replicates by Z21. As stated in Section 4.1, this is because DESeq2 and edgeR are not intended for settings without biological replicates and O21 does not use a (possibly nonideal) workaround solution as done by Z21.

4.3 | Results

Performance based on the original study design Figure 3A and 3D shows the reproduced performance results of SFMEB and MBCdeg2 with an additional dashed line corresponding to the median AUC of the corresponding method of interest overall simulation repetitions. Note that the method labels are adopted from the original papers although the competing methods DESeq and edgeR in Z21 do not exactly correspond to the actual method in some simulation settings as discussed above.

For SFMEB, we show both the reproduced AUC values that are potentially biased towards higher values and the correct AUC values. As stated in the previous section, we only observe a noteworthy performance difference between the reproduced results and the results shown in Z21 for three simulation settings (i.e., study 5). In these settings, two competing methods consistently show better performance in the reproduced version, leading to SFMEB being the second best instead of the best performing method in two settings. However, these differences become irrelevant when looking at the correct AUC results. In fact, only the AUC values of the competing methods are in some settings affected by the incorrect AUC calculation, resulting in SFMEB outperforming its competing methods more clearly than initially claimed by its authors. The performance results of SFMEB based on the corrected AUC values are thus still consistent with the conclusion of Z21 that SFMEB outperforms its competitors in most settings (achieving rank 1 according to median AUC in 13 out of 15 settings).

MBCdeg2 also performs well in its original study design. As noted by O21, the method tends to achieve higher AUC values in the settings with a small (≤ 0.45) proportion of DE genes. In some settings where the proportion of DE genes is ≥ 0.55; however, the method seems to fail, often resulting in AUC values below 0.25 and not being able to outperform any of its competing methods (the same applies to MBCdeg1). O21 discusses the occasional failure of MBCdeg extensively and concludes that the identification of the non-DE gene cluster (which they state to be the key to the proposed framework) fails in these cases, which leads to an incorrect classification of DE and non-DE genes. However, MBCdeg2 generally performs better than the competing method TCC in settings where TCC performs well (the same applies to MBCdeg1). Given the fact that TCC could be expected to outperform other methods since the datasets are generated using the `TCC` `R` package and the normalization algorithm used by TCC was designed for settings with asymmetric (i.e., $\neq 0.5$) upregulation as considered by O21, O21 see this as the main contribution of their study.

Performance based on the crossed design Figure 3B and 3C displays the performance results of MBCdeg2 and SFMEB based on each other's simulation data and competing methods. In the study design of O21, SFMEB generally shows worse

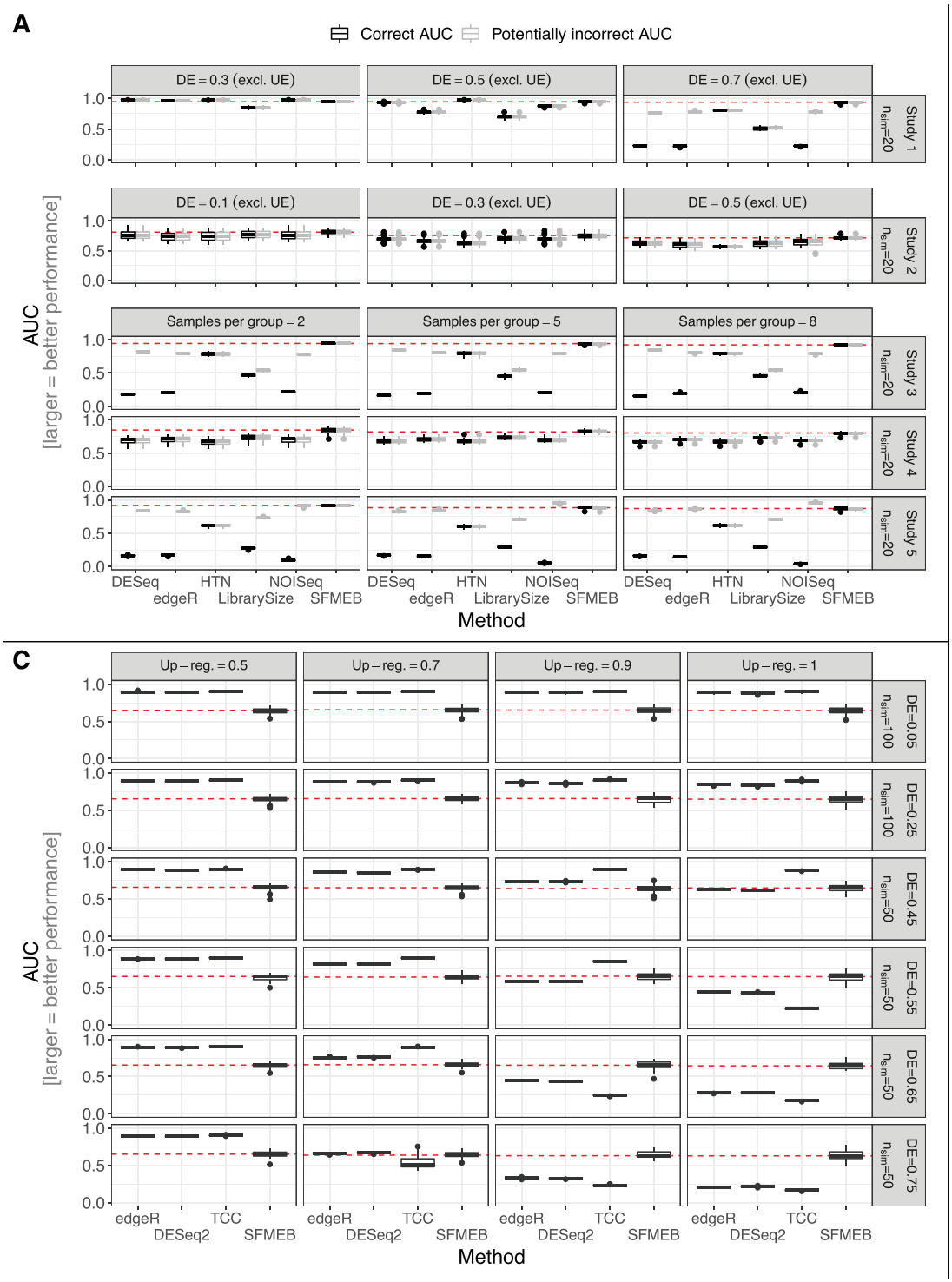

FIGURE 3 Results of the cross-design validation experiment for the differential gene expression analysis example, where the subfigures A–D correspond to the cells of Table 1. (A) Performance of SFMEB based on the design by Zhou et al. (2021) (= original design). (B) Performance of MBCdeg2 based on the design by Zhou et al. (2021) (= crossed design). (C) Performance of SFMEB based on the design by Osabe et al. (2021) (= crossed design). (D) Performance of MBCdeg2 based on the design by Osabe et al. (2021) (= original design). In each subfigure, the boxplots correspond to n_{sim} simulation repetitions, where $n_{sim} \in \{20, 50, 100\}$. The red dashed line corresponds to the median AUC of SFMEB (A and C) and MBCdeg2 (B and D) across all simulation repetitions. In the original paper by Zhou et al. (2021), the AUC has not been calculated as intended by the authors, which is why in subfigure A, both the correct AUC values and the reproduced and potentially incorrect AUC values are provided.

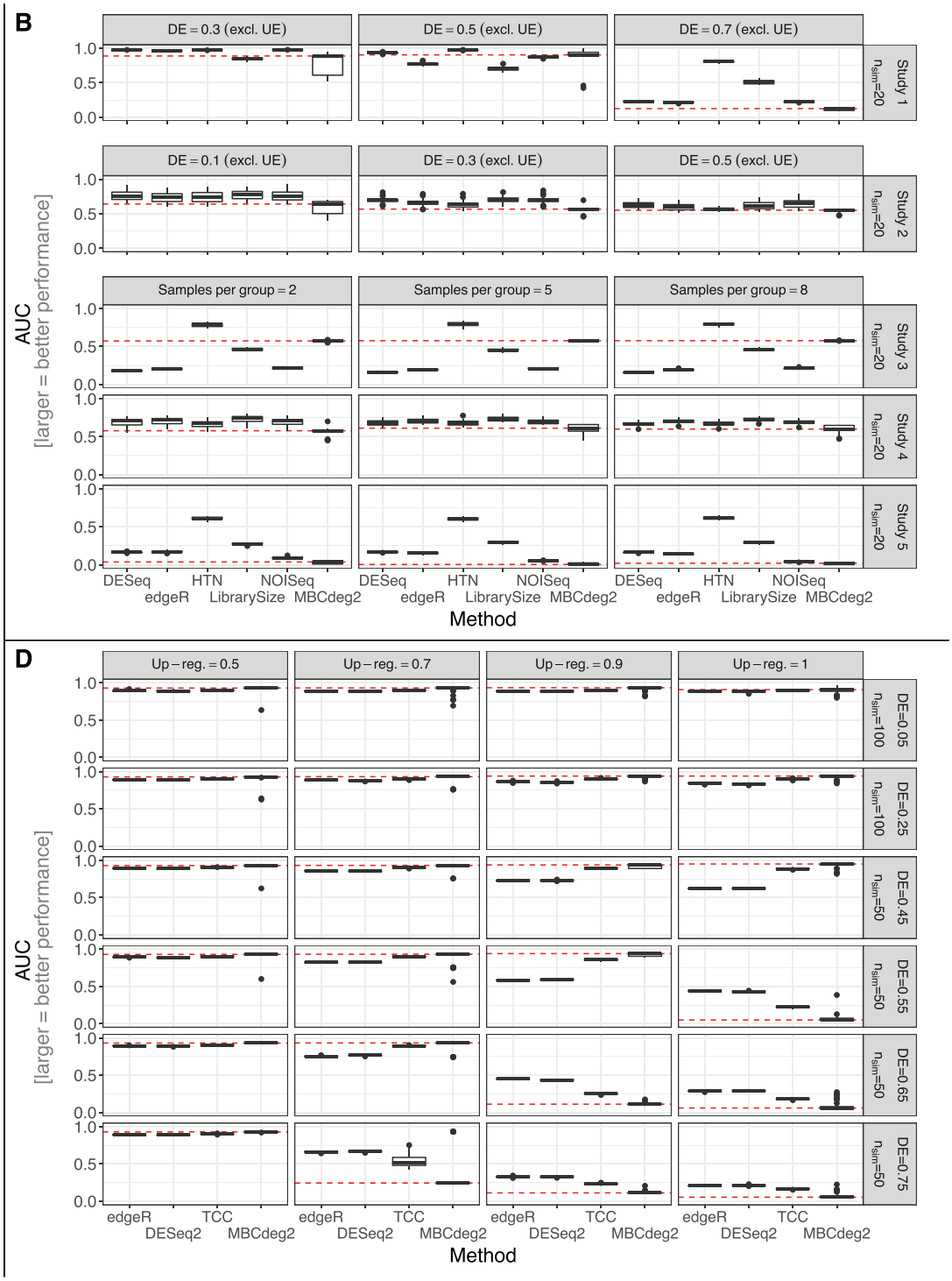

FIGURE 3 Continued

performance than in its original design, having lower median AUC values than all of its competitors in 17 out of 24 settings. However, in five out of the remaining seven settings (the settings with a high proportion of DE genes that are mostly upregulated in one group), SFMEB clearly outperforms the competing methods. Interestingly, this difference in relative performance is mainly caused by the varying AUC values of the competing methods edgeR, DESeq2, and TCC. SFMEB itself, on the other hand, shows very robust AUC values across all settings. However, with a median AUC of about 0.65 in each setting, SFMEB's absolute performance is worse than in the original study, where the lowest median AUC of SFMEB is 0.72.

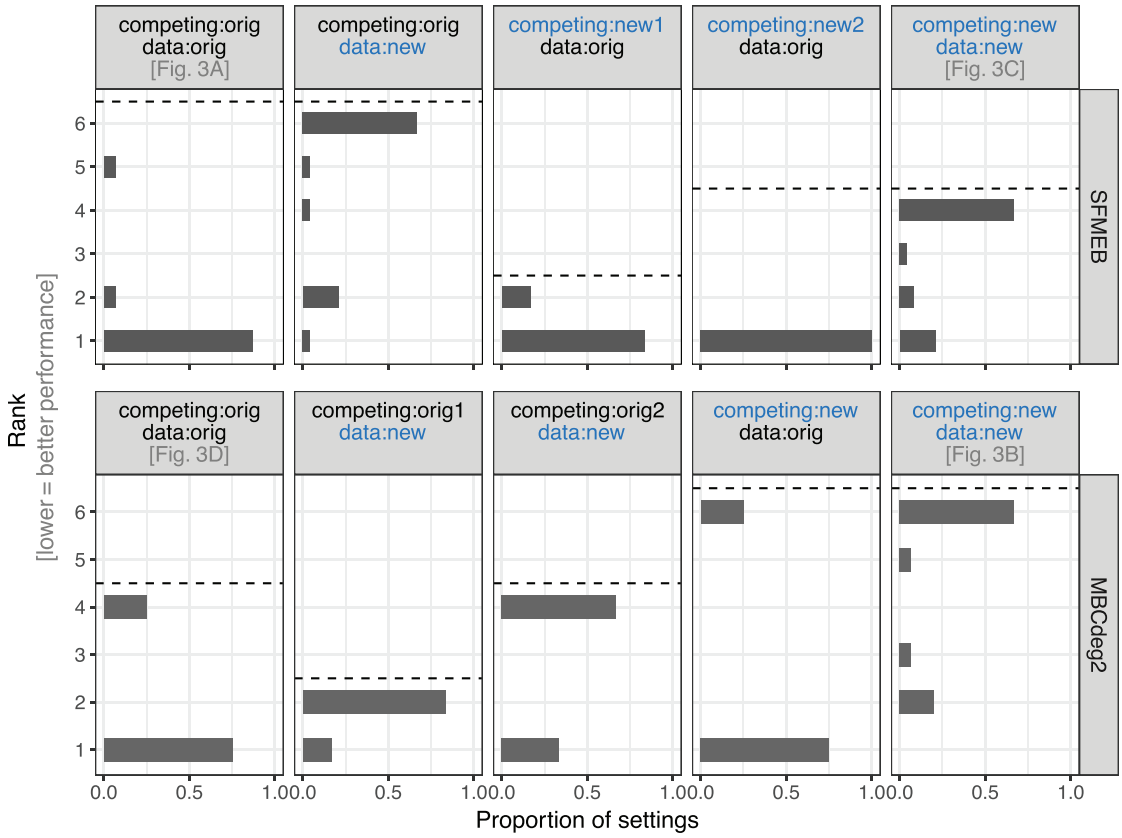

FIGURE 4 Performance ranks of the two differential gene expression analysis methods SFMEB and MBCdeg2 based on datasets and competing methods that either correspond to the original (SFMEB: Zhou et al., 2021; MBCdeg2: Osabe et al., 2021) or crossed design (SFMEB: Osabe et al., 2021; MBCdeg2: Zhou et al., 2021). Each horizontal bar plot shows the performance rank distribution of SFMEB or MBCdeg2 for one combination of datasets and competing methods. The number of ranks that is represented by each bar plot corresponds to the number of simulation settings in the respective data source ($n_{setting} = 15$ for data based on Z21; $n_{setting} = 24$ for data based on O21). The rank of each simulation setting is calculated based on the median AUC value across all simulation repetitions. The dashed lines indicate the number of compared methods, that is, the highest possible rank. Note that for both SFMEB and MBCdeg2, the ranks that result from the combination of competing methods by O21 and data by Z21 are represented by two bar plots due to the incompatibility of two competing methods of O21 with some simulation settings of Z21.

Similar to SFMEB, MBCdeg2 generally performs worse compared to its original design. In 10 out of 15 settings, it is outperformed by all competing methods. However, it is the second best method in four of the remaining five settings (based on median AUC). In contrast to SFMEB, the absolute performance varies more across the settings and only reaches a value comparable to the original design (excluding the settings where the method failed) in two settings. Similar to its original design, there are four settings where MBCdeg2 shows extremely low AUC values, which again seems to be caused by the incorrect identification of the non-DE cluster (note that these are all settings where the proportion of DE genes is ≥ 0.6, which is consistent with O21's observation in the original paper). As stated in Section 4.2, we also conduct a sensitivity analysis where MBCdeg2's main parameter is set to a different value. However, this does not improve the AUC values (see Figure S3 in the Supporting Information).

Figure 4 shows the resulting performance ranks of SFMEB and MBCdeg2 when data and competing methods are varied individually. For both SFMEB and MBCdeg2, the datasets and competing methods can either be set to the original or the crossed version, which results in four ($= 2^2$) different study designs. Note that for both SFMEB and MBCdeg2, the study design that is based on the competing methods of O21 and the data of Z21 are represented by two panels instead of one. This is because two of the three competing methods of O21 cannot be run in some simulation settings of Z21 (see Section 4.2), which makes the resulting ranks incomparable to the ranks that are based on the simulation settings with all three competing methods. Within each study design, the ranks are calculated separately for each simulation setting based on the median AUC and are summarized as bar plots (i.e., each bar plot displays the distribution of 15 or 24 ranks, which corresponds to the total number of settings considered by Z21 and O21, respectively). All AUC values are calculated

correctly. For both SFMEB and MBCdeg2, the performance mainly depends on which simulated datasets are considered. In contrast, using different competing methods has no considerable impact on the distribution of ranks, except that the maximum possible rank reflecting the worst method varies according to the number of competing methods. This is also due to the partial overlap of competing methods between Z21 and O21.

The results of MBCdeg1 based on the crossed design are very similar to the results of MBCdeg2 and can be found in the Supporting Information file (Figure S4).

5 | DISCUSSION

5.1 | Summary of results and limitations

In this paper, we conducted a systematic experiment, which we refer to as "cross-validation of methods" and in which we reevaluated methods based on the datasets, competing methods, and evaluation criteria of a paper proposing a method for the same data analysis task. We considered two exemplary data analysis tasks, namely cancer subtyping using multiomic data and differential gene expression analysis. For each analysis task, we selected two methods, PINSPlus (Nguyen et al., 2019) and NEMO (Rappoport & Shamir, 2019) for cancer subtyping, and SFMEB (Zhou et al., 2021) and MBCdeg (Osabe et al., 2021) for differential expression analysis.

Although we did not conduct our cross-design validation experiment on a large scale, several interesting findings emerged. First, the difficulties in finding eligible papers showed that many papers are still being published without openly available code to reproduce the results. For the papers that were selected, running the provided code did not yield the exact same results as presented in the respective paper. Only the results of PINSPlus were close to being fully reproducible with only one differing p-value in one of the competing methods. Although the lack of reproducibility could be partly due to, for example, our computational resources that were different from those of the authors of the four papers, other potential reasons are that some codes were not provided in their final version and that not all `R`/`R` package versions were reported. The latter is particularly relevant for `R`, which is subject to frequent package updates (potentially causing errors or changing results) and, in contrast to the programming language `Stata`, does not have integrated version control. Nevertheless, the reproduced results of all four methods were consistent with the conclusion of the original papers that the respective method shows good performance.

Second, the experiment concretely illustrated the researchers' degrees of freedom regarding the performance assessment of a method. Notably, all four study designs seemed well though-out and the authors provided justifications in most cases. Interestingly, even for the design components that were similar in both papers, the exact implementation was often different. For example, SNF and edgeR were included as competing methods in both papers of the cancer subtyping and differential expression analysis task, respectively, but were run with different parameters.

Third, the experiment showed how differences in the study design can affect the performance of a method. Three out of the four considered methods (PINSPlus, SFMEB, and MBCdeg) performed worse when assessed on the crossed study design, which seems to be consistent with the general concern that the performance of newly proposed methods is overoptimistic (Boulesteix et al., 2013; Buchka et al., 2021; Norel et al., 2011). Only one method, NEMO, performed well when evaluated on the study design of PINSPlus' original paper and only showed slightly worse performance in some settings where datasets, competing methods, and evaluation criteria were varied individually. For both analysis tasks, using different datasets (real or simulated) had the largest impact on the performance results, which was particularly surprising for the real datasets of the cancer subtyping example where both papers used the same data type and source.

It is important to note that while the findings of our experiment might help to see the performance reported in the original papers from a different perspective, they cannot be seen as evidence of any of the four methods generally having good or bad performance. First, our experiment is limited in the sense that we did not include all study designs and corresponding results reported in the papers, which gives an incomplete picture regarding the study design of the papers and, importantly, the individual strengths and weaknesses of each method. This also includes qualitative evaluation criteria such as PINSPlus' user-friendliness regarding the choice of the number of clusters (which was also noted by Duan et al., 2021), NEMO's simplicity and support of partial data, the avoidance of potential error-prone data normalization when using SFMEB, and the high interpretability of MBCdeg's main parameter. Second, the method performances observed in the experiment clearly depend on (i) our own expertise regarding each method and (ii) the respective new design we reevaluated each method on. The latter is the result of an informal and unsystematic search process based on eligibility criteria (i.e., `R` as a programming language and publicly available code) that could have been specified differently. In

addition, reevaluating each method on more than one new design (i.e., extending the 2×2 table in Section 2 [Table 1] to a $K \times K$ table) could lead to different and more nuanced results.

In addition to the restricted informative value regarding the performance assessment of the considered methods, our experiment is also limited in the sense that the deteriorating method performance observed in three of the four methods cannot be transferred to methodological research in general. This is due to the fact that we only considered two data analysis tasks and, as mentioned above, only included two papers per data analysis task that were selected based on an informal search process.

5.2 | Mechanisms leading to an optimistic performance evaluation and possible solutions

Although the results of our experiment cannot be seen as general evidence for the optimistic performance evaluation of newly proposed methods in methodological research, the experiment itself provides insights into the mechanisms that might explain the observed performance differences. In the following, we will discuss four of these mechanisms, which have either been addressed frequently in the literature or are rarely mentioned in the literature but seem to have been present in our experiment. In addition, we point to possible solutions that can help to avoid large performance discrepancies between original and subsequent studies.

Overfitting of study design to method Our experiment illustrated the many degrees of freedom existing in the assessment of a method's performance. This flexibility can tempt researchers to choose the study design in favor of their proposed method. This may happen both at the planning stage when researchers primarily select a study design in which their method is expected to perform well (e.g., leaving competing methods at their default parameters or simulating data from the model underlying the proposed method), and after seeing the results when they add and/or omit certain design components (e.g., simulation parameters or evaluation criteria; Nießl et al., 2022; Pawel et al., 2022; Ullmann et al., 2023). Focusing on advantageous designs at the planning stage is not necessarily a questionable research practice but becomes problematic if not clearly stated. Changing the study design *after* seeing the results may be legitimate in some cases as far as it is transparently reported, for example, if the originally chosen evaluation criterion turns out to behave inadequately for all methods. But changing the study design *is* bad practice if it is performed in a cherry-picking fashion, that is, excluding or including results depending on whether they convey the expected message or not. The "overfitting" of the study design to the method increases the risk of obtaining different, less optimistic conclusions in a subsequent comparison study in which the authors have less incentives to present the corresponding method in a favorable light.

As already noted by Simmons et al. (2011) in the context of applied research, such optimizations most often do not reflect malicious intent. Instead, they are usually the result of self-serving interpretations of ambiguity convincing honest researchers that the decisions (in our case, regarding the study design) matching their expectations and hopes are the most appropriate ones for various other reasons. These mechanisms are certainly encouraged by publication pressure and publication bias (Boulesteix et al., 2017). Selective reporting after seeing the results can be largely avoided by preregistering study designs and documenting all changes that have to be made subsequently (Morris et al., 2019; Pawel et al., 2022). However, it does not prevent authors from selecting advantageous designs from the start when planning their study. This pitfall could be avoided by adapting the designs from previous studies conducted by different authors. Although designs from different studies might not be suitable to demonstrate all features of the new method, the inclusion of at least one setting that is more "fair" for all compared methods and does not obviously favor the new method (even if the setting is generated by the authors themselves and not adapted from a different study) reduces the risk of overoptimistic conclusions.

For the papers considered in our study, we do not assume that any components regarding the datasets, competing methods or evaluation criteria have been optimized to make the corresponding method of interest appear better than it actually is. On the other hand, we cannot completely rule out this possibility, although it is especially unlikely for NEMO, which was evaluated using a study design adopted from a previously conducted comparison study (Rappoport & Shamir, 2018), similar to preregistration where the design is fixed in advance.

Overfitting of the method to study design Just as the study design can be "overfitted" to the method of interest, the method of interest can also be "overfitted," that is, overoptimized to the study design. This was already noted by Jelizarow et al. (2010) and Ullmann et al. (2023) with a focus on overfitting to the considered datasets. Since method development is, in itself, an optimization process that usually consists of several improvements after seeing the performance results, it is difficult to determine the point where further optimization amounts to overfitting the method to the design used for performance assessment. This not only concerns the method characteristics that are not intended to be changed by the

user but also the parameters that can be set by the method user and whose optimal values for different settings might also be overfitted to the considered study design (Pawel et al., 2022; Ullmann et al., 2023). Note that the issue of overfitting of the method to the study design is relevant for any method evaluation whose results are to be generalized to other evaluation criteria or datasets. This also includes methods that are developed for very specific applications, as long as the authors of the method want it to be used for any other evaluation criteria or datasets than those used for performance assessment (at least for the datasets, this usually seems to be the case).

To avoid overfitting of the method of interest to the study design, it is recommended to evaluate the method extensively. This includes using a large number of datasets and/or simulation settings and several evaluation criteria as well as checking the robustness of the method with respect to small changes in the study design since this makes it more difficult for the method to be artificially optimized (Boulesteix, 2015; Nießl et al., 2022; Norel et al., 2011; Ullmann et al., 2023). In principle, this is comparable to the classical context of regression where overfitting is less likely to occur if the number of observations is large.

Moreover, it may be helpful to reevaluate newly developed methods using a different design after the termination of the trial-and-error process, which might yield slightly worse but likely more realistic performance results (in the sense that the performance discrepancy between original and subsequent papers decreases). Although previous literature usually focuses on evaluating the method on new data (Jelizarow et al., 2010; Norel et al., 2011; Ullmann et al., 2023), considering different competing methods and evaluation criteria could also be reasonable. To reduce the risk of choosing the new design in favor of the proposed method, one could apply the design of a previous study conducted by different authors. As discussed above, the design of a previous study might not be suited to present all features of the proposed method (or even fully match the method's potentially very specific field of application) but this might be less relevant if the design is considered as an additional "external validation design". An external validation design could be, for example, the design of a neutral comparison study, or, similar to our experiment, a previously proposed method (e.g., a method that was included as a competing method). This procedure is only feasible without much additional effort if the authors of the previous paper have made the code for reproducing the results openly available and does not protect against systematic manipulation (e.g., modifying the method after seeing the results and thus consciously biasing the external validation).

When reading a paper, it is typically not possible to identify whether the method of interest has been overfitted to the design used for method development and, unless explicitly stated, if there are any settings that have been separated from the development process. This also applies to the papers included in our experiment, which do not have a corresponding statement. However, MBCdeg is mainly based on an algorithm that was developed by different authors for a different analysis task (i.e., clustering of genes that have already been identified as differentially expressed), which means that this part of the method cannot be overfitted to the design of Osabe et al. (2021).

Different levels of expertise While the mechanisms discussed above are mostly attributed to the nonneutrality of the authors proposing their new method, there are also other potential mechanisms leading to deteriorating performances in subsequent papers. One of them originates from the fact that, as already noted by Duin (1996), the performance of a method is not just dependent on the design it is evaluated on but also on the skill of the person who applies the method. The difference in performance between original and subsequent papers may therefore also be due to the lower expertise level of the subsequent authors whose parameter choice when applying the method to the new data is likely to be less optimal than the parameters that the authors of the method would have selected. Of course, the degree to which the performance deteriorates due to the lack of expertise may be different for each method (Boulesteix et al., 2017) and also depends on how much the new design in which the method is applied differs from the design of the original paper.

As described in Sections 3.2 and 4.2, we also faced the challenge of choosing appropriate method parameters when applying the methods of our experiment to the new datasets and we cannot rule out that these decisions might have led to a worse performance than if the authors of the original papers had chosen the parameters themselves. In the first example on cancer subtyping, we note that although the datasets in both papers had the same data type and originated from the same source, the authors of NEMO and PINSPlus might have set different parameters (including method specific preprocessing steps) for their respective method since the datasets have a different distribution of samples and omic variables (due to the different preprocessing steps and number of datasets). For example, the authors of PINSPlus might have normalized the data when applying it to the datasets of NEMO. The same applies to the differential expression analysis example, where we decided to set SFMEB's parameters for the crossed simulation data as in the simulation setting of the original paper that seemed to be the most similar to the new simulation. It is possible that the authors of SFMEB who are experts in this method might have used a different parameter setting. For MBCdeg, we also cannot rule out that our low level of expertise has contributed to the deteriorating performance of the method. Although we evaluated different values for one parameter of MBCdeg as a sensitivity analysis, we only did that to a limited extent and the considered values may still

be suboptimal (e.g., the authors did not specify how the parameter should be set in the presence of uniquely expressed genes, which are not considered in their simulation settings). It also has to be noted that we are nonexpert users for many of the competing methods used for each paper, and, for instance, the performance of Consensus Clustering and iCluster+ (competing methods of PINSPlus) is certainly dependent on the expertise level of the user since the optimal number of clusters has to be specified manually based on different types of plots and is thus very subjective. However, the difference in expertise (i.e., comparing our expertise vs. the expertise of the authors of the four papers) is probably less drastic with regard to the competing methods than for the methods of interest and is thus not of equal relevance.

One possibility to avoid the systematic deterioration of performance in subsequent studies due to a lower level of expertise is to involve the authors of the method in the respective study (Boulesteix et al., 2017; Morris et al., 2019; Pawel et al., 2022). This can be realized if they implement their method themselves, as done, for example, in the study by Zapf et al. (2021) that involved the authors of all considered methods as co-authors or in benchmark studies that are organized as challenges such as the DREAM challenges (https://dreamchallenges.org/). Alternatively, the authors of a method can be contacted to make sure that their method is implemented correctly as done in the comparison study by Herrmann et al. (2021). However, while the authors of a method could potentially be involved in the majority of comparison studies that assess their method, they will not be able to verify the correct implementation of their method in every *applied* study. Although there is value in studying the performance of a method when used by an expert, it might thus be even more important to assess the performance when it is applied by nonexperts (Boulesteix et al., 2017; Duin, 1996), as we did in this experiment. Note, however, that even among the nonexperts of a method, there are different levels of expertise—or a different willingness to gain expertise by getting more familiar with the method (which may apply in particular to authors that use the method as a competitor for their own method).

In general, it might thus be advisable for authors to make the performance of their method less dependent on user expertise by providing high-quality method documentation that includes a description of all method components and parameters, concrete guidelines on how to choose optimal parameter values in different applications, and ideally also tutorials that help users to become more familiar with the method (Bokulich et al., 2020). If feasible, method authors can also implement automated parameter selection, which protects against the above-mentioned tendency to leave method parameters of competing methods at default values. Moreover, reporting the robustness of the method performance with respect to different parameter values (as done by all four papers considered in the experiment) allows method users to gain an understanding of which parameters need to be carefully specified (Ullmann et al., 2023). Of course, reducing the effect of different levels of expertise also requires efforts from the authors of subsequent papers who need to consider the available guidelines and information on how to set the method parameters.

Different fields of application An insight we gained from the experiment that seems to be rarely addressed in the literature but plays an important role in the optimistic performance evaluation of newly proposed methods is related to the appropriate field of application of a method and its individual strengths within this field. If a method performs worse in a subsequent paper, this can indeed be due to the mutual overfitting of method and design or the lack of expertise, as discussed above. However, the deteriorating performance could also be explained by the fact that the field of application of the subsequent study does not exactly match the field of application the method is intended for. Unfortunately, our experiment suggests that it is often hard to assess if this is the case.

For example, although NEMO and PINSPlus obviously have the same *general* field of application (i.e., cancer subtyping using multiomic data), it was clear that PINSPlus, in contrast to NEMO, is not intended to be used on partial multiomic datasets (i.e., datasets where some patients do not have any measurements for one or more omic data type), which is why we excluded them from our experiment. On the other hand, PINSPlus was initially (i.e., in its original paper) only evaluated based on its ability of finding subtypes that have significantly different survival while NEMO was additionally assessed based on the enrichment of certain clinical variables such as the tumor stage. We did not exclude the clinical enrichment criterion, although it could be argued that PINSPlus is only intended for applications where it is relevant to find subtypes with different survival. Similarly, in the differential expression analysis example, we excluded the three-group simulated data used to assess the performance of MBCdeg in the original paper since the authors of SFMEB did not explicitly mention that their method is intended for this type of application. On the other hand, we did not exclude the settings without biological replicates (i.e., n_{obs} =1 in each group) used by the authors of SFMEB from our experiment although the authors of MBCdeg did not explicitly state that settings without biological replicates belong to MBCdeg's field of application (and other popular methods such as edgeR and DESeq2 are explicitly not intended for these settings). Moreover, it is not clear whether MBCdeg can be applied in settings with uniquely expressed genes (i.e., genes with zero counts in one condition), which were included in most settings used to evaluate SFMEB.

These examples show that it is often not clear for method users what the method's exact field of application is, which consequently makes decisions on whether it is appropriate to apply the method to a new design more difficult and subjective. On the other hand, authors proposing a new method cannot be expected to provide an exact definition of the method's appropriate field of application that accounts for every imaginable design, and some authors explicitly state that the method simply requires more evaluation in certain designs to assess whether they belong to the method's appropriate field of application. For example, the authors of MBCdeg mention that their method still needs to be evaluated on additional simulation frameworks and real data with different experimental settings and organisms.

In general, authors proposing a new method should thus try to study and report its field of application as comprehensively as possible, which, in addition to guidelines for choosing adequate method parameters discussed above, we also consider an important part of the method documentation. As a means to this end, authors should investigate their method's performance in relation to the properties of the included datasets instead of focusing on its overall performance (Strobl & Leisch, 2022). On the other hand, authors using the method in a subsequent study should carefully check whether the application of the method is appropriate and ideally point to differences in the study design.

An issue related to the field of application is that methods often have specific strengths or features *within* their field of application, which is typically reflected by the design and not problematic if reported transparently (as discussed above). However, the method's strengths and special features may not be highlighted to the same extent through the design of the subsequent study (which may be, for instance, selected to highlight the strengths of a different method), thus leading to a deteriorating performance.

We also observed this in our experiment. As mentioned above, a special feature of NEMO is that it can handle missing values in the omic data. However, this feature does not come into play in the study design of PINSPlus, which cannot handle missing values (so that its authors did not consider designs with missing data). Notably, NEMO outperformed the competing methods in the original paper even more clearly for the datasets with missing values than for the full datasets, and although NEMO showed good performance in the design of PINSPlus, its performance might have been even better if the crossed design had also included datasets with missing values. In the differential expression example, the authors of SFMEB emphasize its strength of not requiring data normalization, which is an essential step for most other methods that can mislead downstream analysis if not done correctly. The authors of SFMEB include several data settings where normalization can be error-prone, such as heterogeneous datasets with clearly different fold changes between the conditions. This special strength is, however, not relevant for the settings of MBCdeg that are included in our experiment, which may have also led to SFMEB's deteriorating performance.

In contrast to the mismatch regarding the appropriate field of application discussed above, it is not necessarily inappropriate if a subsequent study disregards the strengths of a method, but it should be ideally mentioned. Note that the discussed mechanisms can also be applied to the competing methods of the original and subsequent papers, whose field of application and specific strengths might be more or less reflected by the study design.

6 | CONCLUSION

Based on the insights gained from the cross-design validation experiment, we conclude that while the discrepancy between original and subsequent studies assessing the performance of a method may be, in part, attributed to the nonneutrality of the method's authors, there are also other mechanisms related to different levels of expertise and fields of application that can contribute to a deteriorating method performance. It is important that both the authors proposing a method and the authors applying the method in a subsequent study acknowledge and counteract these mechanisms. On the side of the method authors, this requires not only a transparent and extensive evaluation but also comprehensive method documentation that enables correct usage by other researchers. In terms of transparency, a minimum requirement for all papers proposing and/or comparing methods should be to openly provide the code, software versions, computational environment, and, if possible, data to reproduce the results. This does not guarantee but at least facilitates the detection of potential overoptimistic statements in the original papers and the nonappropriate use of the methods in subsequent papers. In the long run, these efforts will increase the reliability of studies proposing new methods.

ACKNOWLEDGMENTS

This work was supported by the German Federal Ministry of Education and Research (01IS18036A) and by the German Research Foundation (BO3139/4-3, BO3139/7-1) to ALB. The authors of this work take full responsibility for its content. The authors thank Milena Wünsch for helpful comments and Anna Jacob for language correction.

Open access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors have declared no conflict of interest.

DATA AVAILABILITY STATEMENT

The R code and data to reproduce all results are openly available at https://doi.org/10.6084/m9.figshare.20754028. The R code without data is also available as supplementary material.

OPEN RESEARCH BADGES

This article has earned an Open Data badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. The data is available in the Supporting Information section.

This article has earned an open data badge "**Reproducible Research**" for making publicly available the code necessary to reproduce the reported results. The results reported in this article could fully be reproduced.

ORCID

Christina Nießl https://orcid.org/0000-0003-2425-7858
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947

REFERENCES

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, *11*, 106.

Bokulich, N. A., Ziemski, M., Robeson, M. S., & Kaehler, B. D. (2020). Measuring the microbiome: Best practices for developing and benchmarking microbiomics methods. *Computational and Structural Biotechnology Journal*, *18*, 4048–4062.

Boulesteix, A.-L. (2015). Ten simple rules for reducing overoptimistic reporting in methodological computational research. *PLoS Computational Biology*, *11*, e1004191.

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. (2013). A plea for neutral comparison studies in computational sciences. *PloS ONE*, *8*, e61562.

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*, 138.

Buchka, S., Hapfelmeier, A., Gardner, P. P., Wilson, R., & Boulesteix, A.-L. (2021). On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biology*, *22*, 152.

Duan, R., Gao, L., Gao, Y., Hu, Y., Xu, H., Huang, M., Song, K., Wang, H., Dong, Y., Jiang, C., Zhang, C., & Jia, S. (2021). Evaluation and comparison of multi-omics data integration methods for cancer subtyping. *PLoS Computational Biology*, *17*, e1009224.

Duin, R. P. W. (1996). A note on comparing classifiers. *Pattern Recognition Letters*, *17*, 529–536.

Herrmann, M., Probst, P., Hornung, R., Jurinovic, V., & Boulesteix, A.-L. (2021). Large-scale benchmark study of survival prediction methods using multi-omics data. *Briefings in Bioinformatics*, *22*, bbaa167.

Hofner, B., Schmid, M., & Edler, L. (2016). Reproducible research in statistics: A review and guidelines for the Biometrical Journal. *Biometrical Journal*, *58*, 416–427.

Jelizarow, M., Guillemot, V., Tenenhaus, A., Strimmer, K., & Boulesteix, A.-L. (2010). Over-optimism in bioinformatics: An illustration. *Bioinformatics*, *26*, 1990–1998.

Klau, S., Martin-Magniette, M.-L., Boulesteix, A.-L., & Hoffmann, S. (2020). Sampling uncertainty versus method uncertainty: A general framework with applications to omics biomarker selection. *Biometrical Journal*, *62*, 670–687.

Lohmann, A., Astivia, O. L., Morris, T. P., & Groenwold, R. H. (2022). It's time! Ten reasons to start replicating simulation studies. *Frontiers in Epidemiology*, *2*, 973470.

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*, 550.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, *52*, 91–118.

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*, 2074–2102.

Nguyen, H., Shrestha, S., Draghici, S., & Nguyen, T. (2019). PINSPlus: A tool for tumor subtype discovery in integrated genomic data. *Bioinformatics*, *35*, 2843–2846.

Nguyen, T., Tagett, R., Diaz, D., & Draghici, S. (2017). A novel approach for data integration and disease subtyping. *Genome Research*, *27*, 2025–2039.

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2022). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *WIREs Data Mining and Knowledge Discovery*, *12*, e1441.

Norel, R., Rice, J. J., & Stolovitzky, G. (2011). The self-assessment trap: Can we all be better than average? *Molecular Systems Biology*, *7*, 537.

Osabe, T., Shimizu, K., & Kadota, K. (2021). Differential expression analysis using a model-based gene clustering algorithm for RNA-seq data. *BMC Bioinformatics*, *22*, 511.

Pawel, S., Kook, L., & Reeve, K. (2022). *Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method*. https://doi.org/10.48550/arXiv.2203.13076

Rappoport, N., & Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: Review and cancer benchmark. *Nucleic Acids Research*, *46*, 10546–10562.

Rappoport, N., & Shamir, R. (2019). NEMO: Cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, *35*, 3348–3356.

Rigaill, G., Balzergue, S., Brunaud, V., Blondet, E., Rau, A., Rogier, O., Caius, J., Maugis-Rabusseau, C., Soubigou-Taconnat, L., Aubourg, S., Lurin, C., Martin-Magniette, M.-L., & Delannoy, E. (2018). Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis. *Briefings in Bioinformatics*, *19*, 65–76.

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2009). edgeR: A bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*, 139–140.

Robinson, M. D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, *11*, R25.

Seyednasrollah, F., Laiho, A., & Elo, L. L. (2013). Comparison of software packages for detecting differential expression in RNA-seq studies. *Briefings in Bioinformatics*, *16*, 59–70.

Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E. C., Bahník, Š., Bai, F., Bannard, C., Bonnier, E., Carlsson, R., Cheung, F., Christensen, G., Clay, R., Craig, M. A., Dalla Rosa, A., Dam, L., Evans, M. H., Flores Cervantes, I., … Nosek, B. A. (2018). Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, *1*, 337–356.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*, 1359–1366.

Sonabend, R., Bender, A., & Vollmer, S. (2022). Avoiding C-hacking when evaluating survival distribution predictions with discrimination measures. *Bioinformatics*, *38*, 4178–4184.

Soneson, C. (2014). compcodeR - an R package for benchmarking differential expression methods for RNA-seq data. *Bioinformatics*, *30*, 2517–2518.

Soneson, C., & Delorenzi, M. (2013). A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, *14*, 91.

Strobl, C., & Leisch, F. (2022). Against the "one method fits all data sets" philosophy for comparison studies in methodological research. *Biometrical Journal*. Advanced online publication. https://doi.org/10.1002/bimj.202200104

Subramanian, I., Verma, S., Kumar, S., Jere, A., & Anamika, K. (2020). Multi-omics data integration, interpretation, and its application. *Bioinformatics and Biology Insights*, *14*, 7–9.

Sun, J., Nishiyama, T., Shimizu, K., & Kadota, K. (2013). TCC: An R package for comparing tag count data with robust normalization strategies. *BMC Bioinformatics*, *14*, 219.

Tepeli, Y. I., Ünal, A. B., Akdemir, F. M., & Tastan, O. (2020). PAMOGK: A pathway graph kernel-based multiomics approach for patient clustering. *Bioinformatics*, *36*, 5237–5246.

Ullmann, T., Beer, A., Hünemörder, M., Seidl, T., & Boulesteix, A.-L. (2023). Over-optimistic evaluation and reporting of novel cluster algorithms: an illustrative study. *Advances in Data Analysis and Classification*, *17*, 211–238.

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., & Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, *11*, 333–337.

Zapf, A., Albert, C., Frömke, C., Haase, M., Hoyer, A., Jones, H. E., & Rücker, G. (2021). Meta-analysis of diagnostic accuracy studies with multiple thresholds: Comparison of different approaches. *Biometrical Journal*, *63*, 699–711.

Zhou, Y., Yang, B., Wang, J., Zhu, J., & Tian, G. (2021). A scaling-free minimum enclosing ball method to detect differentially expressed genes for RNA-seq data. *BMC Genomics*, *22*, 479.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2023). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *Biometrical Journal*, 2200238. https://doi.org/10.1002/bimj.202200238

Explaining the optimistic performance evaluation of newly proposed methods: a cross-design validation experiment Supporting Information

Christina Nießl*[1,2], Sabine Hoffmann[1,3], Theresa Ullmann[1] and Anne-Laure Boulesteix[1]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, LMU Munich, Marchioninistr. 15, 81377, München, Germany
[2]Munich Center for Machine Learning (MCML)
[3]Department of Statistics, LMU Munich, Ludwigstr. 33, 80539, München, Germany

A Supporting Information: Cancer subtyping using multi-omic data

A.1 R and R package version information

Table S1 displays the (not necessarily complete) information on the `R` and `R` package versions that are reported in the original paper of PINSPlus and NEMO.

A.2 Data pre-processing

As displayed in Table S2, N19 and R19 use different data pre-processing steps. This includes for example the way missing data are handled or omic variables are selected and normalized. For SNF, the only method that is considered as competing method in both papers, each omic variable (for all three omic types) is normalized to have a mean of 0 and a standard deviation of 1 in N19, while R19 also remove omic variables with zero variance and select the 5000 variables with the highest variance for the methylation data (this is done for all methods in R19). Note that the information on pre-processing shown in Table S2 is based on the published code and, as far as early pre-processing that generates the data provided by the authors is concerned, on the text in the papers. This means that there could have been more pre-processing steps that are not reported. Table S3 shows the resulting number of patients and omic variables for N19 and R19 after applying the pre-processing steps.

As stated in Section 2.1, we consider all pre-processing steps that are performed for *all* methods as belonging to the data component and method-specific pre-processing steps as belonging to the respective methods. However, some refinements are necessary when crossing the designs. More specifically, we note that iClusterBayes and LRAcluster (competing methods of R19) have very long runtimes when run on the data sets of N19. This is because N19 do not perform any variable selection as a general pre-processing step

*e-mail: cniessl@ibe.med.uni-muenchen.de

Table S1: R and R package version information provided in the original papers of the two cancer subtyping methods PINSPlus and NEMO.

	PINSPlus (Nguyen et al., 2019)	**NEMO** (Rappoport and Shamir, 2019)
R version	3.4.3	3.5.0
R package version	Method packages: • ConensusClusterPlus 1.46.0 (Consensus Clustering) • iClusterPlus 1.18.0 (iCluster+) • PINSPlus 1.0.2 (PINSPlus) • SNFtool 2.3.0 (SNF) Other packages: cluster 2.0.7-1 doParallel 1.0.11, entropy 1.2.1, flexclust 1.3-5, foreach 1.4.4, future 1.8.0, iterators 1.0.9, lattice 0.20-35, modeltools 0.2-21, pbmcapply 1.2.4, survival 2.42-3 Biobase 2.38.0, BiocGenerics 0.24.0, codetools 0.2-15, compiler 3.4.3, digest 0.6.15, globals 0.11.0, heatmap.plus 1.3, listenv 0.7.0, Matrix 1.2-14, splines 3.4.3, tools 3.4.3	Method packages: • iClusterPlus 1.16.0 (iClusterBayes) • LRAcluster 1.0 (LRAcluster) • NEMO 0.1.0 (NEMO) • PINSPlus 1.0.1 (PINS) • PMA 1.0.11 (MCCA) • SNFtool 2.3.0 (spectral clustering, SNF) Other packages: No information provided

(only for iCluster+). Hence, when running iClusterBayes and LRAcluster on the data from N19, we select the 2000 omic variables with the highest variance for each omic data type as it is done for k-means, spectral clustering, MCCA, and MultiNMF in R19.

A.3 Reproducibility issues for two competing methods

We have to exclude two competing methods of NEMO (rMKL-LPP and MultiNMF) from the experiment. In the README file accompanying the code of Rappoport and Shamir (2018), the authors state that reproducing the results of rMKL-LPP requires the source code of the method, which they report is only available on request from the authors of rMKL-LPP. It seems that the method can also be run on a web server by now (www.web-rMKL.org), which, however, is not available at the time of writing (last checked in August 2022). Moreover, we have to exclude MultiNMF since running the R code provided by Rappoport and Shamir (2018) (and thus by R19) requires that the user inserts MATLAB commands, which we are not able to specify correctly. Note that Tepeli et al. (2020) were also not able to reproduce the results of MultiNMF shown in Rappoport and Shamir (2018)

A.4 Approximation-based vs. permutation-based p-values

Rappoport and Shamir (2018) note that the χ^2 distribution assumed for the test statistics of the logrank, the χ^2, and the Kruskal-Wallis test is not an accurate approximation for small sample sizes and unbalanced cluster sizes, especially for large values of the test statistic. Hence, Rappoport and Shamir (2018) (and thus also R19) estimate the p-values using permutation procedures, i.e., they randomly permute the cluster labels and calculate empirical p-values as the fraction of permutations for which the test statistic is greater or equal than the test statistic yielded by the original clustering. Rappoport and Shamir (2018) report that they observed large differences between approximation-based (i.e., assuming χ^2 distribution) and permutation-based p-values, with the former yielding increased type 1 errors. They conclude that at least for TCGA data sets, analyses that use approximation-based p-values might not be valid. In our experiment, the approximation-based p-values are indeed generally smaller, as can be seen from Figure S1.

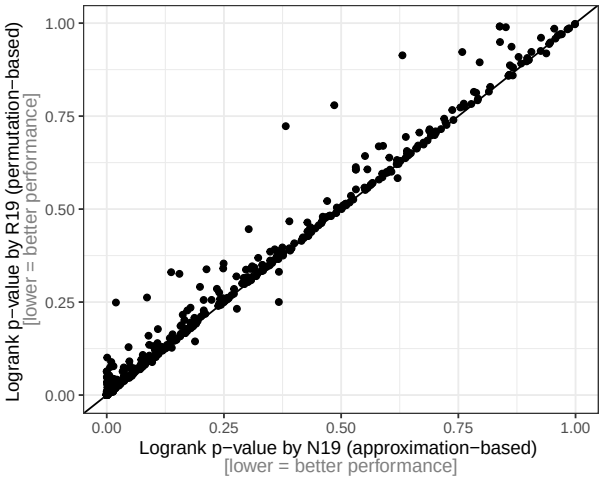

Figure S1: Comparison of approximation-based and permutation-based p-values. Each point refers to the logrank p-value of a method when applied to a data set. All methods and data sets considered by N19 and R19 are included, resulting in 528 points (12 methods $\times$ 44 data sets).

A.5 Reproduced performance results of PINSPlus and NEMO for each data set

Table S4 and S5 display the reproduced results of NEMO and PINSPlus for each data set.

A.6 Comparison of SNF implementations

The cancer subtyping method SNF is used as competing method for both PINSPlus and NEMO. However, N19 and R19 set different method parameters for SNF. Figure S2 shows the logrank p-values and number of enriched clinical variables resulting from the two different implementations, revealing a considerable but non-systematic performance difference.

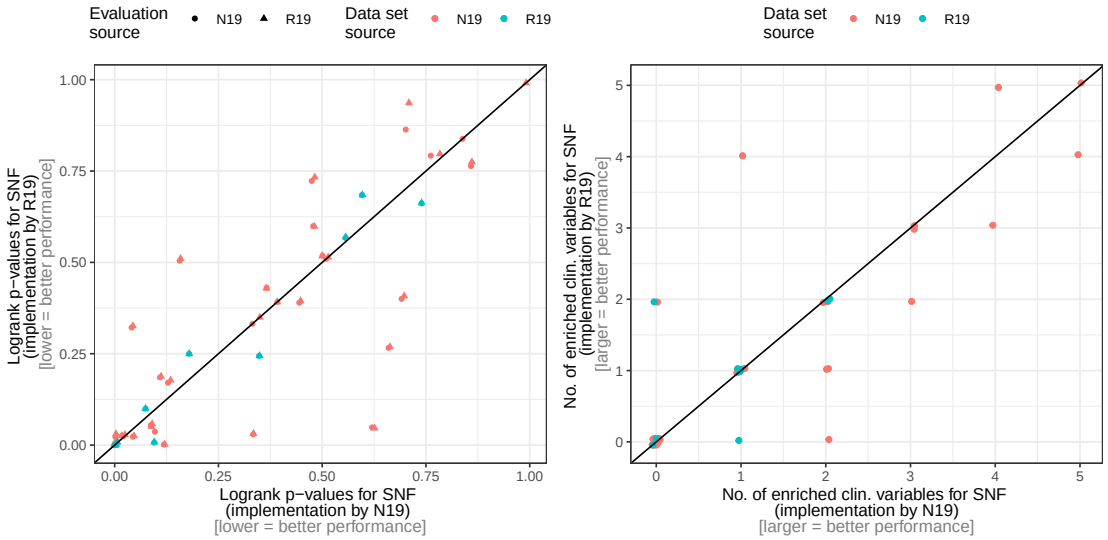

Figure S2: Logrank p-values and number of enriched clinical variables resulting from the two different SNF implementations specified by N19 and R19. The left panel includes 88 points, representing the two different p-value estimation procedures on all 44 (34 +10) data sets considered by N19 and R19. The right panel includes 44 points since the p-values for clinical enrichment are only calculated based on permutation tests.

Table S2: Data pre-processing steps performed in the original papers of PINSPlus and NEMO. Competing methods not included in the experiment are indicated by asterisks (*). In case of design-implementation-gaps, the information shown in the table refers to the code for reproducing the results.

	PINSPlus (Nguyen et al., 2019)	**NEMO** (Rappoport and Shamir, 2019)
Missing values in omic data	• No information on removal and imputation of missing values • Only use patients with observations in all three omics	• Remove patients and omic variables with more than 20% missing values; impute remaining missing values with k nearest neighbor imputation • Only use patients with observations in all three omics
Missing values in survival data	Only include patients with non-missing survival information	Set missing survival times and death info to 0 (these patients can still be used for clinical enrichment)
Remove sample types not corresponding to "Primary solid Tumor" (e.g., "Metastatic")	No information	Yes, except for LAML and SKCM data set
Omic variable pre-processing for all methods	• $\log_2$ transformation for gene expression and miRNA expression	• $\log$ transformation for gene expression and miRNA expression • In all three omics, remove variables with zero variance • For methylation data, select 5000 variables with maximal variance
Method-specific pre-processing	For all three types of omic data: • Normalize variables (mean 0, standard deviation 1): SNF, Consensus Clustering • Subtract median after normalization: Consensus Clustering • Select 2000 variables with max. median absolute deviation: iCluster+ • Remove variables with zero variance: iCluster+	For all three types of omic data: • Normalize variables (mean 0, standard deviation 1) : k-means, spectral clustering, SNF, MCCA, NEMO, *rMKL-LPP • Select 2000 variables with highest variance: k-means, Spectral, MCCA, *MultiNMF

Table S3: Number of patients and omic variables (gene expression, methylation,miRNA expression) after all pre-processing steps (except method-specific pre-processing) have been performed.

	PINSPlus (Nguyen et al., 2019)				**NEMO** (Rappoport and Shamir, 2019)			
Data set	Patients	Gene expression	Methylation	MiRNA expression	Patients	Gene expression	Methylation	MiRNA expression
BRCA	622	239322	363763	2588	621	20226	5000	891
COAD	220	239322	374946	30771	220	19991	5000	613
GBM	273	12042	22833	534	274	12042	5000	534
KIRC	124	17974	23165	590	183	20087	5000	796
LAML	164	16818	22288	552	170	19938	5000	558
LIHC	366	73599	369193	540	367	20153	5000	852
LUSC	110	12042	23348	706	341	20237	5000	878
OV	286	239322	21675	705	287	20174	5000	616
SARC	257	20531	374752	1046	257	20221	5000	838
SKCM	439	20531	373814	586	448	20226	5000	901

Table S4: Reproduced performance results (logrank p-values) of PINSPlus and its competing methods for each data set based on the original study design by N19.

	Data set	PINSPlus	CC	SNF	iCluster+
1	KIRC	6e-05	0.118	0.691	0.058
2	GBM	8.7e-05	0.014	0.021	0.103
3	LAML	0.00087	0.292	0.002	0.083
4	LUSC	0.008	0.688	0.087	0.224
5	BLCA	0.019	0.089	0.109	0.17
6	HNSC	0.046	0.428	0.366	0.364
7	LIHC	0.03	0.622	0.334	0.072
8	STAD	0.002	0.428	0.041	0.434
9	THYM	0.013	0.139	0.097	0.24
10	GBMLGG	7.5e-17	0.00052	4.8e-14	5.4e-14
11	LGG	7.7e-25	2e-06	1.6e-14	2.7e-14
12	PAAD	0.00025	0.013	0.00074	0.00063
13	SKCM	0.048	0.604	0.478	0.108
14	COADREAD	0.003	0.946	0.66	0.178
15	UCEC	0.001	0.105	0.018	0.619
16	CESC	0.03	0.376	0.51	0.201
17	COAD	0.001	0.419	0.128	0.884
18	BRCA	0.007	0.008	0.119	0.046
19	STES	0.007	0.301	0.157	0.46
20	KIRP	1.1e-09	0.367	0.005	0.013
21	KICH	0.028	0.955	0.701	0.788
22	UVM	0.00075	0.005	0.00017	0.003
23	ACC	0.007	0.014	4.3e-05	0.00071
24	SARC	0.03	0.148	0.044	4e-04
25	MESO	0.00073	0.272	0.00042	0.00022
26	READ	0.649	0.737	0.762	0.249
27	UCS	0.458	0.207	0.859	0.983
28	OV	0.319	0.859	0.445	0.062
29	ESCA	0.33	0.791	0.392	0.16
30	PCPG	0.866	0.938	0.332	0.55
31	LUAD	0.099	0.926	0.501	0.118
32	PRAD	0.349	0.638	0.475	0.879
33	THCA	0.166	0.64	0.62	0.111
34	TGCT	0.531	0.758	0.838	0.58

Table S5: Reproduced performance results (number of enriched clinical variables / $-\log_{10}$ logrank p-values) of NEMO and its competing methods for each data set based on the original study design by R19.

	Data set	K-Means	Spectral	LRACluster	PINS	SNF	MCCA	iClusterBayes	NEMO
1	LAML	1/2.9	1/1.9	1/2	1/1.1	1/2.9	1/1.4	1/0.9	1/2.1
2	BRCA	0/0.6	2/1.6	4/1.3	1/1.2	2/1	0/3.2	3/0.2	3/1.4
3	COAD	0/0	0/0.2	0/0.5	0/0	0/0.2	1/0.3	0/0.2	0/0.2
4	GBM	2/2.3	2/2.3	1/1.4	1/3.6	1/4.2	1/1.9	0/1	1/1.9
5	KIRC	0/0.2	0/0.3	0/4.5	0/1.8	1/2.1	1/3.8	1/2	1/1.2
6	LIHC	1/0.2	2/0.4	0/0.8	2/2	2/0.2	2/0.9	2/1	3/3.3
7	LUSC	1/0.2	2/0.3	1/0.9	0/0.3	0/0.6	0/0.4	2/0.6	0/0.4
8	SKCM	2/0.6	2/0.9	3/2.7	1/2.8	1/0.6	2/4.3	3/4.4	3/3.9
9	OV	1/0.1	1/0.8	1/0.6	0/0	0/0.2	1/0.7	0/0	1/0.1
10	SARC	2/1.3	2/1.3	2/1	2/1.2	2/2.1	2/0.6	2/0.8	2/1.8

B Supporting Information: Differential gene expression analysis

B.1 R and R package version information

Table S6 displays the (not necessarily complete) information on the `R` and `R` package versions that are reported in the original paper of SFMEB and MBCdeg. Note that the version of the `ROC` package specified by O21 does not exist, which is why a different version is used in the cross-design experiment.

Table S6: `R` and `R` package version information provided in the original papers of the two differential gene expression analysis methods SFMEB and MBCdeg.

	SFMEB (Zhou et al., 2021)	**MBCdeg** (Osabe et al., 2021)
`R` version	No information provided	3.6.3
`R` package version	No information provided	Method packages: • `MBCluster.Seq` 1.0 (MBCdeg) • `TCC` 1.26.0 (TCC) • `edgeR` 3.28.1 (edgeR) • `DESeq2` 1.26.0 (DESeq2) Other packages: `ROC` 1.6.3, `recount` 1.12.1

B.2 Different edgeR implementations

While the edgeR implementation used by O21 corresponds to one of the edgeR standard workflows, Z21 use three different versions of edgeR, of which only one can be considered as standard edgeR workflow (still using a slightly different version than O21). In six simulation settings, Z21 only use an edgeR-like implementation, which is not based on the negative binomial distribution that is usually considered for edgeR but on the Poisson distribution (presumably, this is done because the counts in these settings are generated using Poisson distribution). Since Z21 also include settings with no biological replicates (i.e., $n = 1$ in each group) where edgeR results in an error, they instead use a testing procedure involving a binomial test. While there are in fact several options suggested by the edgeR user manual (Section 2.12 - *What to do if you have no replicates*) for settings with no biological replicates (although it is stated that these options are not ideal), these do not include the procedure used by Z21. Instead, it is mentioned as an option for technical replicates (i.e., repeated measurements of the same sample that represent independent measures of the random noise associated with protocols or equipment; Blainey et al., 2014).

B.3 Incorrect AUC calculation

Z21 use the `pROC` (Robin et al., 2011) to calculate the AUC. Z21 and O21 use different `R` packages for calculating the AUC, namely `ROC` (Carey and Redestig, 2021) and `pROC` (Robin et al., 2011), respectively. In the `pROC` package, the function that calculates the ROC curve (`roc`) takes the argument `direction`, which determines whether values higher or lower than the threshold should be considered as cases (i.e., DE genes in this context). Per default, the package sets the direction *automatically* according to the medians of the predicted values (see argument `direction` in the `roc` function of the `pROC` manual), which implies

that the ROC curves are biased towards higher AUC values if the `direction` argument is not set explicitly. More precisely, this means that if the automatically defined `direction` argument is not correct, the resulting AUC will be 1 minus the correct AUC. It seems as if Z21 were not aware of this unfortunate default option since they did not explicitly specify the `direction` argument, potentially leading to incorrect AUC values.

B.4 Sensitivity analysis of MBCdeg

The main parameter of MBCdeg (which is based on a clustering algorithm) is the number of clusters K to be found by the method. The number of clusters does not have a default value and is set to $K = 3$ by O21 in the simulation settings that we reproduce in our experiment. This reflects the assumption that there are three gene expression patterns: non-DE genes, DE genes up-regulated in group 1, and DE genes up-regulated in group 2 (where up-regulated in group j again means having higher expression in group j). However, O21 note that for settings where genes that are up-regulated in one group show different degrees of differential expression (i.e., fold-changes), allowing MBCdeg to generate a higher number of clusters could lead to more accurate results. This could apply to the settings of study 2 and 4 considered in Z21, which consist of two data sets with two different $\log_2$ fold-changes (i.e., 2 and 3). As a sensitivity analysis, we thus set $K = 5$ for these settings, reflecting non-DE genes and the two different degrees of differential expression for both groups, which however does not result in higher AUC values (see Figure S3). Moreover, O21 state that for settings where all DE genes are up-regulated in one group, the true number of clusters is actually $K = 2$, reflecting non-DE genes and DE genes (all up-regulated in one group). Since this situation is present for the three settings of study 5 in Z21, we also run MBCdeg with $K = 2$, which, however, does not lead to improved results (see Figure S3).

B.5 Experiment results of MBCdeg1

Figure S4 presents the performance ranks of MBCdeg1, which, in contrast to MBCdeg2 uses the default normalization algorithm.

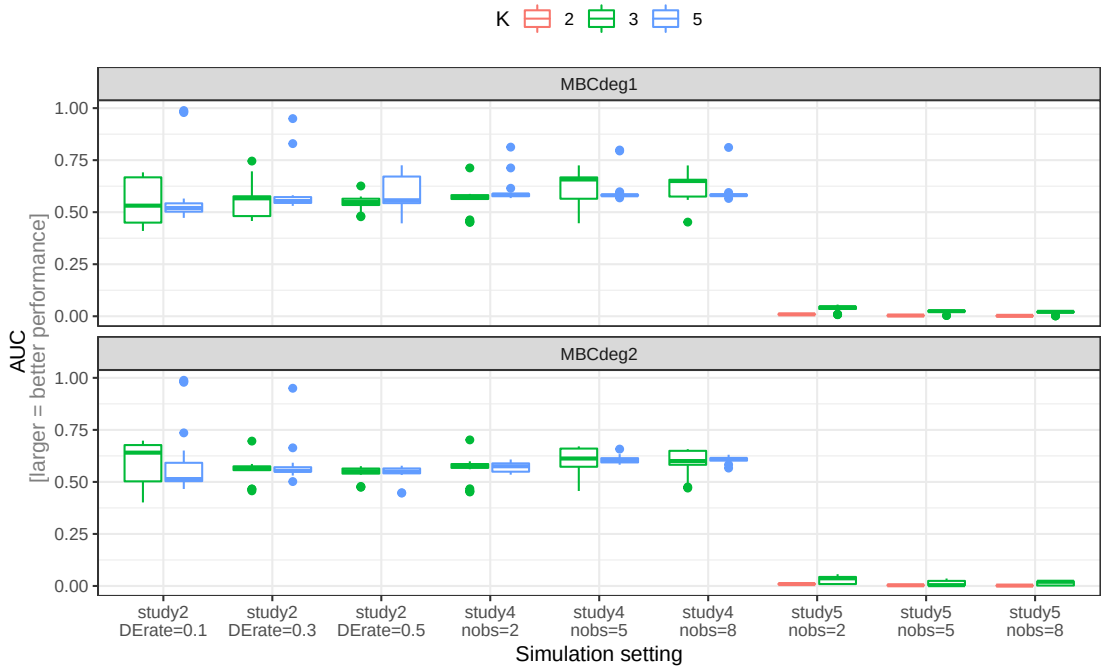

Figure S3: Performance results for MBCdeg1 and MBCdeg2 when using different values for K. $K = 3$ is the value used in the main analysis.

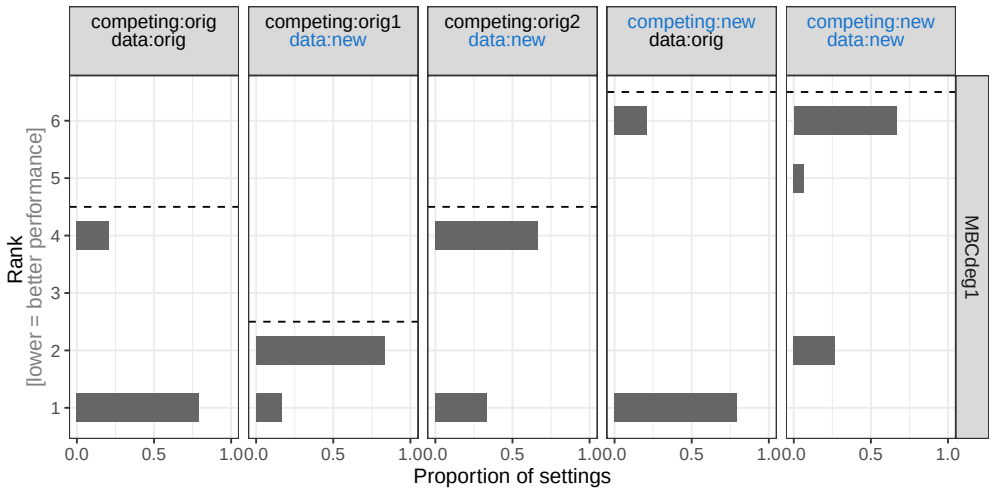

Figure S4: Performance ranks of the differential gene expression analysis method MBCdeg1 based on data sets and competing methods that either correspond to the original (Osabe et al., 2021) or crossed design (Zhou et al., 2021). Each horizontal bar plot shows the performance rank distribution of MBCdeg1 for one combination of data sets and competing methods. The number of ranks that is represented by each bar plot corresponds to the number of simulation settings in the respective data source ($n_{setting} = 15$ for data based on Z21; $n_{setting} = 24$ for data based on O21). The rank of each simulation setting is calculated based on the median AUC value across all simulation repetitions. The dashed lines indicate the number of compared methods, i.e., the highest possible rank. Note that the ranks that result from the combination of competing methods by O21 and data by Z21 are represented by two bar plots due to the incompatibility of two competing methods of O21 with some simulation settings of Z21.

References

Blainey, P., Krzywinski, M., and Altman, N. (2014). Points of significance: Replication. *Nature Methods*, 11:879–880.

Carey, V. and Redestig, H. (2021). *ROC: utilities for ROC, with microarray focus*. R package version 1.62.0.

Nguyen, H., Shrestha, S., Draghici, S., and Nguyen, T. (2019). PINSPlus: A tool for tumor subtype discovery in integrated genomic data. *Bioinformatics*, 35:2843–2846.

Osabe, T., Shimizu, K., and Kadota, K. (2021). Differential expression analysis using a model-based gene clustering algorithm for RNA-seq data. *BMC Bioinformatics*, 22:511.

Rappoport, N. and Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic Acids Research*, 46:10546–10562.

Rappoport, N. and Shamir, R. (2019). NEMO: Cancer subtyping by integration of partial multi-omic data. *Bioinformatics*, 35:3348–3356.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12:77.

Tepeli, Y. I., Ünal, A. B., Akdemir, F. M., and Tastan, O. (2020). PAMOGK: A pathway graph kernel-based multiomics approach for patient clustering. *Bioinformatics*, 36:5237–5246.

Zhou, Y., Yang, B., Wang, J., Zhu, J., and Tian, G. (2021). A scaling-free minimum enclosing ball method to detect differentially expressed genes for RNA-seq data. *BMC Genomics*, 22.

D Contribution 4: "Statistical parametric simulation studies based on real data"

This section is a reprint of:

Sauer*, C., Lange*, F. J. D., Thurow, M., Dormuth, I., & Boulesteix, A.-L. (2025). Statistical parametric simulation studies based on real data. https://doi.org/10.48550/arXiv.2504.04864 (*contributed equally).

Copyright:

Author contributions:

C. Sauer and F. J. D. Lange conceptualized the paper together with A.-L. Boulesteix, drawing inspiration from prior work and experiences shared by M. Thurow and I. Dormuth. C. Sauer and F. J. D. Lange jointly developed the methodology and wrote the R code, with C. Sauer focusing on the design and implementation of the code, and F. J. D. Lange on ensuring its reproducibility and correctness. C. Sauer and F. J. D. Lange wrote the original draft and led the review and editing of the manuscript, incorporating comments from A.-L. Boulesteix, M. Thurow, and I. Dormuth.

Statistical parametric simulation studies based on real data

Christina Sauer*† [1,2,3], F. Julian D. Lange* [1,3], Maria Thurow[4,5], Ina Dormuth[4], and Anne-Laure Boulesteix [1,3]

[1]Institute for Medical Information Processing, Biometry and Epidemiology, Faculty of Medicine, LMU Munich, Munich, Germany
[2]Department of Statistics, LMU Munich, Munich, Germany
[3]Munich Center for Machine Learning (MCML), Munich, Germany
[4]Department of Statistics, TU Dortmund University, Dortmund, Germany
[5]Research Center Trustworthy Data Science and Security, UA Ruhr, Dortmund, Germany

June 2, 2025

Abstract

Simulation studies are indispensable for evaluating and comparing statistical methods. The most common simulation approach is parametric simulation, where the data-generating mechanism (DGM) corresponds to a predefined parametric model from which observations are drawn. Many statistical simulation studies aim to provide practical recommendations on a method's suitability for a given application; however, parametric simulations in particular are frequently criticized for being too simplistic and not reflecting reality. To overcome this drawback, it is generally considered a sensible approach to employ real data for constructing the parametric DGMs. However, while the concept of real-data-based parametric DGMs is widely recognized, the specific ways in which DGM components are inferred from real data vary, and their implications may not always be well understood. Additionally, researchers often rely on a limited selection of real datasets, with the rationale for their selection often unclear. This paper addresses these issues by formally discussing how components of parametric DGMs can be inferred from real data and how dataset selection can be performed more systematically. By doing so, we aim to support researchers in conducting simulation studies with a lower risk of overgeneralization and misinterpretation. We illustrate the construction of parametric DGMs based on a systematically selected set of real datasets using two examples: one on ordinal outcomes in randomized controlled trials and one on differential gene expression analysis.

Keywords: data-generating mechanism, empirical methodological research, Monte Carlo experiments, real-data-based simulation, realistic simulation settings

1 Introduction

In medicine and other disciplines, researchers applying statistical methods are faced with an ever-growing number of options to choose from. To aid them in these decisions and provide well-founded practical recommendations regarding the suitability of a method for a given application, empirical methodological studies, i.e. studies that empirically evaluate and compare statistical

*These authors contributed equally to this work.
†Corresponding author, e-mail: christina.sauer@stat.uni-muenchen.de.

methods, are indispensable. In these studies, the data to which the statistical methods are applied can be divided into two main categories: simulated data and real data. Accordingly, we refer to these studies as simulation studies and real-data studies, respectively, although they may be presented within the same publication, potentially in combination with the introduction of a new method.

The first key distinction between the two study types lies in how the data-generating mechanism (DGM) is determined: While in simulation studies, the DGM(s) must be explicitly specified by the researcher, in real-data studies, the DGM underlying each real dataset is inherently determined by real-world processes (Hothorn et al., 2005). For real-data studies, this shifts the focus from specifying a DGM to the careful selection of appropriate datasets for the study. Although simulation studies may employ semi-parametric approaches, where part of the DGM involves resampling from a real dataset, the most common approach is parametric simulation, where a predefined parametric model is used to draw observations (Morris et al., 2019; Siepe et al., 2024). Focusing for now on parametric simulations, another critical distinction between simulation studies and real-data studies emerges: Unlike real-data studies, where the true DGMs remain unknown and reflect complex real-world processes, simulation studies operate with DGMs that are fully known, as they are explicitly constructed by the researcher.

Based on these two distinctions, simulation studies offer two key advantages over real-data studies. First, the full knowledge of the DGM (often referred to as access to the "ground truth") enables researchers to evaluate the performance of statistical methods with respect to essentially any target of interest, such as a true effect size or the validity of a null hypothesis (Boulesteix et al., 2020; Friedrich & Friede, 2024). In contrast, real datasets typically provide only a limited set of targets for which the truth is known, with prediction tasks (where the target corresponds to the true outcome of an observation) being a notable exception. Second, the full control over the DGM in simulation studies allows researchers to investigate statistical methods under virtually any scenario they wish to explore (e.g., varying parameter values and distributions) and to generate an unlimited number of datasets from the same DGM. In stark contrast to real-data studies, the amount of available data in simulation studies is essentially only limited by the available computational resources (Boulesteix et al., 2020).

Importantly, however, having full control over the DGM, while offering clear advantages, also places a substantial responsibility on the researcher, making it something of a mixed blessing. While this applies to all decisions in the design and execution of a study—commonly referred to as researcher degrees of freedom (RDFs; Simmons et al., 2011)—DGM-related RDFs are particularly impactful, as the choice of the DGM(s) can strongly influence the results of a simulation study and, as a consequence, the recommendations derived from them (Astivia & Zumbo, 2015; Fairchild et al., 2024; Jansen & Holling, 2023; Kulinskaya et al., 2021; Metcalfe & Thompson, 2006; Pateras et al., 2018). To illustrate the mixed blessing of having full control over the DGM, consider a researcher conducting a simulation study to evaluate methods with respect to a specific target in a clinical trial with two treatment groups, a continuous outcome, and under the scenario where a specific assumption Z is violated. While it is advantageous that the DGM can be easily tailored to match the researcher's specific interests (e.g., continuous

outcome, two treatment groups, violation of Z), several parameters (e.g., effect size, sample size) and parts of the model structure (e.g., outcome distribution, presence of covariates, extent of Z's violation) still need to be defined, which is often a challenging process that requires careful consideration.

Practical relevance For simplicity, assume for now that only a single numerical parameter θ remains to be specified. To enhance the generalizability of the simulation findings to its domain of interest, a reasonable approach is to choose values for θ that are practically relevant for that domain. Here, the *domain of interest* refers to the (hypothetical) population of true real-world DGMs to which the simulation study's results and recommendations should apply, and the set of chosen values for θ is *practically relevant* if its distribution closely aligns with that of θ in the real-world DGMs within the domain of interest; this concept can similarly be extended to all components of the DGM. Importantly, having *realistic* DGMs—defined as reflecting *any* real-world DGM—does not automatically ensure practical relevance. Using again the example of a specific parameter θ, practical relevance requires that the distribution of its values aligns with the distribution observed in the DGMs that actually belong to the study's domain of interest. Thus, realistic DGMs are a necessary but insufficient condition for practical relevance. The concept of specifying practically relevant DGMs (or at least realistic ones) aligns with recommendations in the literature, including formal guidelines and related discussions (Boulesteix et al., 2020; Burton et al., 2006; Chipman and Bingham, 2022; Harwell et al., 2017; Paxton et al., 2001; White, 2023). Of course, simulations using intentionally simplistic or unrealistic DGMs also serve important purposes, such as identifying a method's breaking point (Heinze et al., 2024; Morris et al., 2019). However, we argue that for many simulations, researchers implicitly aim for practical relevance, or at least this is what readers are likely to assume unless explicitly stated otherwise.

Assuming an aim of practical relevance, two main issues arise: First, achieving practically relevant DGMs for a given domain of interest is challenging. This is supported by several reviews identifying discrepancies between the DGMs used in simulations and real datasets. For example, when reviewing simulation studies on meta-analyses, Langan et al. (2017) and Fernández-Castilla et al. (2020) found that the number of simulated individual studies often exceeded the number typically observed in existing meta-analyses. Similar discrepancies were found in the context of recurrent events data (Pénichoux et al., 2015), missing data (Guevara Morel et al., 2022), fMRI data (Welvaert & Rosseel, 2014), or the health, educational, and social sciences in general (Bono et al., 2017). Since these reviews treat the discrepancies as criticisms, it is reasonable to assume that the studies aimed for practical relevance. These discrepancies may arise from arbitrary or non-neutral choices, with the latter reflecting often well-intentioned but potentially biased decisions that favor specific outcomes (e.g., demonstrating the superiority of a particular method; Pawel et al., 2024) and thus exploit the RDFs associated with the DGM. Regardless of the reason for the discrepancies, if a simulation study gives the impression of using practically relevant DGMs when, in fact, they are not—even though it is widely understood that such studies inevitably involve simplifications—this can lead to misinterpretation of the findings by readers and possibly even the researchers themselves.

Second, the definition of practical relevance given above inherently relies on a precise specification of the domain of interest, which is usually not clear from the context of the study. For instance, in the example above, assume the researcher has selected a set of DGMs; the domain of interest to which the results are intended to generalize could then lie anywhere between the real-world DGMs exactly corresponding to those considered in the study and all real-world DGMs with a continuous outcome, two treatment groups, and a violation of assumption Z. While Strobl and Leisch (2024) rightly argue that it is nearly infeasible to formally and unambiguously define a domain of interest, failing to specify it entirely is not a better alternative. Without such a specification, the domain of interest may be assumed to be broader than it actually is, which can increase the risk of overgeneralizing the results (Nießl et al., 2024).
To address these issues, a reasonable approach in the analysis of results is to focus on analyzing the relationship between DGM characteristics and method performance (Strobl and Leisch, 2024), as solely considering overall performance can be misleading if the DGMs lack practical relevance and is generally hard to interpret without a clearly defined domain of interest. However, this approach is limited if, for example, most of the values selected for a specific parameter in the DGM do not reflect any real-world DGMs. As a complementary perspective, attention may thus also be directed toward specifying the DGMs themselves.

Real-data-based parametric simulations To ensure that the DGMs reflect any real-world DGM (a necessary condition for practical relevance, as noted above), a natural approach is to base them on real datasets. This can be done in a direct manner by resampling parts of the simulated data from real datasets, which, however, requires transitioning from parametric to semi-parametric simulation approaches, such as Plasmode simulation (Franklin et al., 2014; Schreck et al., 2024). If one wishes to remain within the framework of parametric simulations, specific parameters or parts of the model structure could still be derived from real datasets (as suggested, e.g., by Burton et al., 2006). This approach has been adopted in a number of simulation studies (see reviews by Morris et al., 2019, and Siepe et al., 2024), but the concept is typically implemented differently—both in terms of which parts of the DGM are informed by real data (Friedrich & Friede, 2024) and how directly the data inform these parts, which inherently affects the degree of realism achieved. Thus, while real-data-based simulations reduce RDFs associated with the direct specification of DGMs, they introduce new RDFs related to the process of specifying the DGM based on the real dataset. To our knowledge, in contrast to Plasmode simulation, there is no literature systematically discussing the rationale or implications for different implementations of real-data-based parametric simulation.
In addition, basing simulations on real datasets also creates new RDFs related to the selection of these datasets. In practice, very few datasets are typically used for this purpose. For example, in the reviews by Morris et al. (2019) and Siepe et al. (2024), the real-data-based parametric simulation studies almost always rely on just one or two datasets. The selection of these datasets is rarely justified, often appearing to be one of convenience, and it is usually unclear which domain of interest they are meant to represent. Consequently, while the resulting DGMs might be realistic, they are not necessarily practically relevant—at least not beyond the specific DGMs

underlying the selected datasets. In principle, the criticisms regarding dataset selection are similar to those raised for real-data studies (see, e.g., Herrmann et al., 2024, and references therein). A promising strategy for addressing these issues in real-data-based simulation studies could thus be to adopt what has already been suggested for real-data studies: systematically and transparently selecting datasets by specifying a database and clear eligibility criteria for dataset inclusion (see, e.g., Boulesteix et al., 2017). While this approach does not fully resolve the challenge of formally defining the domain of interest, the selection process can serve as a proxy, providing more clarity to readers about the practical settings in which a simulation study's findings are expected to hold. Additionally, shifting RDFs from the selection of individual datasets to defining a systematic selection process can facilitate more meaningful and neutral decisions, thereby enhancing the practical relevance of the considered DGMs.

Our contribution Overall, while the concept of real-data-based simulations is widely recognized, its specific implementation for parametric simulations has not yet been thoroughly addressed, and the process of selecting real datasets is often not conducted systematically. This paper aims to address these gaps by discussing the possibilities, rationale, and implications of all steps of real-data-based simulations. While our focus is primarily on parametric simulations, the insights provided also apply to the parametric element of semi-parametric simulations. Additionally, the discussion on dataset selection is also relevant for the resampling element of semi-parametric simulations.

The paper is organized as follows: In Section 2, we cover all necessary preliminaries, including the types of DGMs used in simulations and key distinctions in the components of parametric DGMs. In Section 3, we detail the construction of real-data-based simulations, considering both the inference of DGMs from real datasets and the systematic selection of these datasets. In Section 4, we present two empirical examples of parametric simulations based on a systematically selected set of datasets, demonstrating that considering only purely researcher-specified DGMs or relying on a single dataset can lead to an incomplete picture of a method and results that do not generalize well. In Section 5, we provide a structured step-by-step workflow, and we conclude our paper in Section 6.

2 Preliminaries

2.1 DGM types

As outlined in the introduction, this paper focuses on parametric simulation. Hence, we refer to the DGMs employed in this approach as parametric DGMs. DGMs of this type correspond to parametric stochastic models that can be represented in closed form (Morris et al., 2019; Schreck et al., 2024). A given parametric DGM is fully specified by its model structure, which consists of various parts specifying the relationships among variables and the statistical distributions assigned to them, and the numerical parameters that provide the specific values required to fully define the model (e.g., sample sizes or effect sizes). We refer to the model structure (or its parts) and the parameters as the components of the DGM. Importantly, the model structure

inherently determines the set of parameters by outlining the distributions, relationships, and other aspects that require numerical values for full definition. Detailed examples of parametric DGMs are provided in Section 2.3.

Instead of parametric DGMs, simulation studies may also use *semi-parametric DGMs*. As the name suggests, this type of DGM is not fully parametric but includes a non-parametric element in the form of resampling from a real dataset (Schreck et al., 2024). Examples of resampling schemes are simple resampling of observations (with or without replacement) or more advanced methods such as sampling from a smoothed empirical distribution of the dataset estimated via kernel density estimation (see Stolte et al., 2024, for further options). Note that although the term "non-parametric" might suggest the absence of parameters, it refers only to the absence of a predefined parametric form. Researchers still need to specify parameters for the resampling scheme, such as the number of observations to be drawn. A specific implementation of semi-parametric DGMs has become known as Plasmode simulation (Franklin et al., 2014; Schreck et al., 2024), which combines resampling of covariate information from a real dataset (non-parametric element) with an outcome-generating model specified by the researcher (parametric element).

Building on the description of semi-parametric DGMs, one might also consider DGMs that are entirely based on resampling without any parametric element. However, studies that rely solely on resampling without incorporating any parametric element are commonly classified as real-data studies, particularly in the context of prediction tasks (Hothorn et al., 2005). Nevertheless, generating data by resampling from an existing dataset can also be regarded as an approach for simulations (see, e.g., Morris et al., 2019). This is why we briefly address it here as well, even though we categorize resampling-only studies as real-data studies. As noted in the introduction, parametric DGMs are widely used, likely because of two key advantages: access to the ground truth and full control over the DGM. However, when the aim is to utilize real datasets to improve practical relevance, other options may, at first glance, appear even more suitable for this purpose than parametric DGMs. More formally, let $\mathcal{D}$ be a real dataset that is considered given for now, and let $\mathcal{G}$ denote the DGM we aim to specify to closely approximate the true but unknown DGM of $\mathcal{D}$, denoted as $\mathcal{G}^*_{\mathcal{D}}$. While it is then possible to make the parametric DGM real-data-based by deriving it from $\mathcal{D}$ (a process we intentionally leave vague for now in the context of parametric DGMs but will elaborate on in Section 3), generating data by resampling might initially seem like a more natural choice: it is intrinsically real-data-based, not constrained by a parametric model, and therefore generally expected to yield a DGM $\mathcal{G}$ that aligns more closely with $\mathcal{G}^*_{\mathcal{D}}$. At the same time, as already outlined in the introduction, real-data studies face the critical limitation that $\mathcal{G}^*_{\mathcal{D}}$, as stated above, remains unknown. As a result, the set of known targets available for evaluating methods is inherently restricted. In this respect, semi-parametric DGMs represent a promising compromise. For instance, in the case of Plasmode simulations, the non-parametric resampling element allows to preserve complex covariate structures present in the real dataset, while the parametric element (which can also be based on $\mathcal{D}$) offers knowledge of the truth for specific aspects of the DGM, such as the relationship between the covariates and the outcome (Schreck et al., 2024).

Based on these considerations, one might conclude that semi-parametric DGMs should generally be preferred over parametric DGMs for specifying realistic DGMs. However, this conclusion is not universally valid, as it depends on the characteristics of the real dataset $\mathcal{D}$ and the specific procedure used to derive the parametric or semi-parametric DGMs from $\mathcal{D}$. Moreover, there are also arguments in favor of (real-data-based) parametric DGMs over semi-parametric DGMs. First, the non-parametric element of semi-parametric DGMs lacks a closed-form representation, making it more difficult to comprehensively describe or evaluate its plausibility. If undesirable characteristics of the dataset $\mathcal{D}$ (e.g., spurious correlations) are inadvertently incorporated into the DGM, these issues are more likely to go unnoticed in semi-parametric DGMs than in parametric DGMs. Additionally, when multiple real datasets are considered, comparing relevant differences between the resulting DGMs can be more challenging in the semi-parametric case. Second, semi-parametric DGMs face practical limitations related to the accessibility of the real dataset. If the dataset $\mathcal{D}$ is not openly available (though we do not recommend this practice), a parametric DGM based on $\mathcal{D}$ can still be shared, even when the actual dataset $\mathcal{D}$ cannot be disclosed. In contrast, semi-parametric DGMs rely on the availability of the complete dataset for reproducibility. Furthermore, it may sometimes be possible to construct real-data-based parametric DGMs without accessing the original dataset at all (this will be discussed in more detail in Section 3.3). For example, relevant parameters can often be derived from summary tables or similar sources, a convenience that semi-parametric DGMs generally lack.
Although semi-parametric DGMs also merit discussion regarding their specification based on real datasets, this paper primarily focuses on parametric DGMs, as they remain the most commonly used type and, to our knowledge, lack a systematic examination in this respect. At the same time, the discussion on deriving parametric DGMs from real datasets (Sections 3.1 and 3.2) is equally applicable to the parametric element of semi-parametric DGMs, while the considerations for dataset selection (Section 3.3) are partially also relevant for the resampling element of semi-parametric DGMs.

2.2 Differentiating components in parametric DGMs

Before discussing how parametric DGMs can be inferred from real data, we first need to examine their components in more detail. For this purpose, we consider a given parametric DGM $\mathcal{G}$, of which some components were inferred from a real dataset $\mathcal{D}$. Given this setup, we introduce additional differentiations beyond the distinction between model structure and parameters. Specifically, we consider two differentiations: one regarding how the components of $\mathcal{G}$ were specified and another regarding our knowledge of their form or value in the true DGM underlying $\mathcal{D}$. These distinctions apply to both individual parts of the model structure of $\mathcal{G}$ and its parameters.

2.2.1 Specification-based differentiation of components

The first important differentiation concerns how exactly the components of $\mathcal{G}$ have been specified. This differentiation, which should be made by the researcher when planning the study, is essential because it determines how real datasets (here: a single dataset $\mathcal{D}$) are selected and which components are based on real data. Each component of $\mathcal{G}$ (i.e. each specific part of the model structure and each parameter) falls into one of the following three categories:

i. Researcher-specified components of interest: These are components of $\mathcal{G}$ that were explicitly specified by the researcher based on their research question. Conceptually, these components anchor the domain of interest and thus determine the selection of real datasets (together with any constraints on real-data-based components, see below). For example, the hypothetical researcher in the introduction was interested in a setting with two treatment groups, a continuous outcome, and the violation of a specific assumption Z. In this case, the researcher-specified components of interest in $\mathcal{G}$ include the type of outcome variable, the status of assumption Z (both are parts of the model structure), and the number of treatment groups (a parameter).

ii. Researcher-specified components of convenience: These are components of $\mathcal{G}$ that were specified by the researcher but not because they are directly aligned with the domain of interest. While, in general, one might argue that all components that are not of specific interest should be based on real data (see the next category), some might still be reasonably specified by the researcher for practical reasons. This may be because they are assumed to have negligible effects on simulation results, are expected to hold by default, or are impractical to infer from real data. In the example above, such a component could be the distribution chosen for the covariates. Importantly, the choice to directly specify components of the DGM that are not of primary interest is a delicate one and should be justified while considering potential unintended consequences for simulation results.

iii. Real-data-based components: These are components of $\mathcal{G}$ that were not specified by the researcher but instead were inferred from real data. During study planning, these components are not yet specified, but researchers might still impose explicit or implicit constraints on them based on their research question. These constraints, along with the researcher-specified components of interest, help define the domain of interest and, for this reason, are also relevant for dataset selection. For example, consider an effect size parameter. Researchers may explicitly specify a range of interest, directly restricting its possible values. Alternatively, they may restrict the simulation study to a specific context, such as a certain disease type, which will implicitly constrain the possible values of the effect size parameter when inferred from datasets of that disease type.

2.2.2 Knowledge-based differentiation of components

The second differentiation of the components of $\mathcal{G}$ considers whether their true form or value—depending on whether the component is a part of the model structure or a parameter—in the true DGM $\mathcal{G}^*_{\mathcal{D}}$ underlying the real dataset $\mathcal{D}$ is known or unknown. This differentiation is relevant because, in general, unknown components introduce uncertainty that affects both the selection of real datasets and the inference of components from them (the specific issues arising will be discussed in Section 3). For the given DGM $\mathcal{G}$, this means that the real-data-based components were inferred from $\mathcal{D}$ despite their true form or value in $\mathcal{G}^*_{\mathcal{D}}$ being unknown. Meanwhile, for $\mathcal{G}$'s researcher-specified components of interest, which guide dataset selection, it remains uncertain whether their specified form or value truly matches that in $\mathcal{G}^*_{\mathcal{D}}$. Note that the latter implies that the knowledge-based differentiation is relevant not only for real-data-based components but also

for researcher-specified components of interest, highlighting its conceptual independence from the specification-based differentiation.

As in the specification-based differentiation, the knowledge-based differentiation applies to both individual parts of the model structure of $\mathcal{G}$ and individual parameters of $\mathcal{G}$. Since parameters in any parametric DGM are generally defined conditionally on the model structure, applying this differentiation to a parameter of $\mathcal{G}$ requires assuming that at least the relevant part of its model structure matches that of $\mathcal{G}^*_{\mathcal{D}}$. Otherwise, the corresponding parameter (and its true value) would not be meaningfully defined in $\mathcal{G}^*_{\mathcal{D}}$. While we provide examples for both categories below, more detailed examples will be given in Section 2.3.

i. Known components: These are components of $\mathcal{G}$ whose true form or value in $\mathcal{G}^*_{\mathcal{D}}$ is known, either because it can be directly determined from $\mathcal{D}$ or because it is established by external knowledge about the application where $\mathcal{D}$ originates (e.g., study design information). Typically, only a limited number of components belong to this category. For parts of the model structure, examples of known components include variable types, such as whether a variable is dichotomous or continuous. For parameters, examples include the number of treatment groups or the number of observations, as these can be observed directly from $\mathcal{D}$.

ii. Unknown components: These are components of $\mathcal{G}$ whose true form or value in $\mathcal{G}^*_{\mathcal{D}}$ is unknown and can only be inferred from $\mathcal{D}$ with uncertainty. For example, if $\mathcal{G}$ includes a relationship between the outcome variable and covariates, unknown components include the functional form of this relationship (a part of the model structure) and the corresponding effect sizes (parameters). Within unknown DGM components, a further distinction can be made between those that explicitly or implicitly appear in the formulation of the simulation's target and those that do not. For instance, if, as in the previous example, $\mathcal{G}$ includes a relationship between an outcome and covariates, and the simulation aims to compare methods for estimating the effect size, this parameter is directly relevant to the simulation target.

2.3 Notation and examples

When describing the construction of real-data-based DGMs in Section 3, we will start with the researcher-specified components already set, while the real-data-based components remain unspecified. In this section, we introduce the corresponding notation needed for this process and provide example simulation study descriptions that illustrate its use. To simplify the notation, we make the following two assumptions.

First, in many cases, researcher-specified components of interest will not be set to a single option or value. Returning to the hypothetical researcher from the introduction, we have, up to this point, suggested that they are only interested in a DGM with two treatment groups, a continuous outcome, and the violation of a specific assumption Z. However, they may also be interested in DGMs with three or four treatment groups or with a dichotomous outcome instead of a continuous one. This would result in up to six possible combinations. To simplify the discussion, we assume that the researcher-specified components of interest are fixed to a single

option or value at a time. In the example, this means that only one of these six combinations would be considered within a given process of constructing real-data-based DGMs (which will be described in Section 3). Accordingly, in practice, this process would need to be repeated separately for each combination. In contrast to the researcher-specified components of interest, the researcher-specified components of convenience are more likely to be set to a single option or value—particularly when their influence is assumed to be negligible or they are expected to hold by default. However, if such a component is assigned multiple options or values (e.g., because it is impractical to infer from real data but still important to vary), this does not require repeating the full process described in Section 3. Instead, it only requires deciding how these components with multiple options or values will be combined with the later inferred real-data-based components. For simplicity, we assume that all researcher-specified components of convenience are also fixed to a single option or value in Section 3, though we encounter the non-simplified case in the empirical illustrations in Section 4.

Second, while both parts of the model structure and individual parameters can, in principle, be inferred from real data, this paper primarily focuses on parameter inference. Consequently, we assume as a base scenario that only parameters are inferred from real data, meaning that all parts of the model structure are specified by the researcher, either as components of interest or convenience, and regard the case where parts of the model structure are also based on real data as an extended scenario.

Based on these considerations, we introduce notation that will be used in the remainder of this paper. To maintain clarity, we only introduce notation for elements that are directly relevant to the discussion. Since all researcher-specified components are assumed to be set to a single option or value, the base scenario implies a single, fully researcher-specified model structure, which we denote as $\mathcal{M}$. If, in addition, parts of the model structure are inferred from real data, this would lead to multiple possible model structures, and notation for this case will be introduced at the relevant point in the discussion in Section 3. For parameters, which are inherently determined by the model structure, we denote the set of parameters that are intended to be real-data-based as $\boldsymbol{\theta}$, while the set of parameters that are researcher-specified, either as components of interest or as components of convenience, is denoted as $\boldsymbol{\lambda}$. Because researcher-specified components are assumed to be fixed to a single option or value, each parameter in $\boldsymbol{\lambda}$ is set to a single value. Consequently, given $\mathcal{M}$ and $\boldsymbol{\lambda}$, if a single value were inferred for each parameter in $\boldsymbol{\theta}$, this would fully specify a single DGM.

To also integrate the knowledge-based differentiation specifically for parameters into the notation, we denote $\boldsymbol{\theta}_{\text{known}}$ and $\boldsymbol{\lambda}_{\text{known}}$ as the parameters whose true values in the true DGM underlying the real dataset are known, while $\boldsymbol{\theta}_{\text{unknown}}$ and $\boldsymbol{\lambda}_{\text{unknown}}$ are those whose true values are unknown. As explained in Section 2.2.2, this differentiation is relevant for all parameters, regardless of whether they belong to $\boldsymbol{\theta}$ (real-data-based) or $\boldsymbol{\lambda}$ (researcher-specified). Within the category of unknown parameters, we further refine the notation to reflect the distinction between parameters that explicitly or implicitly appear in the formulation of the simulation target and those that do not. Accordingly, we write $\boldsymbol{\theta}_{\text{unknown,target}}$ and $\boldsymbol{\lambda}_{\text{unknown,target}}$ for the former and $\boldsymbol{\theta}_{\text{unknown,other}}$ and $\boldsymbol{\lambda}_{\text{unknown,other}}$ for the latter.

To illustrate the introduced notation, we now present descriptions of four simulation studies, which will also serve as recurring examples throughout the paper. These examples do not specify full study designs but focus on the aim, the model structure of the DGM, and the target/estimand, aligning with the "A", "D", and "E" aspects of the ADEMP framework for simulation studies proposed by Morris et al. (2019). Since the knowledge-based differentiation of parameters may require additional explanation, we explicitly highlight it in these examples. To avoid unnecessary complexity, we assume all parameters to be based on real data and thus initially left unspecified, meaning that they are all contained in $\boldsymbol{\theta}$ (although, since the purpose of the examples is to elaborate on the knowledge-based differentiation, it is not relevant that the parameters are in $\boldsymbol{\theta}$, and we could have just as well assigned values to them, i.e. made them part of $\boldsymbol{\lambda}$, instead). The only exception is the number of groups (K), which we set as a researcher-specified parameter of interest, simplifying the notation, as explicitly formulating the model structure for a general number of groups would be impractical. Since the true value of K can be known in the true DGM underlying a dataset $\mathcal{D}$, we have $\boldsymbol{\lambda}_{\text{known}} = \{K\}$ (while $\boldsymbol{\lambda}_{\text{unknown}} = \varnothing$, as there are no other researcher-specified parameters).
Examples 2–4 are based on actual published simulation studies (references provided below). However, only a single aim, target, and model structure were selected per study (if multiple were present), and in some cases, these were slightly modified or simplified. Additionally, the notation was adjusted to ensure consistency across the examples. A summary of the parameters used in the examples, including their role in the knowledge-based differentiation, is provided in Table 1.

Example 1 (*Example-Ordinal*)

- **Aim**: Evaluation of methods testing the null hypothesis H_0 of no treatment differences in two-arm (i.e. $K = 2$) randomized controlled trials with ordinal outcomes having M categories, in settings where H_0 is false
- **Model structure $\mathcal{M}$ of the DGM**: Each simulated dataset is an $n \times 2$ matrix containing the ordinal outcome $y_i \in \{1, \dots, M\}$ and the treatment assignment $x_i \in \{1, 2\}$ for n individuals, $i = 1, \dots, n$. Half of the individuals ($n/2$) are assigned to each treatment group. For each individual in group $k \in \{1, 2\}$, the outcome is generated by drawing from Multinomial$(1, \boldsymbol{\pi}_k)$, where $\boldsymbol{\pi}_k = (\pi_{1,k}, \dots, \pi_{M,k})$, with $\pi_{m,k} = P(Y = m \mid X = k)$, $m = 1, \dots, M$, and $\sum_{m=1}^{M} \pi_{m,k} = 1$. In addition, since H_0 is false, $\pi_{m,1} \neq \pi_{m,2}$ for at least one $m \in \{1, \dots, M\}$.
- **Estimand/Target**: The null hypothesis $H_0 : \pi_{m,1} = \pi_{m,2}$ for all $m \in \{1, \dots, M\}$

The parameters in $\boldsymbol{\theta}$ in *Example-Ordinal* are the number of individuals, n, the number of ordinal categories, M, and the outcome probabilities for each group, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. The true values of n and M in $\mathcal{G}^*_{\mathcal{D}}$ are known, making them part of $\boldsymbol{\theta}_{\text{known}}$, i.e. $\boldsymbol{\theta}_{\text{known}} = \{n, M\}$. In contrast, the true probabilities $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ in $\mathcal{G}^*_{\mathcal{D}}$ cannot be known. Since the simulation target is to evaluate the null hypothesis $H_0 : \boldsymbol{\pi}_1 = \boldsymbol{\pi}_2$, it follows that $\boldsymbol{\theta}_{\text{unknown,target}} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2\}$. In this example, there are no additional parameters, so $\boldsymbol{\theta}_{\text{unknown,other}} = \varnothing$.

Example 2 (*Example-Survival*, based on the study by Dormuth et al., 2023)

- **Aim**: Evaluation of methods testing the null hypothesis H_0 of no differences in two-arm (i.e. $K = 2$) clinical trials with survival outcome, in settings where H_0 is false
- **Model structure $\mathcal{M}$ of the DGM**: Each simulated dataset is an $n \times 3$ matrix containing the (uncensored or right-censored) survival time $y_i \in \mathbb{R}^+$, the censoring indicator $d_i \in \{0, 1\}$ (with $d_i = 1$ if the event was observed and $d_i = 0$ otherwise), and the treatment assignment $x_i \in \{1, 2\}$ for n individuals, $i = 1, \ldots, n$. Each treatment group contains $n/2$ individuals. For each individual in group $k \in \{1, 2\}$, the observed survival time and the censoring indicator are generated as $y = \min(t, c)$ and $d = \mathbb{1}(t \leq c)$, respectively, with theoretically observable survival time t and censoring time c being drawn independently from $\text{Exp}(\eta_k)$ and $\text{Unif}(0, u)$, respectively. Since H_0 is false, $\eta_1 \neq \eta_2$.
- **Estimand/Target**: The null hypothesis $H_0 : S_1(t) = S_2(t)$ for all $t \in \mathbb{R}^+$, where $S_1(t)$ and $S_2(t)$ are the survival functions of groups 1 and 2

In *Example-Survival*, the parameters in $\boldsymbol{\theta}$ are the number of individuals, n, the event rate parameters η_1 and η_2, and the upper bound of the censoring distribution, u. Similar to the first example, the true value of n is known, i.e. $\boldsymbol{\theta}_{\text{known}} = \{n\}$. This is in contrast to the parameters η_1, η_2, and u, whose true values cannot be known (unless u is explicitly determined by the study design). Since the target considers the survival function, which for group k under the exponential distribution is given by $S_k(t) = \exp(-\eta_k t)$, it follows that $\boldsymbol{\theta}_{\text{unknown,target}} = \{\eta_1, \eta_2\}$, while $\boldsymbol{\theta}_{\text{unknown,other}} = \{u\}$.

Example 3 (*Example-Meta-Analysis*, based on the study by Langan et al., 2019)

- **Aim**: Evaluation of methods to estimate the variance of the true effect sizes (between-study heterogeneity variance) in meta-analyses of studies with two groups (i.e. $K = 2$) and continuous outcomes
- **Model structure $\mathcal{M}$ of the DGM**: Each simulated dataset represents a meta-analysis of n_{study} studies. It is an $n_{\text{study}} \times 2$ matrix containing the estimated effect size $\hat{\delta}_i \in \mathbb{R}$ and its estimated within-study variance $\hat{\sigma}_i^2 \in \mathbb{R}^+$ for the n_{study} studies, $i = 1, \ldots, n_{\text{study}}$. The evaluated methods are applied exclusively to the meta-analysis dataset. However, to generate this dataset, additional study-level data must be simulated for each of the n_{study} studies. For each study i, the true study effect δ_i is drawn from $\mathcal{N}(\delta, \tau^2)$, where δ is the true overall effect and τ^2 is the between-study heterogeneity variance, and a study sample size n_{obs_i} is drawn from $\text{Unif}(u_{\min}, u_{\max})$ and then split evenly into two groups. Outcome values for the individuals in group 1 and 2 are drawn from $\mathcal{N}(\mu_{1,i}, \sigma_{1,i}^2)$ and $\mathcal{N}(\mu_{2,i}, \sigma_{2,i}^2)$, respectively, where $\mu_{2,i} - \mu_{1,i} = \delta_i$ and $\sigma_{1,i}^2 = \sigma_{2,i}^2 = \sigma^2$. Based on the simulated data at the study level, the estimated effect size $\hat{\delta}_i$ and within-study variance $\hat{\sigma}_i^2$ are calculated using Hedges' g.
- **Estimand/Target**: The between-study heterogeneity variance (τ^2)

The parameters in $\boldsymbol{\theta}$ in *Example-Meta-Analysis* can be grouped by study level: At the meta-analysis level, they include the number of studies, n_{study}, the overall effect δ, the between-

Table 1: Summary of all parameters in $\boldsymbol{\theta}$, categorized according to the knowledge-based differentiation of components, in the example simulation studies. Except for the number of groups, which is researcher-specified and set to $K = 2$ in all examples, these constitute the full set of parameters defined by the corresponding model structure $\mathcal{M}$.

Example	**Aim: Evaluate methods for ...**	$\boldsymbol{\theta}_{\text{known}}$	$\boldsymbol{\theta}_{\text{unknown,target}}$	$\boldsymbol{\theta}_{\text{unknown,other}}$
Ordinal	... testing H_0 of no treatment differences in two-arm randomized controlled trials with ordinal outcomes	• n: No. of individuals • M: No. of outcome categories	• $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2$: Outcome probabilities per group	–
Survival	... testing H_0 of no differences in two-arm trials with survival outcomes	• n: No. of individuals	• η_1, η_2: Event rate per group	• u: Censoring upper bound
Meta-Analysis	... estimating the variance of true effect sizes (between-study heterogeneity variance)	• n_{study}: No. of studies	• τ^2: Between-study heterogeneity	• δ: Overall effect • $u_{\min}, u_{\max}$: Range for sample size • $\mu_{1,i}$: Mean for group 1 (per study) • σ^2: Within-group variance
DE-Analysis	... identifying differentially expressed genes between two groups	• n: No. of samples • p: No. of genes	• FC_j: Fold change • p_{DE}: Proportion of DE genes	• μ_j, ϕ_j: Expression mean and dispersion

study heterogeneity variance τ^2, and the parameters $u_{\min}$ and $u_{\max}$, which define the range for the study sample sizes. At the study level, the parameters are the mean for group 1 (for each study i), $\mu_{1,i}$, and the within-group variance σ^2. Accordingly, $\mathcal{G}^*_{\mathcal{D}}$ essentially represents a two-level mechanism, specifying both the generation of study-level data and the meta-analysis dataset. If $\mathcal{D}$ represents a meta-analysis dataset (where each row corresponds to a study), additional datasets for the n_{study} studies summarized in $\mathcal{D}$ would be needed to infer the study-level parameter values. Similar to the previous examples, the true value of n_{study} in $\mathcal{G}^*_{\mathcal{D}}$ can be known, i.e. $\boldsymbol{\theta}_{\text{known}} = \{n_{\text{study}}\}$, while the true values of the remaining parameters cannot. Among the latter, τ^2 represents the target of the simulation, i.e. $\boldsymbol{\theta}_{\text{unknown,target}} = \{\tau^2\}$, while the remaining parameters are included in $\boldsymbol{\theta}_{\text{unknown,other}} = \{\delta, \sigma^2, u_{\min}, u_{\max}, \mu_{1,i} \mid i = 1, \ldots, n_{\text{study}}\}$.

Example 4 (*Example-DE-Analysis*, based on the study by Baik et al., 2020)

- **Aim**: Evaluation of methods for differential gene expression analysis, i.e. methods that identify genes with differences in their RNA-Seq expression levels, in a two-group (i.e. $K = 2$) setting (e.g., cancer vs. normal)
- **Model structure $\mathcal{M}$ of the DGM**: Each simulated dataset is an $n \times (p+1)$ matrix containing the RNA-Seq read count $r_{i,j} \in \mathbb{Z}^{0+}$ for n samples, $i = 1, \ldots, n$, and p genes, $j = 1, \ldots, p$, where the read count represents the gene expression level, with a larger count indicating

higher expression. The matrix also includes the group indicator $x_i \in \{1, 2\}$, and each group contains $n/2$ samples. For sample i and gene j, the read count is generated by drawing from a negative binomial distribution, specifically $\text{NB}(\mu_j \cdot \text{FC}_j, \phi_j)$, $\mu_j, \phi_j \geq 0$, if $x_i = 1$, and $\text{NB}(\mu_j, \phi_j)$, $\mu_j, \phi_j \geq 0$, if $x_i = 2$. Here, the fold change FC_j quantifies the relative change in expression. Among all genes, a proportion p_{DE} is simulated as differentially expressed (DE), with $\text{FC}_j \neq 1$ for those genes, while $\text{FC}_j = 1$ for non-differentially expressed genes.

- **Estimand/Target**: The null hypothesis $H_0 : \text{FC}_j = 1$ for all $j \in \{1, \ldots, p\}$

In *Example-DE-Analysis*, the parameters in $\boldsymbol{\theta}$ are the number of samples, n, the number of genes, p, the mean expression level μ_j and the dispersion parameter ϕ_j for each gene j, $j = 1, \ldots, p$, the proportion of DE genes, p_{DE}, and the fold change FC_j (where $\text{FC}_j \neq 1$ only for DE genes). While the true values of n and p in $\mathcal{G}^*_{\mathcal{D}}$ are known, i.e. $\boldsymbol{\theta}_{\text{known}} = \{n, p\}$, the true values of μ_j, ϕ_j, p_{DE}, and FC_j are not. Since the simulation target is to evaluate the null hypothesis of no differential expression ($\text{FC}_j = 1$) for each gene, $\boldsymbol{\theta}_{\text{unknown,target}}$ includes FC_j and p_{DE}, as p_{DE} represents the proportion of genes where the null hypothesis does not hold. Accordingly, $\boldsymbol{\theta}_{\text{unknown,target}} = \{\text{FC}_j \mid \text{FC}_j \neq 1\} \cup \{p_{\text{DE}}\}$. The remaining parameters, μ_j and ϕ_j, are part of $\boldsymbol{\theta}_{\text{unknown,other}}$, as they are required for generating the data but not directly related to the simulation target, i.e. $\boldsymbol{\theta}_{\text{unknown,other}} = \{\mu_j, \phi_j \mid j = 1, \ldots, p\}$.

3 Constructing real-data-based parametric DGMs

In this section, we provide a detailed discussion on the construction of real-data-based parametric DGMs, along with practical recommendations. We begin with the inference of DGM components from a set of real datasets, first considering the base scenario with a single, fully researcher-specified model structure $\mathcal{M}$, where only the parameters in $\boldsymbol{\theta}$ are inferred (Section 3.1), and then the extended scenario, where parts of the model structure are no longer specified by the researcher and instead also inferred from real data (Section 3.2). For both scenarios, we assume that any researcher-specified parameters $\boldsymbol{\lambda}$ are fixed to a single value. We then address the systematic selection of these datasets (Section 3.3).

3.1 Inferring parameters from a set of real datasets

Given a set of R real datasets (the selection of which will be discussed in Section 3.3), $\boldsymbol{\mathcal{D}} = \{\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(R)}\}$, a model structure $\mathcal{M}$, and a set of Q parameters, $\boldsymbol{\theta}$, which we now consider to be arranged as a vector, i.e. $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_Q)$, there are different approaches for inferring the set of L considered parameter vectors, $\Theta = \{\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(L)}\}$, from the real datasets. Each considered parameter vector $\hat{\boldsymbol{\theta}}_l$, $l = 1, \ldots, L$, contains a single value for each parameter in $\boldsymbol{\theta}$ (i.e. Q values in total). The inference process can be broken down into two steps: in the first step, for each parameter, values are inferred from $\boldsymbol{\mathcal{D}}$, resulting in a set of inferred parameter values for each parameter in $\boldsymbol{\theta}$. In the second step, those sets are mapped to the set of considered parameter vectors, Θ.

The approach that is expected to yield DGMs most closely approximating the true DGMs of the real datasets proceeds as follows: In the first step, for each parameter, a value is inferred from each dataset (i.e. R values are inferred from $\boldsymbol{\mathcal{D}}$, and values are not aggregated across datasets).

In the second step, the inferred values for each parameter are combined per dataset to form the set of considered parameter vectors, with each of the vectors containing the parameter values inferred from one dataset. Since this approach essentially maps each dataset to one of the considered parameter vectors (and thus is equivalent to constructing one DGM per dataset, given $\mathcal{M}$ and $\boldsymbol{\lambda}$), we will refer to it as the one-to-one inference approach. However, one may also deviate from this approach and consider alternative strategies for the two outlined steps of the inference process. An overview is presented in Table 2. In the following, we first discuss the one-to-one inference approach before exploring these alternatives. Note that regardless of the chosen inference approach, the resulting DGMs should be checked for plausibility.

Table 2: Overview of approaches for inferring parameters from a set of real datasets, $\boldsymbol{\mathcal{D}} = \{\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(R)}\}$.

Step of the inference process	**One-to-one approach**	**Deviation**
1. Infer set of parameter values for each parameter θ_q in $\boldsymbol{\theta}$. $\rightarrow \boldsymbol{\theta}'_q$	Direct inference: Use value from each of the R datasets directly. $\rightarrow \boldsymbol{\theta}'_q \coloneqq \hat{\boldsymbol{\theta}}_q = \{\hat{\theta}_q^{(1)}, \dots, \hat{\theta}_q^{(R)}\}$	Aggregated inference: Use information from R datasets in aggregated form to generate A_q values. $\rightarrow \boldsymbol{\theta}'_q \coloneqq \tilde{\boldsymbol{\theta}}_q = \{\tilde{\theta}_q^{(1)}, \dots, \tilde{\theta}_q^{(A_q)}\}$
2. Map sets of inferred values for individual parameters, $\boldsymbol{\theta}'_1, \dots, \boldsymbol{\theta}'_Q$, to set of considered parameter vectors. $\rightarrow \Theta = \{\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(L)}\}$	Combine values of directly inferred parameters per dataset, i.e. $\hat{\boldsymbol{\theta}}^{(r)} = (\hat{\theta}_1^{(r)}, \dots, \hat{\theta}_Q^{(r)})$ for each $\mathcal{D}^{(r)}$. $\rightarrow \hat{\boldsymbol{\theta}}_l$ contains the values from one dataset ($L = R$). $\rightarrow \Theta \coloneqq \{\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(R)}\}$	Combine values of directly inferred parameters across datasets and/or aggregately inferred parameters. $\rightarrow \hat{\boldsymbol{\theta}}_l$ contains not only values from one dataset (typically $L \neq R$). $\rightarrow \Theta \coloneqq \boldsymbol{\theta}'_1 \times \dots \times \boldsymbol{\theta}'_Q$ (for fully factorial design), with $\boldsymbol{\theta}'_q \coloneqq \hat{\boldsymbol{\theta}}_q$ or $\boldsymbol{\theta}'_q \coloneqq \tilde{\boldsymbol{\theta}}_q$

3.1.1 One-to-one inference approach

In the one-to-one approach, each parameter θ_q in the parameter vector $\boldsymbol{\theta}$, $q = 1, \dots, Q$, is first directly inferred from each real dataset $\mathcal{D}^{(r)}$, $r = 1, \dots, R$. For a given parameter θ_q, this results in a set $\hat{\boldsymbol{\theta}}_q$, which contains R values:

$$\hat{\boldsymbol{\theta}}_q = \{\hat{\theta}_q^{(1)}, \dots, \hat{\theta}_q^{(R)}\}. \tag{1}$$

We refer to this way of inferring the values of a given parameter θ_q from $\boldsymbol{\mathcal{D}}$ as direct inference. To subsequently map the sets $\hat{\boldsymbol{\theta}}_1, \dots, \hat{\boldsymbol{\theta}}_Q$ to the set of considered parameter vectors, Θ, in the one-to-one approach, these directly inferred parameter values are combined per dataset, forming a full parameter vector $\hat{\boldsymbol{\theta}}^{(r)} = (\hat{\theta}_1^{(r)}, \dots, \hat{\theta}_Q^{(r)})$ for each dataset $\mathcal{D}^{(r)}$. The set of considered parameter vectors therefore contains $L = R$ considered parameter vectors:

$$\Theta = \{\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(L)}\} = \{\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(R)}\} =: \Theta^{\text{one-to-one}}. \tag{2}$$

This way of mapping the sets of inferred values for the individual parameters to the set of considered parameter vectors corresponds to a scattershot design where each $\hat{\boldsymbol{\theta}}^{(r)}$ contains potentially distinct values (Siepe et al., 2024).

Given a model structure $\mathcal{M}$, the one-to-one approach results in R DGMs, $\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(R)}$, each

corresponding to a specific dataset and individually parameterized by $\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(R)}$ (and $\boldsymbol{\lambda}$). Since each dataset is essentially treated individually in the one-to-one inference approach, the following considerations focus on the direct inference of parameters from a single real dataset $\mathcal{D}^{(r)}$. For this dataset, we denote its true underlying DGM as $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ and the corresponding true model structure as $\mathcal{M}^*_{\mathcal{D}^{(r)}}$. To explore the direct inference of parameters $\boldsymbol{\theta}$ from $\mathcal{D}^{(r)}$, we apply the categorization of parameters introduced in Section 2.3, distinguishing between $\boldsymbol{\theta}_{\text{known}}$ and $\boldsymbol{\theta}_{\text{unknown}}$, with a further differentiation between $\boldsymbol{\theta}_{\text{unknown,target}}$ and $\boldsymbol{\theta}_{\text{unknown,other}}$.

Parameters in $\boldsymbol{\theta}_{\text{known}}$ As illustrated by the four example simulation studies, parameters whose true values in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ can be known typically specify quantities such as the number of variables, the number of categories within a variable, or the number of observations in the DGM. While it is straightforward to infer their true value in $\mathcal{D}^{(r)}$ (assuming that the corresponding parameter exists in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$), these parameters often have the least need to be explicitly real-data-based (i.e. to be included in $\boldsymbol{\theta}_{\text{known}}$ rather than $\boldsymbol{\lambda}_{\text{known}}$), as researchers are likely to choose reasonably realistic values even without relying on specific datasets. Additionally, parameters such as the number of observations are often chosen as rounded, tidy values (e.g., $n \in \{10, 50, 100\}$). Using values that deviate from these conventions might confuse readers or make results harder to interpret. Nevertheless, unintentionally unrealistic values can arise. As mentioned in the introductory section, Langan et al. (2017) observed that the number of studies used in meta-analysis simulations was often unrealistically large compared to real-life meta-analyses. Therefore, having parameters in $\boldsymbol{\theta}_{\text{known}}$ can be reasonable, and while directly inferring their values from real datasets may not be necessary, they can still be based on aggregated information derived from the real datasets (see Section 3.1.2).

Parameters in $\boldsymbol{\theta}_{\text{unknown}}$ As discussed in Section 2.3, the true values of $\boldsymbol{\theta}_{\text{unknown}}$ in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ are not known and can only be estimated from $\mathcal{D}^{(r)}$. This estimation process introduces several challenges, which we now address in detail. We begin by discussing the general challenges of estimating parameters in $\boldsymbol{\theta}_{\text{unknown}}$, before highlighting the additional considerations specific to $\boldsymbol{\theta}_{\text{unknown,target}}$.

A straightforward approach to estimate the parameters in $\boldsymbol{\theta}_{\text{unknown}}$ is to apply a maximum likelihood (ML) estimation method, using the given model structure $\mathcal{M}$ as a basis. This approach relies on the assumption that $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ align, at least for the parts relevant for estimating $\boldsymbol{\theta}_{\text{unknown}}$—an assumption that may not hold in practice, particularly if $\mathcal{M}$ is overly simplistic. While perfect alignment between $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ is likely rare, even a close alignment may suffice in many cases. However, substantial mismatches can make the dataset $\mathcal{D}^{(r)}$ unsuitable for parameter estimation, resulting in estimates that differ considerably from their true values or become effectively meaningless. To address this issue, estimation methods that account for and correct potential deviations between $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ can be employed. For instance, in *Example-Meta-Analysis*, the model structure $\mathcal{M}$ assumes a normal distribution for the study effect sizes ($\delta_i \sim \mathcal{N}(\delta, \tau^2)$). However, in $\mathcal{M}^*_{\mathcal{D}^{(r)}}$, the true distribution of effect sizes may deviate from normality, which can result in biased estimates of the between-study heterogeneity vari-

ance τ^2. This issue can be mitigated by employing estimators that do not rely on the normality assumption (e.g., the Sidik-Jonkman estimator; Sidik and Jonkman, 2005). In other cases, the mismatch between $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ may be so pronounced that it cannot be reasonably addressed, leading to essentially meaningless parameter estimates. For example, in *Example-Survival*, if the distribution of the (theoretically observable) survival times in $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ deviates substantially from an exponential distribution, the resulting parameters would fail to capture the characteristics of $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ and no longer serve a meaningful purpose in the simulation study. Similarly, looking beyond the four examples from Section 2.3, estimating regression coefficients for p covariates $(\beta_1, \ldots, \beta_p)$ in a linear regression model specified by $\mathcal{M}$ becomes problematic if the relationship between the outcome and covariates is, for example, strongly non-linear. In such cases, it may be necessary to revise the model structure $\mathcal{M}$ or select real datasets $\boldsymbol{\mathcal{D}}$ based on criteria that ensure a better alignment with $\mathcal{M}$. While we treat both $\mathcal{M}$ and $\boldsymbol{\mathcal{D}}$ as fixed here, these considerations are addressed in Sections 3.2 and 3.3.

Even when $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ align closely, additional challenges arise due to the finite nature of $\mathcal{D}^{(r)}$, which represents a sample generated by $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ and is thus subject to sampling variability. Consider, for example, *Example-Ordinal*, where the true probabilities $\pi_{m,k}$ of the ordinal outcome variable in treatment group k can be reasonably estimated from $\mathcal{D}^{(r)}$ via ML estimation (i.e. by calculating the proportion of individuals within treatment group k who fall into ordinal category m). Note that even in this simple example, $\mathcal{M}$ and $\mathcal{M}^*_{\mathcal{D}^{(r)}}$ are not perfectly aligned: While $\mathcal{M}$ assumes the outcome depends solely on treatment, other (observable and latent) factors also influence it in $\mathcal{M}^*_{\mathcal{D}^{(r)}}$. Due to randomization, however, the outcome's distribution within treatment groups remains unaffected, allowing the probabilities to be estimated without bias. Although the ML estimator seems appropriate in this context, it can exhibit substantial variance, leading to discrepancies between the estimated and true probabilities. This can cause practical challenges, such as categories with low (but non-zero) true probabilities having zero counts in $\mathcal{D}^{(r)}$ due to sampling variability, resulting in estimated probabilities of zero. Consequently, simulated datasets derived from such estimates would lack certain outcome categories entirely. One potential solution is to impose a minimum sample size criterion when selecting the real datasets, though this pertains to dataset selection (see Section 3.3). Another issue for *Example-Ordinal* arises because its target is a null hypothesis. Specifically, the estimated probabilities almost always exhibit small differences across treatment groups, even when the true probabilities in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ are equal (i.e. under H_0). As a result, in the DGM constructed from the estimated probabilities, H_0 will almost always be false, whether H_0 actually holds in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$ or not. While this aligns with the model structure $\mathcal{M}$ of *Example-Ordinal*, which specifies unequal probabilities across treatment groups for at least one ordinal category, it raises concerns about how well the constructed DGM reflects reality if H_0 might plausibly hold in $\mathcal{G}^*_{\mathcal{D}^{(r)}}$. A pragmatic ad hoc approach to address this issue would be to incorporate the variance in parameter estimation through a statistical test (e.g., a Wilcoxon rank-sum test). The resulting p-values could then be used to decide whether H_0 should be assumed true or false for a given $\mathcal{D}^{(r)}$. This could, for example, be achieved by applying a threshold (e.g., $p < 0.05$) or using p-values as sampling weights for determining H_0 status (a similar idea is applied by Benidt and Nettleton, 2015, who

used p-values derived from the real dataset to sample genes for which the null hypothesis of no differential expression should hold in the resulting DGM). Only probabilities derived from datasets meeting these criteria would then be used. However, as with the sample size criterion discussed earlier, such procedures effectively act as additional exclusion criteria for dataset selection (see Section 3.3.1), which $\mathcal{D}^{(r)}$ would already need to satisfy. Although explored here with *Example-Ordinal*, similar or additional finite-sample challenges may arise in other simulation contexts.

So far, we have discussed the estimation of $\boldsymbol{\theta}_{\text{unknown}}$ in general without differentiating between $\boldsymbol{\theta}_{\text{unknown,target}}$ and $\boldsymbol{\theta}_{\text{unknown,other}}$. While the estimation tasks for both parameter types face similar challenges, the estimation of $\boldsymbol{\theta}_{\text{unknown,target}}$ is particularly critical. This is because the estimates of $\boldsymbol{\theta}_{\text{unknown,target}}$ determine the parameter values involved in the target that the methods being examined in the simulation study must recover, making the selection of an appropriate estimation method inherently more impactful. Moreover, when the statistical task of interest is estimation, the method used to estimate $\boldsymbol{\theta}_{\text{unknown,target}}$ could itself be among the competing methods evaluated in the simulation study. This introduces an element of circularity and potential bias, as the chosen method might gain an unfair advantage. To address this, all competing methods could be applied to estimate $\boldsymbol{\theta}_{\text{unknown,target}}$, with their results aggregated to provide a more balanced basis for the simulation study.

While the specific challenges associated with estimating $\boldsymbol{\theta}_{\text{unknown}}$ may differ, the provided examples illustrate the inherent complexity of the process. In principle, the challenge of inferring information from a real dataset that is not directly observable mirrors the issues encountered when analyzing data in real-world applications. For parameters related to the target, these challenges, as noted above, are closely tied to the very issues the simulation study aims to investigate or improve upon.

3.1.2 Deviating from the one-to-one inference approach: Aggregated inference and factorial designs

While the one-to-one approach, i.e. the direct inference of values from $\boldsymbol{\mathcal{D}}$ for every parameter and the subsequent mapping to Θ by combining the inferred values per dataset, is generally expected to produce DGMs that closely represent the true DGMs of the selected real datasets, there are alternative strategies worth considering for either step of the inference process from $\boldsymbol{\mathcal{D}}$ to Θ.

Aggregated inference For the first step of inferring the values for each parameter in $\boldsymbol{\theta}$, there are several reasons not to use direct inference for every parameter. As discussed in the previous section, for $\boldsymbol{\theta}_{\text{unknown}}$, the uncertainties associated with estimation may make it impractical to rely strictly on specific values inferred from the datasets. For $\boldsymbol{\theta}_{\text{known}}$, practical considerations—such as the preference for numerically tidy values—may influence the decision against direct inference. In addition to these previously discussed issues, the number of datasets, even if adjustable during their selection (see Section 3.3.1), may prove insufficient or excessive, as the researcher might prefer to consider more or fewer than R parameter values for a given parameter. In all these cases, a reasonable alternative may be to use the information from the real datasets

in aggregated form to generate the values for a parameter. We refer to this alternative as aggregated inference. A simple implementation involves identifying the minimum and maximum values for each parameter across the datasets to approximate a reasonable range and then selecting a number of values, A_q, that are systematically distributed within this range (e.g., equidistant). Alternatively, these boundaries could define a uniform distribution, from which A_q values are sampled. Moving beyond uniform distributions (and considering values beyond just the minimum and maximum), parameter values could also be used to fit and sample from other distributions (e.g., a normal distribution). Generally speaking, if A_q values are generated for a given parameter θ_q through aggregated inference instead of direct inference, the result is not a set $\hat{\boldsymbol{\theta}}_q$ containing R values (see Equation 1) but a set $\tilde{\boldsymbol{\theta}}_q$ containing A_q values:

$$\tilde{\boldsymbol{\theta}}_q = \{\tilde{\theta}_q^{(1)}, \ldots, \tilde{\theta}_q^{(A_q)}\}. \tag{3}$$

The described procedures are typically applied individually for each parameter, although in principle, one could also model a joint distribution and then draw the values for multiple parameters simultaneously. While inferring parameters via aggregated inference is unlikely to yield entirely unrealistic values, it may, depending on the chosen procedure for generating parameter values, still produce a (slightly) distorted representation of the true distribution of parameter values in the real-world DGMs targeted by the simulation study. Note that unless joint modeling is used, there is no natural way to map the sets of aggregated values for the individual parameters into realistic considered parameter vectors, as there is no inherent correspondence between the individual values. Accordingly, combining the values per dataset (as in the second step of the one-to-one approach) is not possible, and any other method of combination—including other scattershot designs and factorial designs (see below)—carries the risk of producing unrealistic parameter combinations.

Factorial designs For the second step of mapping the sets of inferred values for the individual parameters to the set of considered parameter vectors, we also want to discuss an alternative to the scattershot design that is implied in the one-to-one approach. The combination of parameter values per dataset, therefore only considering R specific (and potentially unique) combinations of values for all parameters, can complicate the analysis of the effects of individual parameters and their interactions on the performance of the methods being evaluated. An alternative is to construct a factorial design using the inferred parameter values. The implementation of factorial designs, which are employed in most simulation studies, involves combining parameter values independently—considering either all possible combinations (fully factorial) or a subset thereof (partially factorial) (Morris et al., 2019; Siepe et al., 2024). When a fully factorial design is used, the resulting set of considered parameter vectors is given by

$$\Theta = \{\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(L)}\} = \boldsymbol{\theta}'_1 \times \cdots \times \boldsymbol{\theta}'_Q =: \Theta^{\text{factorial}}, \tag{4}$$

where $\boldsymbol{\theta}'_q$ represents the set of inferred values for a given parameter θ_q and is the result of either direct inference ($\boldsymbol{\theta}'_q := \hat{\boldsymbol{\theta}}_q$, see Equation 1) or aggregated inference ($\boldsymbol{\theta}'_q := \tilde{\boldsymbol{\theta}}_q$, see Equation 3). If

every parameter is inferred directly from the R datasets (i.e. $\boldsymbol{\theta}'_q := \hat{\boldsymbol{\theta}}_q$ for all $q \in \{1, \ldots, Q\}$), the set $\Theta^{\text{factorial}}$ (see Equation 4) includes the set of considered parameter vectors that would result from the second step of the one-to-one approach (see Equation 2; i.e. $\Theta^{\text{factorial}} \supsetneq \Theta^{\text{one-to-one}}$). If, additionally, each dataset provides unique values for every parameter, the fully factorial design results in a total of $L = |\Theta^{\text{factorial}}| = R^Q$ considered parameter vectors in $\Theta^{\text{factorial}}$. Importantly, when using direct inference, employing a factorial design instead of a scattershot design introduces the risk of producing unrealistic considered parameter vectors: while the individual parameter values inferred from real datasets may each be plausible, their combinations might not be, potentially resulting in DGMs that fail to represent any real-world DGM relevant to the simulation study. If aggregated inference is used instead, the same concern applies in principle—except that, as noted above, mapping sets of aggregated parameter values carries this risk regardless of the way in which the values are combined.

3.2 Inferring parts of the model structure from a set of real datasets

In the previous section, we considered the scenario where only parameters are set to be real-data-based. However, researchers may also wish to infer specific parts of the model structure from the selected real datasets. For parts of the model structure whose true form in the underlying DGM of a real dataset can only be inferred with uncertainty (e.g., a distribution or functional relationship), similar issues as those discussed in Section 3.1.1 for $\boldsymbol{\theta}_{\text{unknown}}$ arise and must be taken into account—such as the impact of sampling variability—when choosing an inference method. As with parameter inference, the choice of method can substantially influence the results of the simulation study, particularly for parts of the model structure that explicitly appear in the formulation of the simulation target. In our four examples, the only parts of the model structure falling into this category are whether the null hypothesis (or hypotheses) holds in *Example-Ordinal*, *Example-Survival*, and *Example-DE-Analysis*.

Unlike parameter inference, which involves estimating numerical values, inferring parts of the model structure typically requires categorical decisions, such as determining whether a null hypothesis holds or selecting an appropriate distribution for a variable. Often, this can be done using hypothesis tests, as already outlined for *Example-Ordinal* in Section 3.1.1, where we considered how the status of the null hypothesis of no treatment effect could be assessed via a test (albeit as a criterion for selecting the real datasets rather than for inference from them). Other examples include testing for the presence of an interaction effect or correlation. In some cases—such as selecting the distribution of a variable—inferring a specific part of the model structure via hypothesis testing additionally requires defining a set of plausible options, from which the best-fitting choice is then selected using an appropriate test. Importantly, this approach does not guarantee that any of the considered options closely approximate the true model structure of any of the selected real datasets. To illustrate the data-driven selection of distributions, consider *Example-Survival*, for which we discussed in Section 3.1.1 that the true distribution of survival times may deviate from the exponential distribution assumed by $\mathcal{M}$. This can be addressed by considering more flexible distribution options, such as Weibull, gamma, Gompertz, or mixture distributions, with the best-fitting distribution identified through

goodness-of-fit tests like the Cramér–von Mises test. This procedure is implemented by Thurow et al. (2024) in a meta-scientific study on simulating realistic survival data.
If parts of the model structure are real-data-based and thus may vary across datasets, multiple model structures can emerge. Accordingly, when inferring parts of the model structure from the set of real datasets, this procedure must be applied before inferring the parameters. Specifically, the procedure described in Section 3.1 must be applied separately for each subset of datasets corresponding to each resulting model structure derived from the inference of real-data-based parts. For example, in an extended version of *Example-Survival*, suppose half of the real datasets suggest an exponential distribution, while the other half align more closely with a Weibull distribution. In this case, parameter inference must be conducted separately for the datasets associated with $\mathcal{M}_{\text{Exp}}$ and $\mathcal{M}_{\text{Weibull}}$.

3.3 Selecting real datasets as a basis for the DGMs

We propose three general requirements for real datasets used to construct real-data-based DGMs: (D1) they must be accessible to others, with a transparent and reproducible selection process; (D2) their true DGMs must correspond to a representative subset of the simulation study's domain of interest; and (D3) they must provide the necessary information to both meaningfully infer the DGM components intended to be real-data-based and assess their eligibility. In the following, we examine these requirements in more detail, including strategies to fulfill them and the challenges that may arise. Specifically, we discuss the identification of a database likely to contain eligible datasets as well as the specification of additional eligibility criteria to ensure that the final selection meets all requirements.

3.3.1 Database

Regarding the choice of database, requirement (D1) excludes collections of datasets that are only accessible to the researchers conducting the simulation study. Public data repositories, therefore, represent a natural solution to fulfill (D1). For example, in the context of *Example-DE-Analysis*, the open data repository of The Cancer Genome Atlas (TCGA) program (https://www.cancer.gov/tcga) could be utilized, offering genomic, epigenomic, transcriptomic, and proteomic data for 33 cancer types. Similarly, publicly accessible platforms like OpenML (Vanschoren et al., 2014) or the UCI Machine Learning Repository (Kelly et al., n.d.), which offer datasets across various research domains and data types, can serve as valuable resources. Although primarily used for studies focused on prediction tasks, where methods are evaluated directly on the real datasets, these repositories can also be used for real-data-based simulation studies with other aims (e.g., Stolte et al., 2024, albeit in a meta-scientific context). In addition to data repositories that are fully open to the public, we also consider data repositories that are broadly accessible to the research community under controlled conditions to fulfill (D1). Examples include clinical research data-sharing platforms such as Vivli (https://vivli.org; Bierer et al., 2016), the Yale Open Data Access (YODA) Project (https://yoda.yale.edu/; Ross et al., 2018), and the Virtual International Stroke Trials Archive (VISTA; https://www.virtualtrialsarchives.org/vista/; Ali et al., 2007).

Public data repositories may offer many datasets eligible under requirement (D3) (i.e. providing sufficient information for inference and eligibility assessment), as researchers have direct access to the dataset. However, finding a repository with enough datasets to fulfill requirement (D2) (i.e. adequately representing the domain of interest) can be more challenging. An alternative approach is to reconstruct datasets from tables and figures in research publications—such as journal articles and reports—that present aggregated or visualized data. For example, survival data can be reconstructed from digitized survival curves (e.g., Guyot et al., 2012), as employed by Trinquart et al. (2016), Royston et al. (2019), Dormuth et al. (2022), and, in the simulation context, Thurow et al. (2024). In these cases, possible databases may be collections of publications, such as PubMed (https://pubmed.ncbi.nlm.nih.gov/), as referenced by Dormuth et al. (2022) and Thurow et al. (2024), or specific journals (Royston et al., 2019; Trinquart et al., 2016).

Another alternative specifically for parametric DGMs is to rely on aggregated data instead of raw datasets. This approach can still satisfy requirement (D3), provided that all necessary information can be extracted without direct access to the full datasets. Although the underlying datasets are not directly accessible (and also not reconstructed), we continue to use the notation $\mathcal{D}$ to refer to the respective dataset used in the publication, which is implicitly represented through the extracted information. Relying on aggregated data can, for example, be feasible in *Example-Ordinal*, where the information needed to estimate outcome probabilities—the number of individuals in each ordinal category per group—is often available in tables or figures included in publications. In other cases, parameter inference may have already been conducted within the publication itself, meaning that the reported values can be directly adopted. For instance, in the context of *Example-Meta-Analysis*, both meta-analysis-level parameters (e.g., the [estimated] overall effect) and study-level parameters (e.g., group means) are commonly reported in published meta-analyses, thus providing the necessary information for both the meta-analysis dataset and the individual study datasets without requiring access to raw data. The approach of using aggregated information from published meta-analyses was implemented in the simulation study by Langan et al. (2019). When using aggregated data, suitable databases again include PubMed and specific journals but also systematic reviews summarizing multiple publications (the latter serving as a database in contexts other than *Example-Meta-Analysis*). Given the vast number of publications available in collections like PubMed or individual journals, it is often reasonable to define the database as all publications within the collection, restricted to a specific time frame.

Depending on the chosen database, several potential limitations may arise that researchers should be aware of, address transparently, and, if possible, mitigate through additional eligibility criteria (see Section 3.3.2). For public repositories containing existing datasets, similar concerns to those typically discussed in the context of real-data studies apply (see, for example, the discussions by Strobl and Leisch, 2024, Boulesteix et al., 2015, and Friedrich and Friede, 2024). A key concern is that the true DGMs of real datasets "donated" to repositories may fail to adequately represent the domain of interest to which the simulation study's results and recommendations are intended to apply. For instance, the real datasets in public repositories might

focus heavily on specific subpopulations of DGMs (e.g., a particular cancer type). Furthermore, the quality of the data may be low depending on how it was collected and curated, with issues such as poor documentation or missing values.

For databases consisting of collections of publications, similar issues may arise. Specifically, publication bias can result in datasets with true DGMs that are not representative of the domain of interest (e.g., if only results based on datasets with large true effect sizes are published), and the quality and methodological rigor of how the underlying datasets were collected and reported may also vary widely. However, when using publication collections as databases, additional challenges arise, because the original datasets are rarely directly accessible. For reconstruction procedures, the quality and accuracy of reconstructed datasets depend on several factors, including the validity of assumptions underlying the reconstruction algorithm. For instance, the previously mentioned algorithm by Guyot et al. (2012) for reconstructing survival data relies on the assumption that censoring occurs at a constant rate within each time interval, which may not hold in all cases. When aggregated dataset information is used, inconsistencies may emerge if different publications apply varying aggregation methods. Moreover, publication bias may not only introduce issues with representativeness but also lead to optimistically biased parameter estimates by incentivizing practices like p-hacking or selective reporting. In addition, relying on publication collections as databases may demand more time and effort than using open repositories with readily available datasets. This is because reconstructing or extracting relevant information from publications involves additional steps, and assessing publications for eligibility criteria often requires considerably more time.

Finally, a practical issue that arises independently of the chosen database is that the database itself only determines the maximum possible number of datasets that could be selected but not how many will actually meet all eligibility criteria. Accordingly, after applying these criteria, researchers may end up with too few datasets, which would risk inadequate representation of the domain of interest, or too many datasets, which would make it impractical to reasonably process them in subsequent simulation steps. As a pragmatic approach, we suggest that researchers specify a minimum and maximum number of datasets to be selected and, after applying the eligibility criteria, check whether the number of selected datasets falls within this range. If the number is too low, the database should be expanded (e.g., by including additional publication years or related repositories). If the number is too high, we recommend using random selection to reduce the number of datasets to the specified maximum. While we cannot provide a general recommendation for an appropriate minimum and maximum, as these values depend on the specific simulation study, we argue that a minimum of only one or two datasets—which is common in practice—is typically insufficient.

3.3.2 Eligibility criteria

Once the database has been specified, datasets can be selected based on eligibility criteria, which may be formulated as inclusion or exclusion criteria. In general, assessing their fulfillment may involve a combination of automated methods (e.g., filtering datasets in a repository or applying search strings for journal databases) and manual review (e.g., screening data documentation or publications).

Ideally, all eligibility criteria would be specified before dataset selection to prevent bias from post hoc modifications. However, this is often not feasible, as criteria may need to be refined or extended during the assessment process to account for unforeseen challenges or inconsistencies. In any case, to meet requirement (D1) (i.e. ensuring the selection process is transparent and reproducible), it is essential to clearly and comprehensively report the final criteria applied during the selection process.

The criteria themselves can be categorized into those addressing (D2) (i.e. ensuring adequate representation of the domain of interest) and those related to (D3) (i.e. providing sufficient information for inference and eligibility assessment), and researchers should explicitly indicate which requirement each criterion is intended to fulfill.

Defining eligibility criteria to fulfill (D2) involves translating the domain of interest—i.e. the population of true DGMs to which the simulation study's results and recommendations are intended to apply—into concrete criteria. As stated in Section 2.2.1, the specifications that define the domain of interest consist of researcher-specified components of interest (e.g., the number of treatment groups, the violation of a specific assumption) and constraints imposed on real-data-based components (e.g., restricting parameter values to those observed in a specific disease type). In contrast, researcher-specified components of convenience are not incorporated into the eligibility criteria, meaning that datasets differing in these aspects may still be eligible. In general, defining (D2)-related criteria can be a challenging task, in part because researchers typically do not distinguish explicitly between components of interest and components of convenience when planning simulations. Note that for databases consisting of collections of publications, an effective strategy for defining a search string to identify an initial set of relevant publications is to include the names of methods commonly used to analyze datasets from this domain as keywords (e.g., the methods compared in the simulation study).

Beyond the challenge of specifying criteria related to (D2), assessing whether these criteria are met presents additional difficulties. The domain of interest is defined by the *true* DGMs, meaning that, in an ideal scenario, datasets would be selected based directly on these underlying true DGMs. However, in practice, only the observed datasets (or their aggregated versions) are available for assessment. Accordingly, if a criterion refers to parts of the model structure or parameters whose true form or value in the true DGM underlying a real dataset cannot be known with certainty (i.e. unknown components, as defined in Section 2.2.2), its fulfillment cannot be determined with certainty either. This applies, for instance, when only datasets in which a specific assumption does not hold or those with a large true effect size should be selected. In principle, this introduces the same difficulties discussed in Sections 3.1 and 3.2, with the key difference that, in those cases, inference was performed on an already selected set of datasets, $\boldsymbol{\mathcal{D}}$, whereas here, inference is required as part of the process of identifying $\boldsymbol{\mathcal{D}}$ in the first place. As a consequence, when a criterion involves a component whose true form or value in the true DGM underlying a dataset is unknown, the criterion should also specify how fulfillment should be assessed (e.g., by defining a statistical test or other procedure).

As stated above, in contrast to (D2), the criteria addressing (D3) do not relate to the study's domain of interest but rather ensure that the selected datasets contain the necessary informa-

tion for meaningful parameter inference and the assessment of eligibility. Such criteria may, for instance, address the sample size of the dataset. As already mentioned in Section 3.1.1, this could include requiring a minimum total sample size n to ensure stable parameter estimation or, in the case of *Example-Ordinal*, ensuring that there are more than zero observations per group and ordinal category. Note that, in this case, n and other parameters related to sample size cannot be real-data-based, and since their specific values are typically not of direct interest, they would be considered researcher-specified components of convenience (and represent examples of this component category that neither have negligible impact nor are expected to hold by default but also cannot be reasonably inferred; see Section 2.2.1). Other criteria may address the quality of the selected datasets, which, as noted in the previous section, can be a concern. Here, criteria could specify that certain metadata must be available, or they could, for example, restrict inclusion to publications from reputable journals.

While the criteria mentioned above primarily relate to the "meaningfully" aspect of (D3), other criteria—particularly when databases consist of collections of publications—ensure that parameter inference and the assessment of eligibility are possible in the first place. For example, when reconstructing datasets for *Example-Survival* with the algorithm by Guyot et al. (2012), relevant criteria may require that the number at risk is reported and that survival curves are presented in high resolution (Dormuth et al., 2022; Thurow et al., 2024). Similarly, when using aggregated information for *Example-Ordinal*, a relevant criterion is that the number of patients in both treatment groups and ordinal categories is clearly reported in tables or figures.

Importantly, the criteria specified so far determine whether a dataset as a whole should be selected or not. We refer to these as dataset-level criteria. However, some of these criteria also inherently specify which subsets of the data within an included dataset are used. For example, requiring datasets to include specific types of outcome variables already restricts the usable subset of each dataset. Beyond such implicit subset specification, additional explicit subset-level criteria may be applied after dataset selection is complete. These criteria refine the selection of specific elements (e.g., outcome variables, treatment groups, or covariates) within an included dataset to ensure consistency across the selected datasets. Additionally, considering subset-level criteria allows for broader dataset inclusion by enabling the use of relevant subsets within datasets that would otherwise be excluded. For instance, if a simulation study focuses on two treatment groups, rather than excluding all datasets that do not match this criterion exactly, one could exclude only those with a single treatment group while keeping those with more than two groups, subsequently selecting the two treatment groups with the largest sample sizes. This subset-level criterion would relate both to (D2) (ensuring the selection of the two treatment groups) and to (D3) (ensuring an adequate sample size).

4 Example illustrations

To empirically illustrate the implementation of parametric DGMs based on a systematically selected set of real datasets, we conduct two simulation studies that build on two of the examples discussed in the previous sections (*Example-Ordinal* and *Example-DE-Analysis*). We also compare the systematic parameter inference from multiple datasets with other approaches

by additionally considering purely researcher-specified parameters in the first illustration (Section 4.1) and parameters inferred from a single real dataset in the second (Section 4.2). Importantly, the simulations are not intended as comprehensive simulation studies with in-depth analyses—each of which could warrant a dedicated paper—but rather as illustrative examples demonstrating the implementation of parametric DGMs based on a systematically selected set of datasets and their impact on results. The simulations and analyses are conducted in the software environment `R` (R Core Team, 2023), and the code to reproduce all results is available at https://github.com/NiesslC/realdata_simulations.

4.1 Two-arm randomized controlled trial with an ordinal outcome

4.1.1 Design

ADEMP structure An overview of the ADEMP structure for this simulation can be found in Table 3. The "A", "D", and "E" aspects correspond to those listed for *Example-Ordinal* in Section 2.3, at least for the aspects already specified there. Accordingly, we consider a two-arm (i.e. $K = 2$) randomized controlled trial, and we aim to evaluate the ability of methods to detect a true treatment effect between groups.

Table 3: ADEMP structure for the example illustration on hypothesis testing in the context of a two-arm randomized controlled trial with an ordinal outcome. Either all parameters are researcher-specified, or all parameters except the outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ are researcher-specified, with the latter being real-data-based. Accordingly, in the first case, $\boldsymbol{\lambda} = \{K, M, n, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2\}$, while in the second case, $\boldsymbol{\theta} = \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2\}$ and $\boldsymbol{\lambda} = \{K, M, n\}$.

Aim	Evaluation of methods testing the null hypothesis H_0 of no treatment differences in two-arm (i.e. $K = 2$) randomized controlled trials with ordinal outcomes having M categories, in settings where H_0 is false
Data-generating mechanisms (DGMs)	**Model structure $\mathcal{M}$** · Ordinal outcome of individual i, $i = 1, \ldots, n$, in group $k \in \{1, 2\}$ (equal group sizes) is drawn from Multinomial$(1, \boldsymbol{\pi}_k)$, where $\boldsymbol{\pi}_k = (\pi_{1,k}, \ldots, \pi_{M,k})$, $\pi_{m,k} = P(Y = m \mid X = k)$, $\sum_{m=1}^{M} \pi_{m,k} = 1$, and $\pi_{m,1} \neq \pi_{m,2}$ for at least one $m \in \{1, \ldots, M\}$. **Parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ (varied fully factorially)** · $\boldsymbol{\lambda}$: $K = 2$; $M = 7$; $n \in \{60, 120, 200, 300, 600\}$ · $\boldsymbol{\lambda}/\boldsymbol{\theta}$: $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ (4 researcher-specified and 15 real-data-based pairs of outcome probabilities, see Table S1 and Table S2) **Number of repetitions per DGM**: $n_{\text{sim}} = 10{,}000$
Estimand / Target	The null hypothesis $H_0 : \pi_{m,1} = \pi_{m,2}$ for all $m \in \{1, \ldots, M\}$
Methods	4 methods: Chi-square test, Fisher's exact test, Wilcoxon rank-sum test, proportional odds ordinal logistic regression
Performance measure	Power, estimated as $\frac{1}{n_{\text{sim}}} \sum_{s=1}^{n_{\text{sim}}} \mathbb{1}(p_s \leq 0.05)$, with p_s being the p-value from repetition s, $s = 1, \ldots, n_{\text{sim}}$

In *Example-Ordinal*, the remaining parameters to specify are the number of ordinal categories, M, the sample size n, and the outcome probabilities $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$. To simplify the illustration, we focus on real-data-based inference for the outcome probabilities while fixing the number of ordinal categories to $M = 7$ and setting the sample size manually to five predefined values ($n \in \{60, 120, 200, 300, 600\}$). For the outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$, which are specified

jointly for both groups and treated as a single factor, we consider 4 researcher-specified outcome probabilities (Table S1) and 15 real-data-based outcome probabilities (Table S2). The process of generating the real-data-based outcome probabilities, which were inferred after defining the researcher-specified outcome probabilities, will be detailed below. The different sample sizes and outcome probabilities are combined using a fully factorial design, resulting in 20 DGMs when $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ is researcher-specified ($5 \times 4 = 20$) and 75 DGMs when it is real-data-based ($5 \times 15 = 75$). Regarding the remaining ADEMP aspects—methods and performance measures—we evaluate four statistical tests: Chi-square test, Fisher's exact test, Wilcoxon rank-sum test, and proportional odds (PO) ordinal logistic regression, which are all tests that may be used in this context (Selman et al., 2024). For a given DGM, the performance of the methods is assessed by their power to reject the null hypothesis of no treatment difference, which is estimated using the proportion of rejected null hypotheses at a nominal significance level of $\alpha = 0.05$.
The number of repetitions (i.e. simulated datasets) per DGM, denoted as n_{sim}, is set to 10,000, ensuring that the Monte Carlo standard error (MCSE) remains below 0.5% for a worst-case rejection proportion of 0.5 (the probability at which MCSE is maximized; Morris et al., 2019). However, we only run the methods on simulated datasets where all seven ordinal categories are observed, which reduces the number of analyzed datasets for some DGMs. To still ensure stable power estimates, we exclude all DGMs for which the number of repetitions where all seven ordinal categories are observed is lower than 8,000.

Dataset selection and parameter inference As stated in Section 3.3.1, the information needed to estimate $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ is often reported in publications on corresponding trials, meaning access to the raw datasets is not necessarily required. Accordingly, one may use a collection of publications as a database, which in this illustration is specified as all research publications in *The New England Journal of Medicine* (*NEJM*), with publication years restricted to 2017–2022. The dataset selection process from this database is summarized below, with full details provided in Supplementary Section A.2.1.
To construct the search string for identifying relevant publications, we include the terms "randomized" and "ordinal", along with variations of the names of the considered methods. This search yields 270 publications, which are manually screened according to 11 dataset-level eligibility criteria. Of these criteria, eight relate to (D2), including conditions such as requiring at least one ordinal outcome, excluding studies where the ordinal outcome has fewer or more than $M = 7$ categories, and excluding those in which participants were randomized in groups or clusters rather than individually. The remaining three criteria are associated with (D3), two of which ensure that the relevant information is clearly reported and that all ordinal categories contain at least one observation, while the third addresses cases where two publications use the same dataset. Applying these criteria results in $R = 15$ eligible publications and corresponding datasets (see Table S2). On the subset level, we specify the following criteria for the 15 datasets: When a publication includes more than two treatment groups, we select the two with the largest sample sizes. Similarly, if multiple ordinal outcomes are available, we prioritize the outcome considered most important; if no clear priority is established, we select the outcome with the highest sample size.

Notably, we do not impose any criteria on minimum sample size or whether the null hypothesis of no treatment effect, H_0, is false (both of which were discussed as potential criteria throughout Section 3). The latter implies that H_0 may, in fact, be true in some selected datasets. This decision is intentional to illustrate the impact of not applying such a criterion. However, in a formal simulation study, it may be reasonable to include such restrictions.

The outcome probabilities are estimated using simple maximum likelihood estimation, where $\hat{\pi}_{m,k}$ represents the proportion of individuals in treatment group k who fall into ordinal category m.

4.1.2 Results

Parameter characteristics To systematically compare the researcher-specified and real-data-based outcome probabilities, directly examining their individual values is impractical due to their multi-dimensional nature (but see Figure S2 for an example of a researcher-specified and a real-data-based set of outcome probabilities). Instead, one or several summary measures are needed to characterize them. For simplicity, we focus here on the relative effect, which has also been considered by Funatogawa and Funatogawa (2023) in their similar simulation study. The relative effect, denoted as RE, is defined as $P(Y_1 > Y_2) + \frac{1}{2}P(Y_1 = Y_2)$, where Y_1 and Y_2 denote the ordinal outcome variables in the two treatment groups. A relative effect of 0.5 indicates no systematic difference between the groups, whereas values greater or smaller than 0.5 suggest that observations in group 1 tend to be larger or smaller, respectively, relative to those in group 2 (Agresti, 2010; Brunner et al., 2021). Figure 1 presents the relative effects for the researcher-specified and real-data-based outcome probabilities. Since both values below and above 0.5 indicate differences between groups, we consider the absolute deviation from 0.5, $|RE - 0.5|$, where 0 indicates no difference, and larger values correspond to greater differences between the two groups.

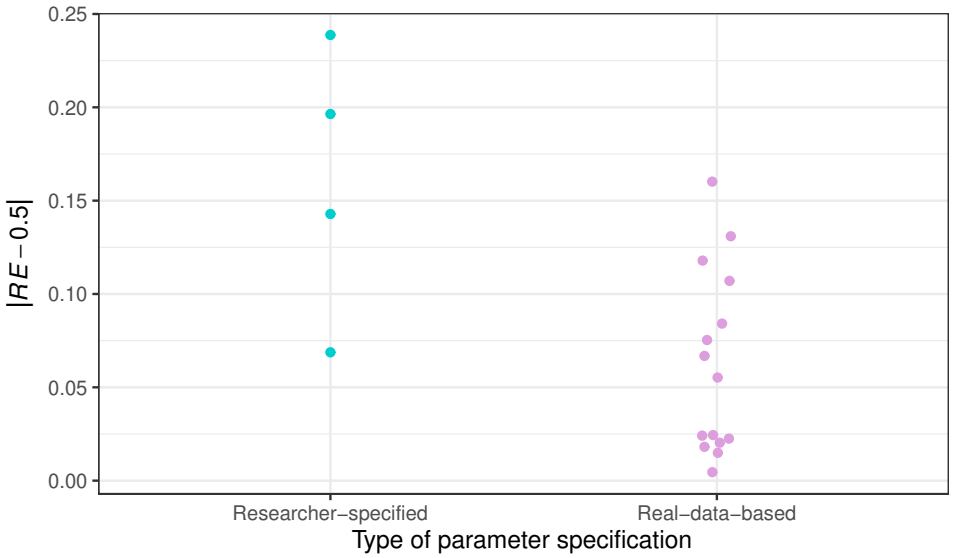

Figure 1: Absolute deviation from 0.5 in the relative effect ($|RE - 0.5|$) for 4 researcher-specified and 15 real-data-based outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$. The more the relative effect deviates from 0.5, the greater the difference between the two treatment groups.

As shown in Figure 1, the real-data-based outcome probabilities generally imply smaller group differences than the researcher-specified probabilities, illustrating that the two different approaches to parameter specification can lead to systematic differences in effect sizes. In the specific case considered here, as discussed in Section 3.1.1, this may be due to the fact that, for some of the true DGMs underlying the 15 real datasets, H_0 of no treatment effect holds. However, even if this applies to some datasets, a systematic difference remains between the two types of parameter specification.

Method performance Figure 2 presents both the absolute performance (panel a: estimated power) and the relative performance (panel b: difference to the best, i.e. highest power) in relation to the relative effect deviation from 0.5 ($|RE - 0.5|$) for different sample sizes n. Note that in both panels, not all 19 ($4 + 15$) outcome probabilities are included. As stated before, we exclude DGMs for which the number of repetitions where all seven ordinal categories are observed is lower than 8,000. For one of the 15 real-data-based outcome probabilities, namely the probabilities extracted from Perkins et al. (2018), this was the case for all five DGMs, i.e. for all five possible values for n. For the other 14 real-data-based outcome probabilities, only DGMs with smaller sample size values ($n \in \{60, 120\}$) fell short of the 8,000 repetitions. Specifically, fewer than 8,000 datasets with observations in all seven ordinal categories were simulated for six real-data-based DGMs with $n = 60$ and one real-data-based DGM with $n = 120$, leading to their exclusion from the respective rows of the panels.

As seen in Figure 2a, the estimated power generally increases with the relative effect deviation from 0.5. For the Wilcoxon rank-sum test and PO ordinal logistic regression, the estimated power follows a nearly deterministic monotonic increase with the relative effect deviation, and both tests yield highly similar results (see Supplementary Section A.3 for a brief explanation of the theoretical basis for this alignment). For the Chi-square test and Fisher's exact test, the relationship between estimated power and the relative effect deviation is less clear. A more detailed investigation into additional characteristics (e.g., asymmetry in outcome probabilities or the expected number of observations per category) would be necessary in a more comprehensive simulation study but is beyond the scope of this illustration.

With regard to the comparison between researcher-specified and real-data-based probabilities, the systematic differences in relative effect deviation (Figure 1) are reflected in corresponding differences in estimated power (Figure 2a). Specifically, all tests tend to yield higher estimated power for the researcher-specified probabilities. Additionally, Figure 2b highlights that the relative performance of the four tests varies across the two types of parameter specification. For Chi-square and Fisher's exact test, the difference in power between the best-performing test and these tests is larger for researcher-specified parameters at smaller sample sizes. Conversely, for the Wilcoxon rank-sum test and PO ordinal logistic regression, the difference in power is more pronounced for real-data-based probabilities at larger sample sizes. Accordingly, if only one type of parameter specification had been considered, the conclusions regarding the relative performance of the tests would have differed, depending on whether the parameters were specified by the researcher or based on real data.

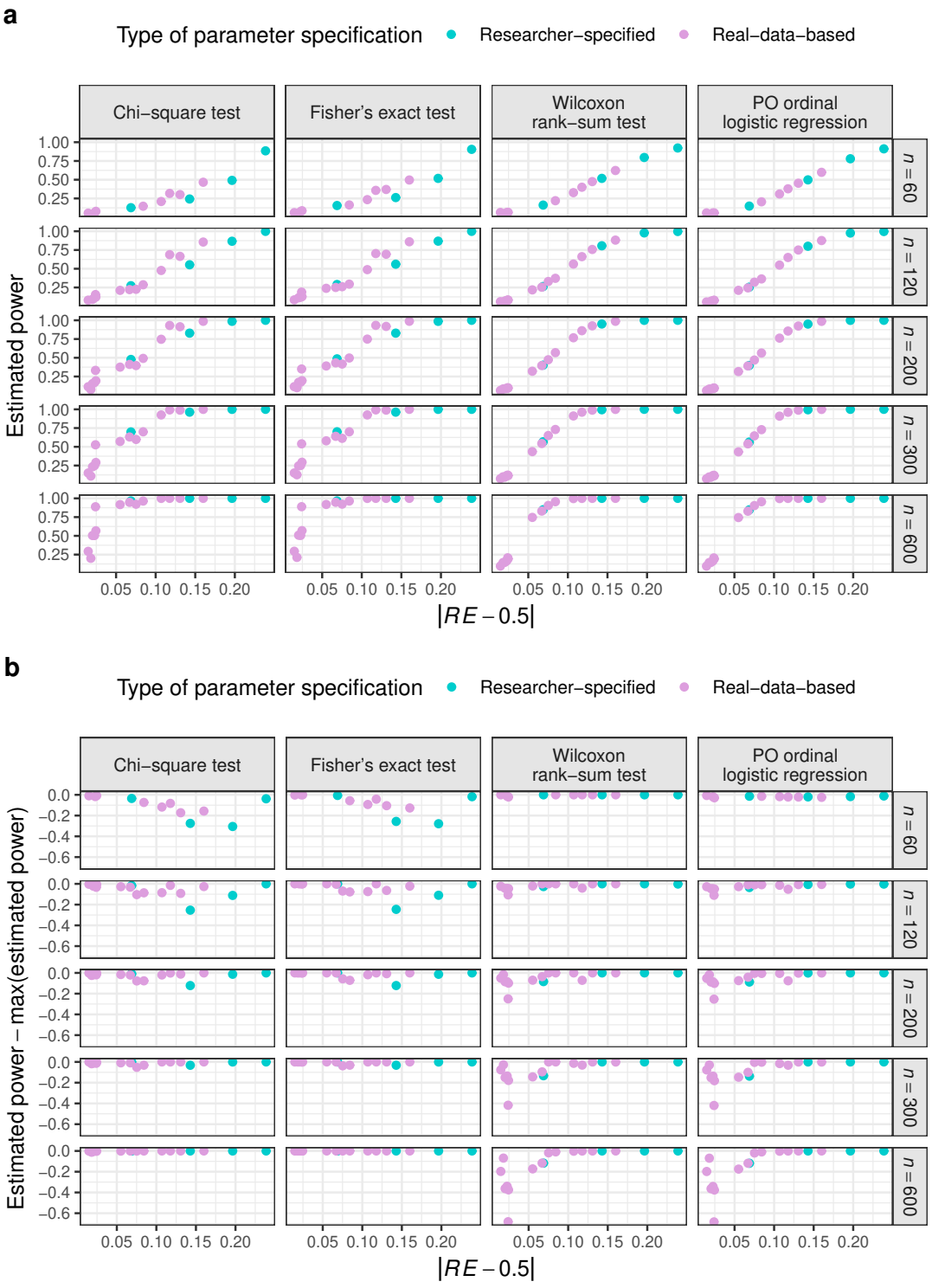

Figure 2: Absolute (panel a) and relative performance (panel b) of four statistical tests in relation to the absolute deviation of the relative effect from 0.5 ($|RE - 0.5|$), calculated from the outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$, across different sample sizes n. Panel a shows the absolute estimated power. Panel b shows the difference between the estimated power and the highest estimated power within each DGM. In both panels, the DGMs for all 4 researcher-specified and 14 of the 15 real-data-based outcome probabilities are included for $n \in \{200, 300, 600\}$, while for $n \in \{60, 120\}$, some of these 14 real-data-based outcome probabilities are not included due to an insufficient number of repetitions where all seven ordinal categories are observed. The remaining one of the 15 real-data-based probabilities is not included at all for the same reason.

4.2 Differential gene expression analysis

4.2.1 Design

ADEMP structure For the second illustrative simulation study, we adopt the simulation design from Baik et al. (2020) (hereafter referred to as B20) on differential gene expression analysis, extending it only by the number of real datasets used for parameter inference, as detailed below. Since B20's simulation is highly comprehensive, we include only a subset of the original DGMs (see Supplementary Section B.1 for details on excluded DGMs) and focus on a single performance measure. An overview of the resulting ADEMP structure is provided in Table 4.

Table 4: ADEMP structure for the example illustration on methods for differential gene expression analysis. Parameters $\boldsymbol{\lambda} = \{K, p, \lambda_{\text{FC}}, minFC, p_{\text{up}}, p_{\text{DE}}, n\}$ are researcher-specified, and parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\phi}\}$ are real-data-based.

Aim	Evaluation of methods for differential gene expression analysis, i.e. methods that identify genes with differences in their RNA-Seq expression levels, in a two-group (i.e. $K = 2$) setting (e.g., cancer vs. normal)
Data-generating mechanisms (DGMs)	**Model structure $\mathcal{M}$** • Read count $r_{i,j}$ of gene j, $j = 1, \ldots, p$, and sample i, $i = 1, \ldots, n$, in group $k \in \{1, 2\}$ (equal group sizes) is simulated as $R_{i,j} \sim \text{NB}(\mu_j \cdot \text{FC}_j, \phi_j)$, $\mu_j, \phi_j \geq 0$, if $x_i = 1$, and $R_{i,j} \sim \text{NB}(\mu_j, \phi_j)$, $\mu_j, \phi_j \geq 0$, if $x_i = 2$. Among p genes, proportion p_{DE} are differentially expressed (DE) with $\text{FC}_j \neq 1$. Among DE genes, proportion p_{up} are upregulated ($\text{FC}_j > 1$), rest are downregulated ($\text{FC}_j < 1$). FC_j is defined as $\text{FC}_j = \begin{cases} (minFC + randFC_j) & \text{if gene } j \text{ is DE and upregulated,} \\ (minFC + randFC_j)^{-1} & \text{if gene } j \text{ is DE and downregulated,} \end{cases}$ where $randFC_j$ is drawn from $\text{Exp}(\lambda_{\text{FC}})$. **Parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}$ (varied fully factorially)** • $\boldsymbol{\lambda}$: $K = 2$; $p = 10{,}000$; $p_{\text{up}} = 0.5$; $p_{\text{DE}} \in \{0.05, 0.10, 0.30, 0.60\}$; $\lambda_{\text{FC}} = 1$; $minFC = 1.5$ for $n = 6$ and $minFC = 1.2$ for $n = 20$; $n \in \{6, 20\}$ • $\boldsymbol{\theta}$: $(\boldsymbol{\mu}, \boldsymbol{\phi}) = ((\mu_1, \ldots, \mu_p), (\phi_1, \ldots, \phi_p))$ (14 pairs of values, see Table S3 for datasets) **Number of repetitions per DGM**: $n_{\text{sim}} = 50$
Estimand / Target	The null hypothesis $H_0 : \text{FC}_j = 1$ for all $j \in \{1, \ldots, p\}$
Methods	11 methods: edgeR, edgeR.ql, edgeR.rb, DESeq.pc, DESeq2, voom.tmm, voom.qn, voom.sw, ROTS, baySeq, PoissonSeq
Performance measure	Area under the receiver operating characteristic curve (AUC)

Similar to the first illustration, the "A", "D", and "E" aspects largely align with those specified for *Example-DE-Analysis* (see Section 2.3), which was itself based on B20's study. That is, we aim to evaluate methods for identifying DE genes from RNA-Seq data in a two-group (i.e. $K = 2$) setting with n samples and p genes. The only difference is that in *Example-DE-Analysis*, the fold changes of DE genes were directly assigned, whereas in the actual study by B20, they are specified in a more refined manner, incorporating additional parameters and a stochastic component. Our simulation study follows this more detailed formulation by B20 (see Table 4). In the subset of DGMs considered in our simulation, all parameters except for the gene-wise mean expression $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)$ and dispersion $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)$ are researcher-specified (by

B20), and only the number of samples and the proportion of DE genes are varied ($n \in \{6, 20\}$ and $p_{\text{DE}} \in \{0.05, 0.10, 0.30, 0.60\}$). B20 estimate the values of $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ from the Kidney Renal Clear Cell Carcinoma (KIRC) RNA-Seq dataset from the dataset collection of TCGA (mentioned earlier in Section 3.3.1), resulting in a single vector for both $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$. As an extension, we consider 13 additional TCGA datasets, leading to 14 $(\boldsymbol{\mu}, \boldsymbol{\phi})$ pairs in total, which we consider jointly. Details on dataset selection and parameter inference are provided below. B20 employ a fully factorial design, which we also adopt. When using only one TCGA dataset to infer $(\boldsymbol{\mu}, \boldsymbol{\phi})$, this results in $2 \times 4 = 8$ DGMs. When all 14 (eligible) TCGA datasets are considered, this results in $8 \times 14 = 112$ DGMs. Following B20, the number of simulated datasets per DGM is set to $n_{\text{sim}} = 50$.

B20 evaluate 12 methods, including, e.g., edgeR (Robinson et al., 2010), DESeq2 (Love et al., 2014), and their variants (see B20 for details). However, in our simulation, we are only able to consider 11 methods, as SAMseq (Li & Tibshirani, 2013) is excluded due to persistent execution errors that prevent it from running. For performance evaluation, we consider only the area under the receiver operating characteristic curve (AUC), which is the primary measure used by B20, although B20 also consider the true positive rate and the false discovery rate.

We conduct the simulation using the `compareDEtools R` package, which accompanies B20's study.

Dataset selection and parameter inference Since the real dataset used by B20 (KIRC) is from the dataset collection of TCGA, we consider TCGA as the database, which contains RNA-Seq datasets for 33 cancer types. To determine dataset eligibility, we follow the approach that B20 used for the KIRC dataset, whereby only tumor and normal tissues are considered and only paired samples (i.e. those in which tumor and normal tissue originate from the same patient) are included. Specifically, we define one (D2) criterion, which excludes datasets that do not contain both tumor and normal samples, and one (D3) criterion, which excludes unmatched samples and, as an additional requirement for our study, datasets with fewer than 10 matched sample pairs. After applying both criteria, $R = 14$ datasets remain. Details on the datasets and the selection process are provided in Supplementary Section B.2.

For each TCGA dataset, the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$ are inferred following the same approach used by B20 for the KIRC dataset. Specifically, for each gene in a given TCGA dataset, the mean expression is computed as the average across all samples, while dispersion is estimated in a more complex manner: first, RNA-Seq counts are normalized to account for differences in sequencing depth (i.e. variations in the total number of reads per sample). Then, an empirical Bayes approach is applied to improve the stability of gene-wise dispersion estimates (see the `compareDEtools` package for more details). Since each TCGA dataset contains 20,501 genes, this process results in 20,501 mean and dispersion values per dataset. From these values, those corresponding to genes with a mean expression of less than 10 are excluded. Finally, $p = 10{,}000$ genes (the number used in the simulation) are randomly selected, and their corresponding mean and dispersion estimates constitute the resulting vectors $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$. Although these exclusion steps may appear to be subset-level criteria (see Section 3.3.2), the excluded genes still contribute to

parameter inference as they are used for dispersion estimation, making this better viewed as a filtering procedure. Moreover, note that we follow the implementation by B20, who draw new genes for each repetition.

4.2.2 Results

Parameter characteristics Similar to the outcome probabilities in the first illustration, the multi-dimensional nature of the mean and dispersion values $(\boldsymbol{\mu}, \boldsymbol{\phi})$ makes a direct comparison across the 14 TCGA datasets impractical. Instead, one or more summary measures are needed to characterize their differences. For further analysis, we focus on the median dispersion of each dataset, calculated from the set of dispersion estimates that serve as the basis for drawing $\boldsymbol{\phi}$ in the simulation—that is, the dispersion values from all genes with a mean expression greater than 10. The median dispersion across datasets ranges from 0.161 to 0.451, with the median values for each dataset shown in Figure 3. Notably, the KIRC dataset, which is used in the original simulation by B20, has one of the lowest median dispersion values at 0.174.

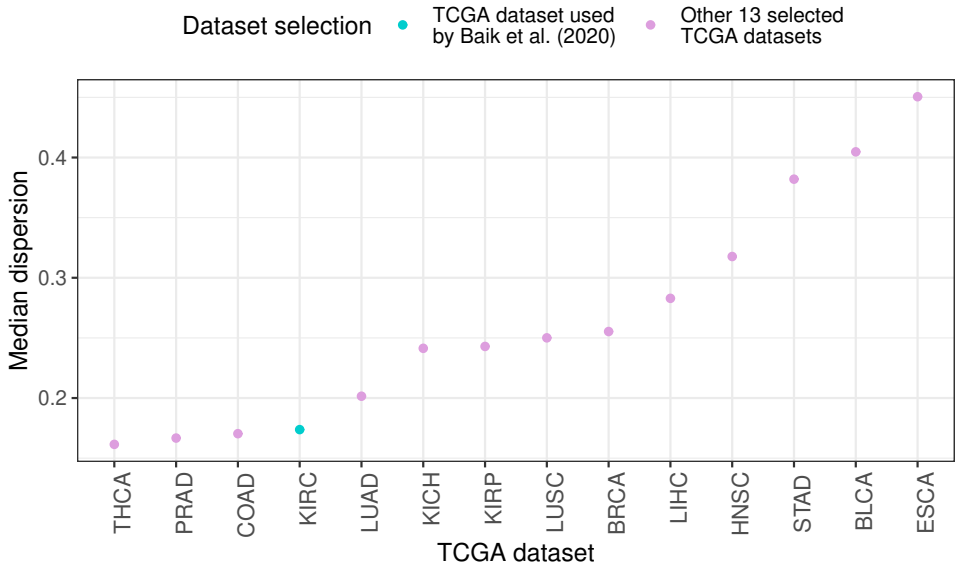

Figure 3: Median dispersion values across the 14 selected TCGA datasets, including the KIRC dataset used by Baik et al. (2020), sorted in ascending order. Median dispersion is calculated based on genes with a mean expression greater than 10, which serve as the basis for drawing dispersion values in the simulation. Dataset abbreviations are detailed in Table S3.

Method performance In Figure 4, the absolute performance (panel a: AUC) and the relative performance (panel b: difference to the best, i.e. the highest AUC within each repetition) are shown in relation to the median dispersion estimated from each TCGA dataset. To improve clarity, only 7 of the 11 considered methods are displayed—those recommended by B20 for the DGMs under consideration (see B20, Table 2). Similarly, Figure 4 only includes results for two of the four specified values of the proportion of DE genes ($p_{\text{DE}} \in \{0.05, 0.30\}$), as these values are used by B20 to differentiate recommendations in their summary table. The results for all methods and p_{DE} values are provided in Supplementary Section B.3 but lead to similar conclusions as those discussed below.

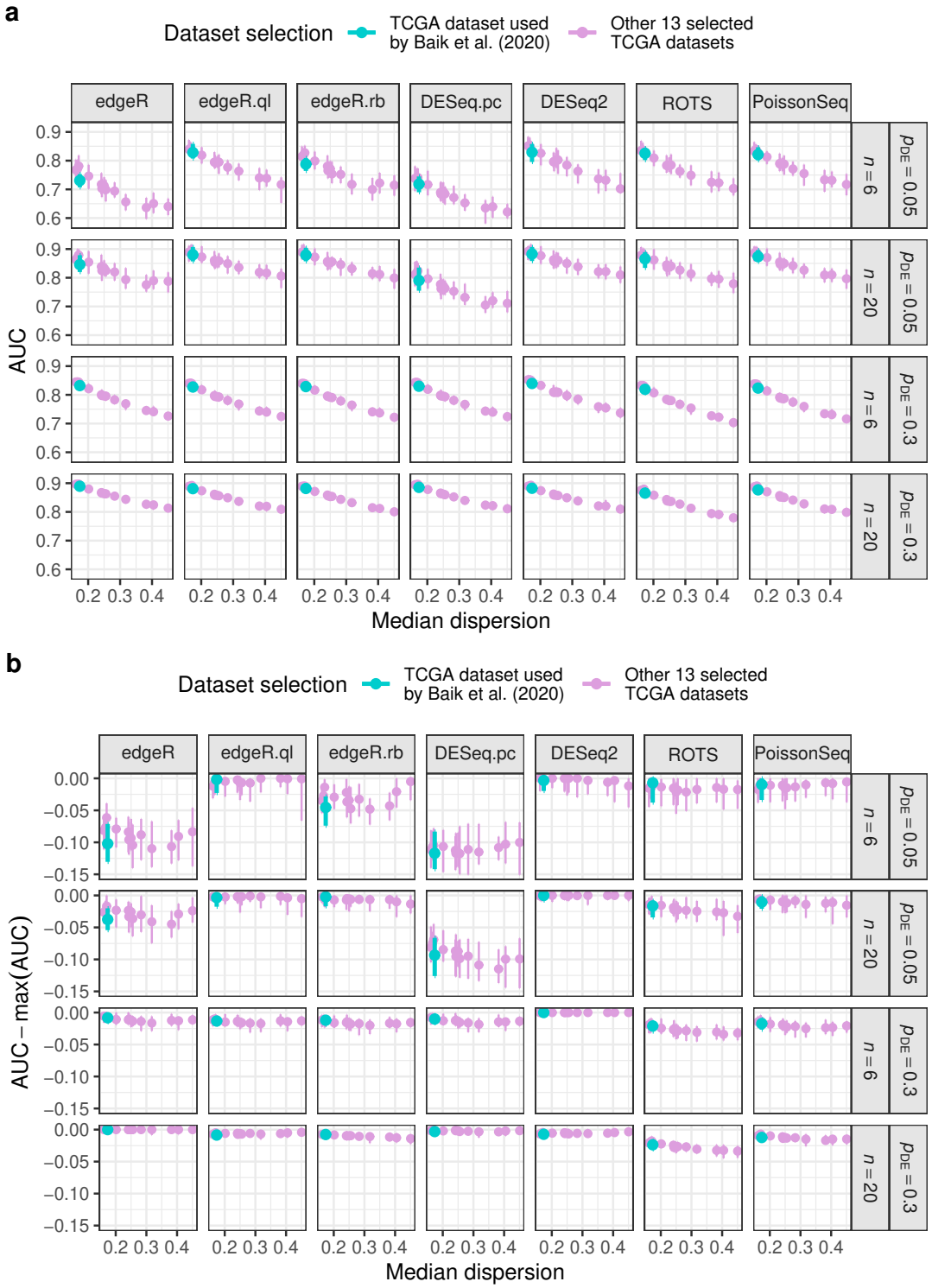

Figure 4: Absolute (panel a) and relative (panel b) performance of six differential expression analysis methods in relation to the median dispersion (averaged across all genes in the real datasets after filtering), across different sample sizes n and proportions of DE genes, p_{DE}, comparing results based on the KIRC dataset used by Baik et al. (2020) and the results based on 13 other selected TCGA datasets. Panel a displays the median and range of absolute AUC values, while panel b presents the median and range of the difference between the AUC and the highest AUC observed within each DGM.

As shown in Figure 4a, regardless of the number of samples and the proportion of DE genes, the AUC of all methods decreases as median dispersion increases. Since the KIRC dataset is among the datasets with the smallest median dispersion, relying solely on this specific dataset for simulation may overestimate the ability of all methods to identify DE genes. With regard to relative performance (Figure 4b), if for a given n and p_{DE} a method is among the top-performing methods when considering only the KIRC dataset (i.e. its AUC difference is close to zero), then its relative performance tends to remain stable across the remaining 13 datasets. However, if a method is not among the best (particularly edgeR, edgeR.rb, and DESeq.pc under DGMs with $p_{\text{DE}} = 0.05$), its performance ranking may change more substantially when additional datasets are considered. While expanding the analysis beyond the KIRC dataset does not lead to different conclusions for all methods, we argue that confirming the stability of relative performance across datasets also provides valuable insight.

5 Suggested workflow

Table 5 provides a structured workflow for constructing parametric DGMs based on a systematically selected set of real datasets, summarizing the considerations discussed in Section 3. Once the real-data-based DGMs are fully specified, the simulation study can proceed as usual. When reporting on this process, the dataset selection and inference steps should be documented in detail, including the database used, eligibility criteria applied, inference methods employed, the mapping of inferred parameter values to DGMs, and any additional information necessary for reproducibility. Finally, as generally recommended in simulation studies, an overview of the resulting DGMs should be provided.

6 Conclusion

Basing parametric simulations on real data can be a viable approach to improve the practical relevance of DGMs, ideally achieving alignment with real-world DGMs in the study's domain of interest. However, current implementations often lack a systematic approach—both in determining which components of the DGM should be based on real data and how they should be inferred, as well as in selecting the real datasets. In particular, the rationale behind dataset selection and the domain of interest they are meant to represent is often unclear, and the number of utilized datasets is typically very limited. As a result, the findings of simulation studies using real-data-based DGMs may not necessarily generalize better to practical applications than those based on fully researcher-specified DGMs, despite potentially creating that impression.

To address these issues, this paper provided a detailed discussion on the construction of real-data-based parametric DGMs, aiming to support researchers in assessing the possibilities and implications of inferring specific DGM components from real datasets and in making dataset selection more systematic. In addition to the formal discussion, we conducted two simulation studies demonstrating the implementation of parametric DGMs based on a systematically selected set of datasets and illustrating that they may lead to different conclusions than fully researcher-specified DGMs or DGMs based on a single real dataset.

Table 5: Structured workflow for constructing parametric DGMs based on a systematically selected set of real datasets.

		Section
Step 1: Apply specification-based and knowledge-based differentiation and plan for variation of researcher-specified components.		
1.1	Determine which components are researcher-specified components of interest, researcher-specified components of convenience, and real-data-based components.	2.2.1
1.2	For researcher-specified components of interest and real-data-based components, clarify whether their true form or value in the true DGM is known, as unknown components introduce uncertainty for dataset selection and inference, respectively.	2.2.2
1.3	Decide whether multiple options/values should be considered for any researcher-specified component of interest. If so, repeat steps 2–7 for each relevant combination. For researcher-specified components of convenience, multiple options/values can be considered without repeating the full process; only the combination with the inferred real-data-based components needs to be specified.	2.3
Step 2: Specify researcher-specified components and additional constraints.		
2.1	Specify the researcher-specified components of interest (one option/value per component; see step 1.3), the researcher-specified components of convenience, and any explicit or implicit constraints for the real-data-based components.	2.2.1
Step 3: Specify inference procedures for real-data-based DGM components.		
3.1	Specify the inference method for each real-data-based parameter or part of the model structure, ensuring that it accounts for potential misalignment between the considered and the true model structure as well as for sampling variability.	3.1, 3.2
3.2	For parameters, specify for each one whether its values will be directly inferred or aggregately inferred and specify how the sets of inferred values for the individual parameters will be mapped to the set of considered parameter vectors, acknowledging the trade-off between realism and greater control or practical feasibility, as implied by the one-to-one approach versus the deviations from it.	3.1
Step 4: Specify the systematic selection of real datasets.		
4.1	Specify eligibility criteria addressing (D2) by translating the researcher-specified components of interest and constraints imposed on real-data-based components (step 2.1) into concrete criteria. Where not directly evident, also specify how the fulfillment of these criteria should be assessed.	3.3.2
4.2	Specify eligibility criteria addressing (D3) based on the inference procedure specified in step 3.1 and the criteria specified for (D2) in step 4.1.	3.3.2
4.3	Specify a minimum and maximum number of datasets to be selected.	3.3.1
4.4	Select a database that meets accessibility requirements to fulfill (D1) and is likely to contain datasets eligible with respect to the criteria specified in steps 4.1 and 4.2.	3.3.1
Step 5: Conduct dataset selection and adjust criteria if necessary.		
5.1	Apply the eligibility criteria to the datasets in the database, refining or extending the criteria during the assessment as needed.	3.3.2
Step 6: Check the number of selected datasets.		
6.1	If the number of selected datasets falls within the predefined range, proceed with step 7; if it is too low, expand the database (e.g., more publication years); if it is too high, randomly draw the specified maximum number of datasets.	3.3.1
Step 7: Infer real-data-based DGM components.		
7.1	If only parameters are inferred, apply the specified inference procedure to the R selected datasets. If parts of the model structure are also inferred, determine them first, then infer parameters separately for each subset of datasets corresponding to each resulting model structure derived from the inference of real-data-based parts.	3.1, 3.2
7.2	Check the resulting DGMs for plausibility.	3.1

Importantly, throughout the paper, several practical and conceptual limitations associated with constructing real-data-based parametric DGMs as we suggest have emerged. One such limitation is finding an adequate database. It is true that parametric DGMs offer more potential database options than, for example, semi-parametric DGMs, which require access to complete datasets, and certain issues—such as dataset quality—can be mitigated through carefully chosen eligibility criteria. However, even with the most meticulous planning, one major limitation remains: some datasets relevant to the domain of interest may not be included in the database at all (e.g., if the database primarily contains datasets from a specific subpopulation of DGMs, a restriction that is not always immediately apparent). Additionally, it is often unclear whether the selected datasets truly belong to the domain of interest, as some eligibility criteria may refer to DGM components whose true form or values in the underlying real-world DGM cannot be determined solely from the real dataset itself. As discussed in detail, this issue arises not only when datasets are selected based on DGM components that cannot be fully known but also when these components are inferred from the selected datasets to construct the DGM. Specifically, in the case of parameters, even the one-to-one inference approach does not guarantee that the inferred values are close to the true ones. Similarly, if parts of the model structure are also intended to be based on real data, the specified set of possible options (e.g., distributions) may already deviate substantially from the true underlying structure (which may not even have a closed-form representation).

As a consequence, there is no guarantee that the suggested approach will lead to practically relevant parametric DGMs—and, at the same time, it demands substantially more effort than simply relying on a single convenience dataset or entirely researcher-specified DGMs. However, we propose the following considerations. First, the increased effort is worthwhile in itself because it encourages researchers to think more deeply about the simulation design. Moreover, reporting how and why the real datasets were selected and how the DGMs were constructed based on them enhances transparency and ideally provides a clearer understanding of the study's domain of interest. In principle, this follows a similar line of reasoning as other (complementary) approaches aimed at improving thoroughness and transparency, such as the writing (and potential preregistration) of research protocols for simulation studies (Siepe et al., 2024), which, despite seeming like an unnecessary burden, already improves study quality by requiring careful consideration and documentation of decisions.

Regarding the extent to which practical relevance can be achieved, we argue that pursuing it should not be abandoned just because it cannot be fully attained—especially since, also due to our certainly idealized definition, it may never be completely achievable in the first place. Moreover, our proposed approach would likely lead to more practically relevant parametric DGMs than the alternatives currently used in practice, i.e. fully researcher-specified parametric DGMs or DGMs based on one or two convenience datasets. Of course, a rigorous discussion of the limitations that hinder the achievement of practical relevance remains essential. This is particularly important to prevent the mere use of a larger-than-usual number of real datasets from being misinterpreted—by readers or even the researchers conducting the study—as a guarantee of practical relevance.

Finally, it is worth reiterating that while this paper focused on parametric DGMs, researchers aiming for practically relevant DGMs may also consider semi-parametric DGMs, for which the use of a systematic selection of real datasets is also uncommon, just as it is for parametric DGMs. Although our discussion is partially applicable to semi-parametric DGMs, a more detailed investigation specifically tailored to these cases would be valuable for future research. Moreover, our approach for constructing real-data-based parametric DGMs does not address all potential pitfalls in simulation studies that may contribute to overgeneralization and misinterpretation—for example, biased post hoc selection of performance measures or considered methods (Pawel et al., 2024). While these issues fall outside the scope of our approach, they underscore the need for careful study design beyond just the choice of DGM. Still, we hope that the proposed approach to constructing real-data-based parametric DGMs is a step toward simulation studies that yield more well-founded recommendations, ultimately helping applied researchers make more informed choices when selecting statistical methods.

Funding information

This work was supported by the German Research Foundation (individual grants BO3139/7-2 and BO3139/9-1 to ALB for the work of CS and project 352692197 for the work of MT). The authors of this work take full responsibility for its content.

Acknowledgments

We thank Luzia Hanßum for assistance with supporting tasks related to the manuscript.

Conflicts of interest

The authors have declared no conflicts of interest for this article.

References

Agresti, A. (2010). *Analysis of ordinal categorical data* (2nd). John Wiley & Sons. https://doi.org/10.1002/9780470594001

Albers, G. W., Marks, M. P., Kemp, S., Christensen, S., Tsai, J. P., Ortega-Gutierrez, S., McTaggart, R. A., Torbey, M. T., Kim-Tenser, M., Leslie-Mazwi, T., Sarraj, A., Kasner, S. E., Ansari, S. A., Yeatts, S. D., Hamilton, S., Mlynash, M., Heit, J. J., Zaharchuk, G., Kim, S., ... Lansberg, M. G. (2018). Thrombectomy for stroke at 6 to 16 hours with selection by perfusion imaging. *New England Journal of Medicine*, *378*(8), 708–718. https://doi.org/10.1056/NEJMoa1713973

Ali, M., Bath, P. M. W., Curram, J., Davis, S. M., Diener, H.-C., Donnan, G. A., Fisher, M., Gregson, B. A., Grotta, J., Hacke, W., Hennerici, M. G., Hommel, M., Kaste, M., Marler, J. R., Sacco, R. L., Teal, P., Wahlgren, N.-G., Warach, S., Weir, C. J., & Lees, K. R. (2007). The Virtual International Stroke Trials Archive. *Stroke*, *38*(6), 1905–1910. https://doi.org/10.1161/STROKEAHA.106.473579

Astivia, O. L. O., & Zumbo, B. D. (2015). A cautionary note on the use of the Vale and Maurelli method to generate multivariate, nonnormal data for simulation purposes. *Educational and Psychological Measurement*, *75*(4), 541–567. https://doi.org/10.1177/0013164414548894

Baik, B., Yoon, S., & Nam, D. (2020). Benchmarking RNA-seq differential expression analysis methods using spike-in and simulation data. *PLOS ONE*, *15*(4), e0232271. https://doi.org/10.1371/journal.pone.0232271

Benidt, S., & Nettleton, D. (2015). SimSeq: A nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics*, *31*(13), 2131–2140. https://doi.org/10.1093/bioinformatics/btv124

Bierer, B. E., Li, R., Barnes, M., & Sim, I. (2016). A global, neutral platform for sharing trial data. *New England Journal of Medicine*, *374*(25), 2411–2413. https://doi.org/10.1056/NEJMp1605348

Bono, R., Blanca, M. J., Arnau, J., & Gómez-Benito, J. (2017). Non-normal distributions commonly used in health, education, and social sciences: A systematic review. *Frontiers in Psychology*, *8*, 1602. https://doi.org/10.3389/fpsyg.2017.01602

Boulesteix, A.-L., Groenwold, R. H. H., Abrahamowicz, M., Binder, H., Briel, M., Hornung, R., Morris, T. P., Rahnenführer, J., & Sauerbrei, W. (2020). Introduction to statistical simulations in health research. *BMJ Open*, *10*(12), e039921. https://doi.org/10.1136/bmjopen-2020-039921

Boulesteix, A.-L., Hable, R., Lauer, S., & Eugster, M. J. A. (2015). A statistical framework for hypothesis testing in real data comparison studies. *The American Statistician*, *69*(3), 201–212. https://doi.org/10.1080/00031305.2015.1005128

Boulesteix, A.-L., Wilson, R., & Hapfelmeier, A. (2017). Towards evidence-based computational statistics: Lessons from clinical research on the role and design of real-data benchmark studies. *BMC Medical Research Methodology*, *17*, 138. https://doi.org/10.1186/s12874-017-0417-2

Brunner, E., Vandemeulebroecke, M., & Mütze, T. (2021). Win odds: An adaptation of the win ratio to include ties. *Statistics in Medicine*, *40*(14), 3367–3384. https://doi.org/10.1002/sim.8967

Burton, A., Altman, D. G., Royston, P., & Holder, R. L. (2006). The design of simulation studies in medical statistics. *Statistics in Medicine*, *25*(24), 4279–4292. https://doi.org/10.1002/sim.2673

Campbell, B. C. V., Mitchell, P. J., Churilov, L., Yassi, N., Kleinig, T. J., Dowling, R. J., Yan, B., Bush, S. J., Dewey, H. M., Thijs, V., Scroop, R., Simpson, M., Brooks, M., Asadi, H., Wu, T. Y., Shah, D. G., Wijeratne, T., Ang, T., Miteff, F., ... Davis, S. M. (2018). Tenecteplase versus alteplase before thrombectomy for ischemic stroke. *New England Journal of Medicine*, *378*(17), 1573–1582. https://doi.org/10.1056/NEJMoa1716405

Cavalcanti, A. B., Zampieri, F. G., Rosa, R. G., Azevedo, L. C. P., Veiga, V. C., Avezum, A., Damiani, L. P., Marcadenti, A., Kawano-Dourado, L., Lisboa, T., Junqueira, D. L. M., de Barros e Silva, P. G., Tramujas, L., Abreu-Silva, E. O., Laranjeira, L. N., Soares, A. T.,

Echenique, L. S., Pereira, A. J., Freitas, F. G. R., … Berwanger, O. (2020). Hydroxychloroquine with or without azithromycin in mild-to-moderate Covid-19. *New England Journal of Medicine*, *383*(21), 2041–2052. https://doi.org/10.1056/NEJMoa2019014

Chipman, H., & Bingham, D. (2022). Let's practice what we preach: Planning and interpreting simulation studies with design and analysis of experiments. *Canadian Journal of Statistics*, *50*(4), 1228–1249. https://doi.org/10.1002/cjs.11719

Dormuth, I., Liu, T., Xu, J., Pauly, M., & Ditzhaus, M. (2023). A comparative study to alternatives to the log-rank test. *Contemporary Clinical Trials*, *128*, 107165. https://doi.org/10.1016/j.cct.2023.107165

Dormuth, I., Liu, T., Xu, J., Yu, M., Pauly, M., & Ditzhaus, M. (2022). Which test for crossing survival curves? A user's guideline. *BMC Medical Research Methodology*, *22*, 34. https://doi.org/10.1186/s12874-022-01520-0

Fairchild, A. J., Yin, Y., Baraldi, A. N., Astivia, O. L. O., & Shi, D. (2024). Many nonnormalities, one simulation: Do different data generation algorithms affect study results? *Behavior Research Methods*, *56*, 6464–6484. https://doi.org/10.3758/s13428-024-02364-w

Fernández-Castilla, B., Jamshidi, L., Declercq, L., Beretvas, S. N., Onghena, P., & Van den Noortgate, W. (2020). The application of meta-analytic (multi-level) models with multiple random effects: A systematic review. *Behavior Research Methods*, *52*(5), 2031–2052. https://doi.org/10.3758/s13428-020-01373-9

Franklin, J. M., Schneeweiss, S., Polinski, J. M., & Rassen, J. A. (2014). Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases. *Computational Statistics & Data Analysis*, *72*, 219–226. https://doi.org/10.1016/j.csda.2013.10.018

Friedrich, S., & Friede, T. (2024). On the role of benchmarking data sets and simulations in method comparison studies. *Biometrical Journal*, *66*(1), 2200212. https://doi.org/10.1002/bimj.202200212

Funatogawa, I., & Funatogawa, T. (2023). Comparison of profile-likelihood-based confidence intervals with other rank-based methods for the two-sample problem in ordered categorical data. *Journal of Biopharmaceutical Statistics*, *33*(3), 371–385. https://doi.org/10.1080/10543406.2022.2152831

Goldman, J. D., Lye, D. C. B., Hui, D. S., Marks, K. M., Bruno, R., Montejano, R., Spinner, C. D., Galli, M., Ahn, M.-Y., Nahass, R. G., Chen, Y.-S., SenGupta, D., Hyland, R. H., Osinusi, A. O., Cao, H., Blair, C., Wei, X., Gaggar, A., Brainard, D. M., … Subramanian, A. (2020). Remdesivir for 5 or 10 days in patients with severe Covid-19. *New England Journal of Medicine*, *383*(19), 1827–1837. https://doi.org/10.1056/NEJMoa2015301

Guevara Morel, A. E., Varga, A. N., Heymans, M. W., Dongen, J. M., Schaik, D. J. F., Tulder, M. W., & Bosmans, J. E. (2022). Dealing with missing data in real-world data: A scoping review of simulation studies. *Preprint (version 1) available at Research Square*. https://doi.org/10.21203/rs.3.rs-1619388/v1

Guyot, P., Ades, A. E., Ouwens, M. J. N. M., & Welton, N. J. (2012). Enhanced secondary analysis of survival data: Reconstructing the data from published Kaplan-Meier survival

curves. *BMC Medical Research Methodology*, *12*, 9. https://doi.org/10.1186/1471-2288-12-9

Harwell, M., Kohli, N., & Peralta, Y. (2017). Experimental design and data analysis in computer simulation studies in the behavioral sciences. *Journal of Modern Applied Statistical Methods*, *16*(2), 3–28. https://doi.org/10.22237/jmasm/1509494520

Heinze, G., Boulesteix, A.-L., Kammer, M., Morris, T. P., & White, I. R. (2024). Phases of methodological research in biostatistics—Building the evidence base for new methods. *Biometrical Journal*, *66*(1), 2200222. https://doi.org/10.1002/bimj.202200222

Herrmann, M., Lange, F. J. D., Eggensperger, K., Casalicchio, G., Wever, M., Feurer, M., Rügamer, D., Hüllermeier, E., Boulesteix, A.-L., & Bischl, B. (2024). Position: Why we must rethink empirical research in machine learning. *Proceedings of the 41st International Conference on Machine Learning*, 18228–18247. https://proceedings.mlr.press/v235/herrmann24b.html

Hothorn, T., Leisch, F., Zeileis, A., & Hornik, K. (2005). The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, *14*(3), 675–699. https://doi.org/10.1198/106186005X59630

Hutchinson, P. J., Edlmann, E., Bulters, D., Zolnourian, A., Holton, P., Suttner, N., Agyemang, K., Thomson, S., Anderson, I. A., Al-Tamimi, Y. Z., Henderson, D., Whitfield, P. C., Gherle, M., Brennan, P. M., Allison, A., Thelin, E. P., Tarantino, S., Pantaleo, B., Caldwell, K., ... Kolias, A. G. (2020). Trial of dexamethasone for chronic subdural hematoma. *New England Journal of Medicine*, *383*(27), 2616–2627. https://doi.org/10.1056/NEJMoa2020473

Jansen, K., & Holling, H. (2023). Random-effects meta-analysis models for the odds ratio in the case of rare events under different data-generating models: A simulation study. *Biometrical Journal*, *65*(3), 2200132. https://doi.org/10.1002/bimj.202200132

Jovin, T. G., Li, C., Wu, L., Wu, C., Chen, J., Jiang, C., Shi, Z., Gao, Z., Song, C., Chen, W., Peng, Y., Yao, C., Wei, M., Li, T., Wei, L., Xiao, G., Yang, H., Ren, M., Duan, J., ... Ji, X. (2022). Trial of thrombectomy 6 to 24 hours after stroke due to basilar-artery occlusion. *New England Journal of Medicine*, *387*(15), 1373–1384. https://doi.org/10.1056/NEJMoa2207576

Kelly, M., Longjohn, R., & Nottingham, K. (n.d.). The UCI Machine Learning Repository [Accessed on December 19, 2024]. https://archive.ics.uci.edu

Kulinskaya, E., Hoaglin, D. C., & Bakbergenuly, I. (2021). Exploring consequences of simulation design for apparent performance of methods of meta-analysis. *Statistical Methods in Medical Research*, *30*(7), 1667–1690. https://doi.org/10.1177/09622802211013065

Langan, D., Higgins, J. P. T., & Simmonds, M. (2017). Comparative performance of heterogeneity variance estimators in meta-analysis: A review of simulation studies. *Research Synthesis Methods*, *8*(2), 181–198. https://doi.org/10.1002/jrsm.1198

Langan, D., Higgins, J. P., Jackson, D., Bowden, J., Veroniki, A. A., Kontopantelis, E., Viechtbauer, W., & Simmonds, M. (2019). A comparison of heterogeneity variance estimators

in simulated random-effects meta-analyses. *Research Synthesis Methods*, *10*(1), 83–98. https://doi.org/10.1002/jrsm.1316

LeCouffe, N. E., Kappelhof, M., Treurniet, K. M., Rinkel, L. A., Bruggeman, A. E., Berkhemer, O. A., Wolff, L., van Voorst, H., Tolhuisen, M. L., Dippel, D. W., van der Lugt, A., van Es, A. C. G. M., Boiten, J., Lycklama à Nijeholt, G. J., Keizer, K., Gons, R. A. R., Yo, L. S. F., van Oostenbrugge, R. J., van Zwam, W. H., ... Roos, Y. B. W. E. M. (2021). A randomized trial of intravenous alteplase before endovascular treatment for stroke. *New England Journal of Medicine*, *385*(20), 1833–1844. https://doi.org/10.1056/NEJMoa2107727

Li, J., & Tibshirani, R. (2013). Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data. *Statistical Methods in Medical Research*, *22*(5), 519–536. https://doi.org/10.1177/0962280211428386

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. https://doi.org/10.1186/s13059-014-0550-8

Ma, H., Campbell, B. C. V., Parsons, M. W., Churilov, L., Levi, C. R., Hsu, C., Kleinig, T. J., Wijeratne, T., Curtze, S., Dewey, H. M., Miteff, F., Tsai, C.-H., Lee, J.-T., Phan, T. G., Mahant, N., Sun, M.-C., Krause, M., Sturm, J., Grimley, R., ... Donnan, G. A. (2019). Thrombolysis guided by perfusion imaging up to 9 hours after onset of stroke. *New England Journal of Medicine*, *380*(19), 1795–1803. https://doi.org/10.1056/NEJMoa1813046

Martins, S. O., Mont'Alverne, F., Rebello, L. C., Abud, D. G., Silva, G. S., Lima, F. O., Parente, B. S. M., Nakiri, G. S., Faria, M. B., Frudit, M. E., de Carvalho, J. J. F., Waihrich, E., Fiorot, J. A., Cardoso, F. B., Hidalgo, R. C. T., Zétola, V. F., Carvalho, F. M., de Souza, A. C., Dias, F. A., ... Nogueira, R. G. (2020). Thrombectomy for stroke in the public health care system of Brazil. *New England Journal of Medicine*, *382*(24), 2316–2326. https://doi.org/10.1056/NEJMoa2000120

Metcalfe, C., & Thompson, S. G. (2006). The importance of varying the event generation process in simulation studies of statistical methods for recurrent events. *Statistics in Medicine*, *25*(1), 165–179. https://doi.org/10.1002/sim.2310

Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, *38*(11), 2074–2102. https://doi.org/10.1002/sim.8086

Nießl, C., Hoffmann, S., Ullmann, T., & Boulesteix, A.-L. (2024). Explaining the optimistic performance evaluation of newly proposed methods: A cross-design validation experiment. *Biometrical Journal*, *66*(1), 2200238. https://doi.org/10.1002/bimj.202200238

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, *372*, n71. https://doi.org/10.1136/bmj.n71

Pateras, K., Nikolakopoulos, S., & Roes, K. (2018). Data-generating models of dichotomous outcomes: Heterogeneity in simulation studies for a random-effects meta-analysis. *Statistics in Medicine*, *37*(7), 1115–1124. https://doi.org/10.1002/sim.7569

Pawel, S., Kook, L., & Reeve, K. (2024). Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method. *Biometrical Journal*, *66*(1), 2200091. https://doi.org/10.1002/bimj.202200091

Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., & Chen, F. (2001). Monte Carlo experiments: Design and implementation. *Structural Equation Modeling: A Multidisciplinary Journal*, *8*(2), 287–312. https://doi.org/10.1207/s15328007sem0802_7

Pénichoux, J., Moreau, T., & Latouche, A. (2015). Simulating recurrent events that mimic actual data: A review of the literature with emphasis on event-dependence. *arXiv:1503.05798 [stat.AP]*. https://doi.org/10.48550/arXiv.1503.05798

Perkins, G. D., Ji, C., Deakin, C. D., Quinn, T., Nolan, J. P., Scomparin, C., Regan, S., Long, J., Slowther, A., Pocock, H., Black, J. J. M., Moore, F., Fothergill, R. T., Rees, N., O'Shea, L., Docherty, M., Gunson, I., Han, K., Charlton, K., ... Lall, R. (2018). A randomized trial of epinephrine in out-of-hospital cardiac arrest. *New England Journal of Medicine*, *379*(8), 711–721. https://doi.org/10.1056/NEJMoa1806842

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Ramos, M., Geistlinger, L., Oh, S., Schiffer, L., Azhar, R., Kodali, H., de Bruijn, I., Gao, J., Carey, V. J., Morgan, M., & Waldron, L. (2020). Multiomic integration of public oncology databases in Bioconductor. *JCO Clinical Cancer Informatics*, *1*(4), 958–971. https://doi.org/10.1200/CCI.19.00119

Ramos, M., Schiffer, L., Re, A., Azhar, R., Basunia, A., Rodriguez, C., Chan, T., Chapman, P., Davis, S. R., Gomez-Cabrero, D., Culhane, A. C., Haibe-Kains, B., Hansen, K. D., Kodali, H., Louis, M. S., Mer, A. S., Riester, M., Morgan, M., Carey, V., & Waldron, L. (2017). Software for the integration of multiomics experiments in Bioconductor. *Cancer Research*, *77*(21), e39–e42. https://doi.org/10.1158/0008-5472.CAN-17-0344

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. https://doi.org/10.1093/bioinformatics/btp616

Rosas, I. O., Bräu, N., Waters, M., Go, R. C., Hunter, B. D., Bhagani, S., Skiest, D., Aziz, M. S., Cooper, N., Douglas, I. S., Savic, S., Youngstein, T., Sorbo, L. D., Gracian, A. C., De La Zerda, D. J., Ustianowski, A., Bao, M., Dimonaco, S., Graham, E., ... Malhotra, A. (2021). Tocilizumab in hospitalized patients with severe Covid-19 pneumonia. *New England Journal of Medicine*, *384*(16), 1503–1516. https://doi.org/10.1056/NEJMoa2028700

Ross, J. S., Waldstreicher, J., Bamford, S., Berlin, J. A., Childers, K., Desai, N. R., Gamble, G., Gross, C. P., Kuntz, R., Lehman, R., Lins, P., Morris, S. A., Ritchie, J. D., & Krumholz,

H. M. (2018). Overview and experience of the YODA Project with clinical trial data sharing after 5 years. *Scientific Data*, *5*, 180268. https://doi.org/10.1038/sdata.2018.268

Royston, P., Choodari-Oskooei, B., Parmar, M. K. B., & Rogers, J. K. (2019). Combined test versus logrank/Cox test in 50 randomised trials. *Trials*, *20*, 172. https://doi.org/10.1186/s13063-019-3251-5

Schreck, N., Slynko, A., Saadati, M., & Benner, A. (2024). Statistical plasmode simulations–Potentials, challenges and recommendations. *Statistics in Medicine*, *43*(9), 1804–1825. https://doi.org/https://doi.org/10.1002/sim.10012

Selman, C. J., Lee, K. J., Ferguson, K. N., Whitehead, C. L., Manley, B. J., & Mahar, R. K. (2024). Statistical analyses of ordinal outcomes in randomised controlled trials: A scoping review. *Trials*, *25*, 241. https://doi.org/10.1186/s13063-024-08072-2

Sidik, K., & Jonkman, J. N. (2005). Simple heterogeneity variance estimation for meta-analysis. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, *54*(2), 367–384. https://doi.org/10.1111/j.1467-9876.2005.00489.x

Siepe, B. S., Bartoš, F., Morris, T. P., Boulesteix, A.-L., Heck, D. W., & Pawel, S. (2024). Simulation studies for methodological research in psychology: A standardized template for planning, preregistration, and reporting. *Psychological Methods*. https://doi.org/10.1037/met0000695

Simmons, J., Nelson, L., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, *22*(11), 1359–1366. https://doi.org/10.1177/0956797611417632

Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., & Bommert, A. (2024). Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression. *PLOS ONE*, *19*(5), e0299989. https://doi.org/10.1371/journal.pone.0299989

Strobl, C., & Leisch, F. (2024). Against the "one method fits all data sets" philosophy for comparison studies in methodological research. *Biometrical Journal*, *66*(1), 2200104. https://doi.org/0.1002/bimj.202200104

Tao, C., Nogueira, R. G., Zhu, Y., Sun, J., Han, H., Yuan, G., Wen, C., Zhou, P., Chen, W., Zeng, G., Li, Y., Ma, Z., Yu, C., Su, J., Zhou, Z., Chen, Z., Liao, G., Sun, Y., Ren, Y., … Hu, W. (2022). Trial of endovascular treatment of acute basilar-artery occlusion. *New England Journal of Medicine*, *387*(15), 1361–1372. https://doi.org/10.1056/NEJMoa2206317

Thas, O. (2010). *Comparing distributions*. Springer. https://doi.org/10.1007/978-0-387-92710-7

Thomalla, G., Simonsen, C. Z., Boutitie, F., Andersen, G., Berthezene, Y., Cheng, B., Cheripelli, B., Cho, T.-H., Fazekas, F., Fiehler, J., Ford, I., Galinovic, I., Gellissen, S., Golsari, A., Gregori, J., Günther, M., Guibernau, J., Häusler, K. G., Hennerici, M., … Gerloff, C. (2018). MRI-guided thrombolysis for stroke with unknown time of onset. *New England Journal of Medicine*, *379*(7), 611–622. https://doi.org/10.1056/NEJMoa1804355

Thurow, M., Dormuth, I., Sauer, C., Ditzhaus, M., & Pauly, M. (2024). How to simulate realistic survival data? A simulation study to compare realistic simulation models. *arXiv:2308.07842v2 [stat.AP]*. https://doi.org/10.48550/arXiv.2308.07842

Trinquart, L., Jacot, J., Conner, S. C., & Porcher, R. (2016). Comparison of treatment effects measured by the hazard ratio and by the ratio of restricted mean survival times in oncology randomized controlled trials. *Journal of Clinical Oncology*, *34*(15), 1813–1819. https://doi.org/10.1200/JCO.2015.64.2488

van den Berg, L. A., Dijkgraaf, M. G. W., Berkhemer, O. A., Fransen, P. S. S., Beumer, D., Lingsma, H. F., Majoie, C. B. L. M., Dippel, D. W. J., van der Lugt, A., van Oostenbrugge, R. J., van Zwam, W. H., & Roos, Y. B. W. E. M. (2017). Two-year outcome after endovascular treatment for acute ischemic stroke. *New England Journal of Medicine*, *376*(14), 1341–1349. https://doi.org/10.1056/NEJMoa1612136

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2014). OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, *15*(2), 49–60. https://doi.org/10.1145/2641190.2641198

Welvaert, M., & Rosseel, Y. (2014). A review of fMRI simulation studies. *PLoS ONE*, *9*(7), e101953. https://doi.org/10.1371/journal.pone.0101953

White, I. R. (2023). The importance of plausible data-generating mechanisms in simulation studies: A response to 'Comparing methods for handling missing covariates in meta-regression' by Lee and Beretvas (doi: 10.1002/jrsm.1585). *Research Synthesis Methods*, *14*(1), 137–139. https://doi.org/10.1002/jrsm.1605

Yang, P., Zhang, Y., Zhang, L., Zhang, Y., Treurniet, K. M., Chen, W., Peng, Y., Han, H., Wang, J., Wang, S., Yin, C., Liu, S., Wang, P., Fang, Q., Shi, H., Yang, J., Wen, C., Li, C., Jiang, C., ... Liu, J. (2020). Endovascular thrombectomy with or without intravenous alteplase in acute stroke. *New England Journal of Medicine*, *382*(21), 1981–1993. https://doi.org/10.1056/NEJMoa2001123

Supplementary material

A Example 1: Two-arm randomized controlled trial with an ordinal outcome

A.1 Specification of researcher-specified parameters

Table S1: The 4 researcher-specified pairs of outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ considered in the example illustration on hypothesis testing in the context of a two-arm randomized controlled trial with an ordinal outcome (see Section 4.1).

Outcome probabilities ID	$\pi_{1,1}$	$\pi_{2,1}$	$\pi_{3,1}$	$\pi_{4,1}$	$\pi_{5,1}$	$\pi_{6,1}$	$\pi_{7,1}$	$\pi_{1,2}$	$\pi_{2,2}$	$\pi_{3,2}$	$\pi_{4,2}$	$\pi_{5,2}$	$\pi_{6,2}$	$\pi_{7,2}$
k7_id1	0.04	0.07	0.11	0.14	0.18	0.21	0.25	0.14	0.14	0.14	0.14	0.14	0.14	0.14
k7_id2	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.05	0.05	0.07	0.10	0.10	0.28	0.35
k7_id3	0.05	0.05	0.07	0.10	0.10	0.28	0.35	0.05	0.10	0.20	0.30	0.20	0.10	0.05
k7_id4	0.05	0.05	0.20	0.20	0.30	0.10	0.10	0.05	0.10	0.20	0.30	0.20	0.10	0.05

A.2 Specification of real-data-based parameters

A.2.1 Dataset selection

Database As a database, we considered all research publications published in *The New England Journal of Medicine* (https://www.nejm.org/) between 2017 and 2022, which corresponds to the journal's volumes 376–387.

Search string We first identified 270 articles for screening using this search string: `fulltext:"randomized" AND (fulltext:"ordinal" OR fulltext:"proportional-odds" OR fulltext:"Mann{Whitney U" OR fulltext:"Mann-Whitney-Wilcoxon" OR fulltext:"Wilcoxon-Mann-Whitney" OR fulltext:"Wilcoxon rank-sum" OR fulltext:"Chi-Square" OR fulltext:"Fisher") AND (startDate:2017-01-01 AND endDate:2022-12-31) AND (articleCategory:"research")`.

Dataset-level criteria for screening The inclusion and exclusion criteria on the dataset level are listed below, followed by some notes to clarify how we defined ordinal outcomes for our assessment of these criteria. Before each criterion, it is indicated to which of the three proposed requirements for real datasets used to construct real-data-based DGMs (see Section 3.3) the criterion is related.

- Inclusion criteria
  - (D2) Randomized controlled trials
  - (D2) At least one ordinal outcome
- Trial exclusion criteria
  - (D2) Trials where individuals were not randomized individually but in groups or clusters, for example
  - (D3) Trials whose data overlaps with another trial considered at this stage, with preference given to the trial with the larger sample size

- Outcome exclusion criteria
  - (D2) Ordinal outcomes that are non-efficacy outcomes (e.g., safety, procedural, treatment adherence, or health economics outcomes)
  - (D2) Ordinal outcomes that are patient-reported outcomes
  - (D2) Ordinal outcomes analyzed according to anything other than the intention-to-treat principle
  - (D2) Ordinal outcomes that were not analyzed beyond the presentation of frequencies, not analyzed as ordinal variables (e.g., if an ordinal outcome was dichotomized for the analysis), or analyzed with methods inappropriate for ordinal data (e.g., methods for continuous data)
  - (D3) Ordinal outcomes for which the data was not clearly reported, either in tables or figures (in the main/full text or supplement), for all categories
  - (D2) Ordinal outcomes with more or fewer than 7 categories
  - (D3) Ordinal outcomes with empty categories
- Details on the definition of ordinal outcomes we applied when assessing articles with respect to the criteria above:
  - Ordinal outcomes must be explicitly declared as trial outcomes either in the main/full text, supplement, or study protocol to be considered.
  - We considered an outcome variable ordinal if it was a categorical variable with ordered categories that are mutually exclusive and explicitly labeled.
  - If a reported distribution contained a category labeled "could not be evaluated" or "unknown" in addition to otherwise ordinal categories, we did not consider the variable ordinal.
  - Both ordinal outcomes based on an ordinal scale and ordinal outcomes defined by categorizing continuous measures were considered suitable for inclusion.
  - Non-ordinal outcomes involving an ordinal scale/measure, i.e. binary or continuous outcomes based on ordinal scales/scores, such as dichotomized ordinal variables or continuous variables reflecting the change in an ordinal scale/score, were not considered, even if there was data available for the involved ordinal scale/score.

We first assessed for each of the 270 articles identified from the search whether or not it met the two inclusion criteria. 174 articles failed to meet the inclusion criteria, resulting in 96 remaining articles. These were then assessed with respect to the trial exclusion criteria. For two articles, the reported randomized controlled trials met trial exclusion criteria, leaving 94 articles with randomized controlled trials with ordinal outcomes to be assessed with respect to the outcome exclusion criteria. Out of these articles, 79 only had ordinal outcomes that met at least one of the outcome exclusion criteria, resulting in a final number of 15 articles with eligible ordinal outcomes. The screening process of the 270 publications is illustrated in Figure S1 and the spreadsheet documenting the eligibility assessment can be found at https://github.com/NiesslC/realdata_simulations. Note that when assessing the eligibility of trials/outcomes, we did not factor in specifics of the randomization procedure (as long as individuals were randomized

individually), treatment of missing values, or small details regarding the conducted analysis (e.g., covariates or random effects in regression models).

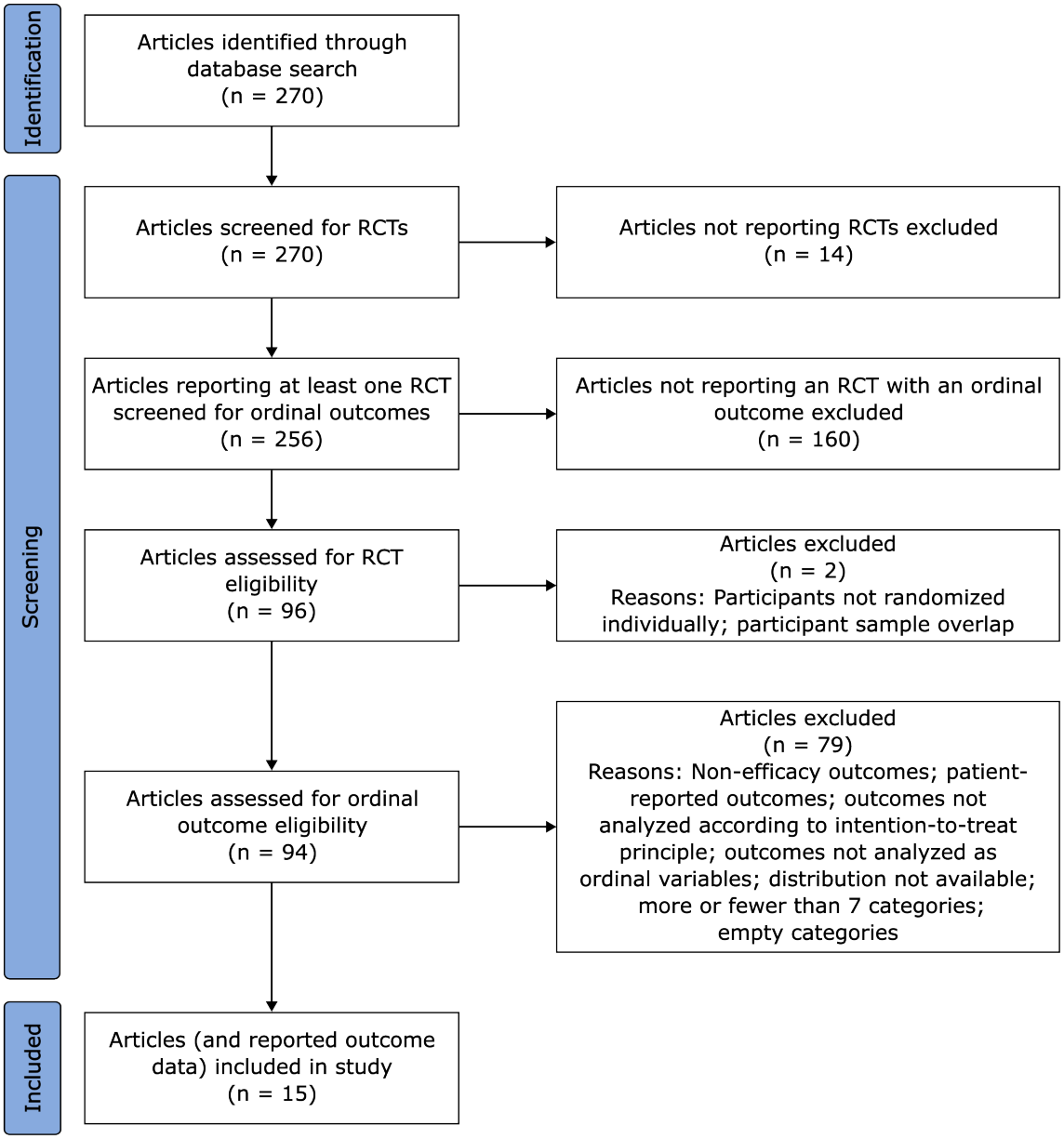

Figure S1: PRISMA flow diagram (Page et al., 2021) to illustrate the dataset-level screening process for the example illustration on hypothesis testing in the context of a two-arm randomized controlled trial with an ordinal outcome (see Section 4.1).

Subset-level criteria For the 15 selected articles, or more specifically, their underlying datasets, we applied the following subset criteria.

- (D2)/(D3) If there are two or more suitable ordinal outcomes in an article/trial, include the outcome that is considered most important in the trial (e.g., prefer primary outcomes to secondary outcomes and prefer secondary outcomes to tertiary/exploratory/additional outcomes). If such a distinction is not possible, include the outcome that has the highest sample size.

- (D2)/(D3) If more than two groups are compared in a trial with a suitable outcome, include the figures for the two groups with the highest sample sizes.

A.2.2 Resulting parameters

Table S2 shows the 15 resulting real-data-based pairs of outcome probabilities. Note that we extracted the data for ordinal outcomes as it was presented in the article, which means that we did not change the order of the categories and extracted the distributions across the categories either in absolute terms (counts) or in relative terms (proportions), whichever was reported. Moreover, some papers reporting the distributions in relative terms included statements such as "percentages may not total 100 because of rounding". If that was the case, we scaled the resulting probabilities to 1.

A.3 Additional results

Parameter characteristics Figure S2 shows an example of a researcher-specified and an example of a real-data-based set of outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$, selected from the four researcher-specified and 15 real-data-based outcome probabilities. As expected, the researcher-specified outcome probabilities are more structured and systematically chosen, whereas those based on real data appear less uniform and more irregular.

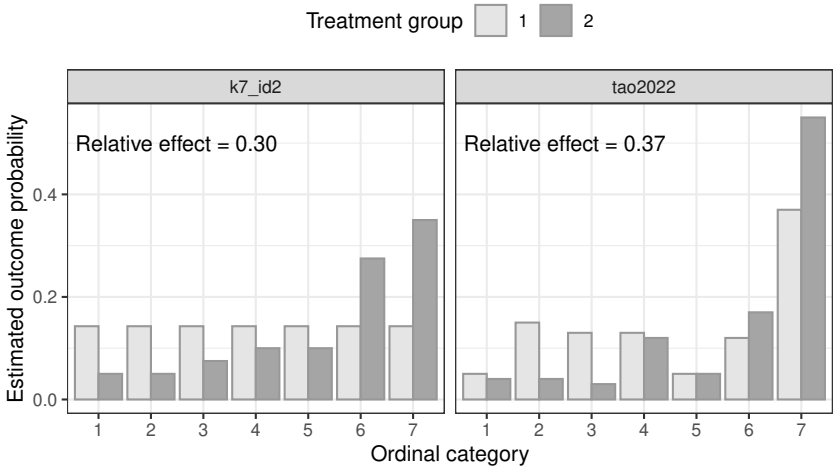

Figure S2: Two of the sets of outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ considered in the example illustration on hypothesis testing in the context of a two-arm randomized controlled trial with an ordinal outcome (see Section 4.1), one researcher-specified (left) and one real-data-based (right), as well as the corresponding relative effect for each set. The shown real-data-based probabilities are the estimates published by Tao et al. (2022).

Method performance The strong alignment between the Wilcoxon rank-sum test and the relative effect deviation from 0.5 arises because the Wilcoxon test statistic is based on rank-based comparisons, which are inherently linked to the relative effect. However, this relationship is not necessarily deterministic in all DGMs beyond those considered in this simulation (e.g., for smaller sample sizes; Thas, 2010). Similarly, the close agreement between the Wilcoxon rank-sum test and PO ordinal logistic regression can be attributed to the fact that the score test for the treatment effect in the PO ordinal logistic regression model is asymptotically equivalent

Table S2: The 15 real-data-based pairs of outcome probabilities $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)$ considered in the example illustration on hypothesis testing in the context of a two-arm randomized controlled trial with an ordinal outcome (see Section 4.1) as well as additional information about the corresponding trials.

Dataset ID	Publication	Treated condition	Measure	n_1	n_2	$\pi_{1,1}$	$\pi_{2,1}$	$\pi_{3,1}$	$\pi_{4,1}$	$\pi_{5,1}$	$\pi_{6,1}$	$\pi_{7,1}$	$\pi_{1,2}$	$\pi_{2,2}$	$\pi_{3,2}$	$\pi_{4,2}$	$\pi_{5,2}$	$\pi_{6,2}$	$\pi_{7,2}$
albers2018	Albers et al. (2018)	Stroke	mRS	92	90	0.10	0.16	0.18	0.15	0.18	0.08	0.14	0.08	0.04	0.04	0.16	0.27	0.16	0.26
campbell2018	Campbell et al. (2018)	Stroke	mRS	101	101	0.28	0.21	0.14	0.14	0.08	0.06	0.10	0.18	0.23	0.09	0.12	0.14	0.07	0.18
cavalcanti2020	Cavalcanti et al. (2020)	COVID-19	Other	159	173	0.64	0.17	0.08	0.04	0.01	0.03	0.03	0.68	0.17	0.05	0.03	0.01	0.04	0.03
goldman2020	Goldman et al. (2020)	COVID-19	Other	200	197	0.08	0.08	0.04	0.10	0.06	0.04	0.60	0.11	0.17	0.05	0.07	0.07	0.02	0.52
hutchinson2020	Hutchinson et al. (2020)	Subdural hematoma	mRS	341	339	0.48	0.14	0.04	0.18	0.03	0.04	0.09	0.48	0.16	0.06	0.19	0.03	0.02	0.05
jovin2022	Jovin et al. (2022)	Stroke	mRS	110	107	0.06	0.18	0.15	0.07	0.09	0.14	0.31	0.01	0.06	0.07	0.10	0.19	0.15	0.42
lecouffe2021	LeCouffe et al. (2021)	Stroke	mRS	273	266	0.04	0.12	0.33	0.10	0.10	0.11	0.21	0.06	0.09	0.36	0.09	0.14	0.09	0.16
ma2019	Ma et al. (2019)	Stroke	mRS	113	112	0.12	0.23	0.14	0.13	0.13	0.12	0.12	0.11	0.19	0.13	0.14	0.21	0.12	0.09
martins2020	Martins et al. (2020)	Stroke	mRS	111	110	0.08	0.12	0.15	0.22	0.13	0.06	0.24	0.03	0.06	0.12	0.15	0.19	0.16	0.30
perkins2018	Perkins et al. (2018)	Cardiac arrest	mRS	4007	3994	0.00	0.00	0.01	0.01	0.00	0.01	0.97	0.00	0.00	0.01	0.01	0.00	0.00	0.98
rosas2021	Rosas et al. (2021)	COVID-19 pneumonia	Other	294	144	0.56	0.02	0.05	0.02	0.09	0.06	0.20	0.49	0.06	0.03	0.07	0.10	0.06	0.19
tao2022	Tao et al. (2022)	Stroke	mRS	226	114	0.05	0.15	0.13	0.13	0.05	0.12	0.37	0.04	0.04	0.03	0.12	0.05	0.17	0.55
thomalla2018	Thomalla et al. (2018)	Stroke	mRS	254	249	0.21	0.32	0.21	0.12	0.07	0.02	0.04	0.15	0.27	0.23	0.17	0.13	0.04	0.01
vandenberg2017	van den Berg et al. (2017)	Stroke	mRS	194	197	0.03	0.05	0.30	0.18	0.06	0.08	0.30	0.01	0.05	0.18	0.17	0.10	0.11	0.39
yang2020	Yang et al. (2020)	Stroke	mRS	326	328	0.13	0.11	0.12	0.19	0.11	0.15	0.18	0.14	0.09	0.14	0.15	0.12	0.18	0.19

Note: Due to rounding to two decimal places, seven non-zero probability values (ranging between 0.0020 and 0.0042) are shown as 0.00.
Abbreviations: COVID-19, Coronavirus disease 2019; mRS, modified Rankin Scale.

to the Wilcoxon rank-sum test under the PO assumption. That is, when the treatment effect results in a constant shift in the log-odds of higher categories, the two tests behave similarly (Agresti, 2010), which appears to hold for the outcome probabilities considered here.

B Example 2: Differential gene expression analysis

B.1 Excluded DGMs

From the DGMs investigated by Baik et al. (2020), we adopted or excluded DGMs as specified below.

- We adopted the DGMs with independent samples within groups; thus, we excluded those with genetically identical replicates within groups.
- We adopted the DGMs where the proportion of DE genes (p_{DE}) is greater than zero; thus, we excluded those with $p_{\text{DE}} = 0$.
- We adopted the DGMs whose results are presented as figures in the main text; thus, we excluded those found only in the supplement.
- We adopted the DGMs representing the default mode (D) with respect to outliers; thus, we excluded those with random outlier counts (R), where 5% of counts are turned into outliers, and those with outlying dispersion samples (OS), where one third of the samples in each group have their dispersions increased fivefold to simulate low-quality samples.

B.2 Dataset selection

Database As a database, we use The Cancer Genome Atlas (TCGA) program (https://www.cancer.gov/tcga), which contains RNA-Seq datasets for 33 different cancer types. The datasets are accessed via the `R` package `curatedTCGAData` (Ramos et al., 2017, 2020).

Dataset-level criteria The following exclusion criteria are applied:

- (D2) Exclude datasets that do not contain samples of both type "01-Primary Solid Tumor" and type "11-Solid Tissue Normal".
- (D3) Exclude datasets with fewer than 10 matched sample pairs across the two groups.

The first criterion excludes 10 datasets, and the second excludes an additional 9 datasets (7 with fewer than 10 samples in total and 2 with fewer than 10 samples in both groups when considering only paired samples). After applying both criteria, 14 datasets remain (see Table S3).

Table S3: Information about the 14 TCGA datasets considered in the example illustration on methods for differential gene expression analysis, including the KIRC dataset used by Baik et al. (2020). Each dataset contains 20,501 genes. More information about the data, cancers, and studies can be found at https://www.cancer.gov/ccg/research/genome-sequencing/tcga/studied-cancers.

Study abbreviation	Study name	n
BLCA	Bladder urothelial carcinoma	38
BRCA	Breast invasive carcinoma	224
COAD	Colon adenocarcinoma	52
ESCA	Esophageal carcinoma	22
HNSC	Head and neck squamous cell carcinoma	86
KICH	Kidney chromophobe	50
KIRC	Kidney renal clear cell carcinoma	144
KIRP	Kidney renal papillary cell carcinoma	64
LIHC	Liver hepatocellular carcinoma	100
LUAD	Lung adenocarcinoma	116
LUSC	Lung squamous cell carcinoma	102
PRAD	Prostate adenocarcinoma	104
STAD	Stomach adenocarcinoma	64
THCA	Thyroid carcinoma	118

B.3 Additional results

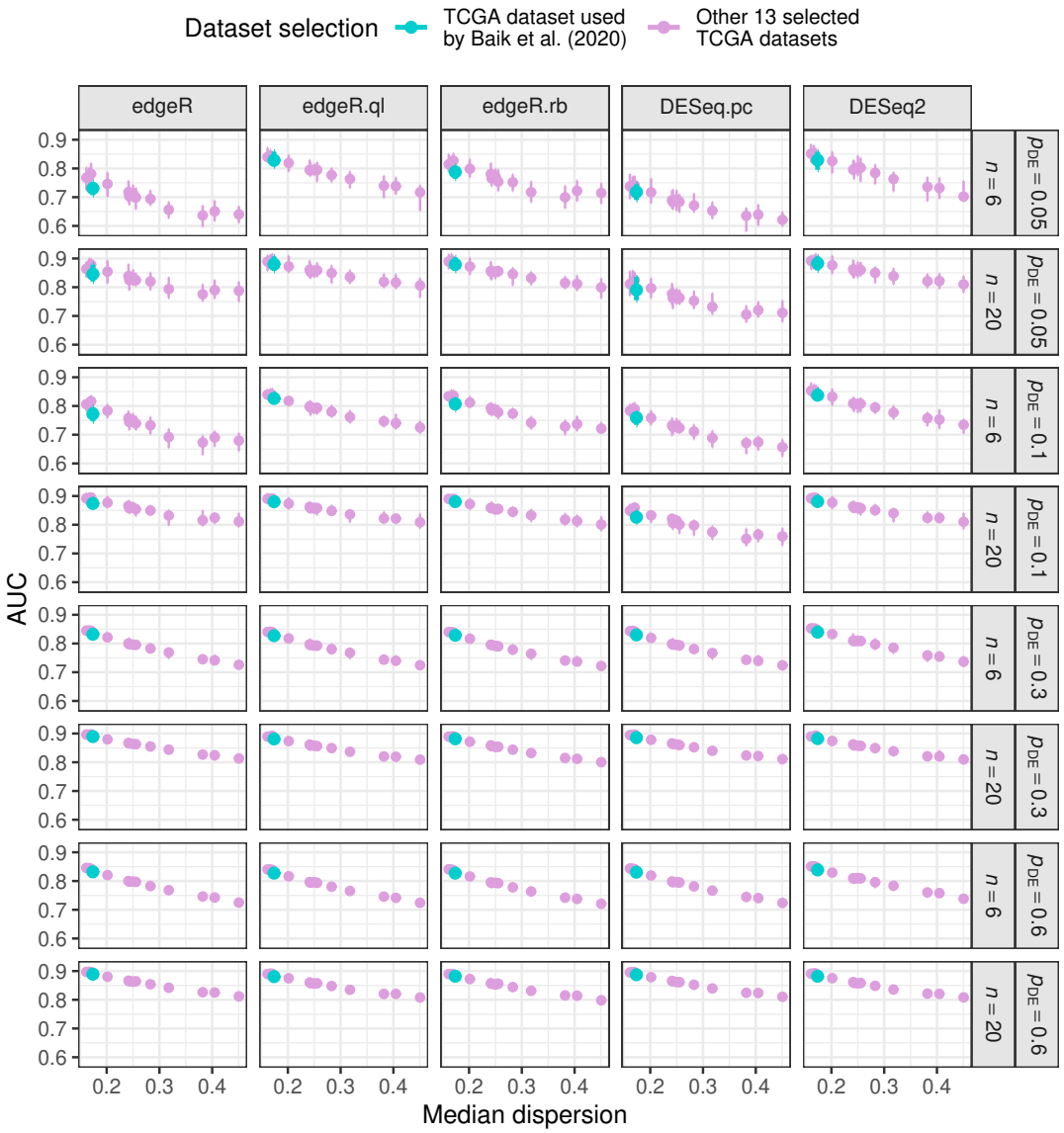

Figure S3: Absolute performance of edgeR, edgeR.ql, edgeR.rb, DESeq.pc, and DESeq2 in relation to the median dispersion (averaged across all genes in the real datasets after filtering), across all considered sample sizes ($n \in \{6, 20\}$) and proportions of DE genes ($p_{\text{DE}} \in \{0.05, 0.1, 0.3, 0.6\}$), comparing results based on the KIRC dataset used by Baik et al. (2020) and the results based on 13 other selected TCGA datasets. Each panel displays the median and range of absolute AUC values.

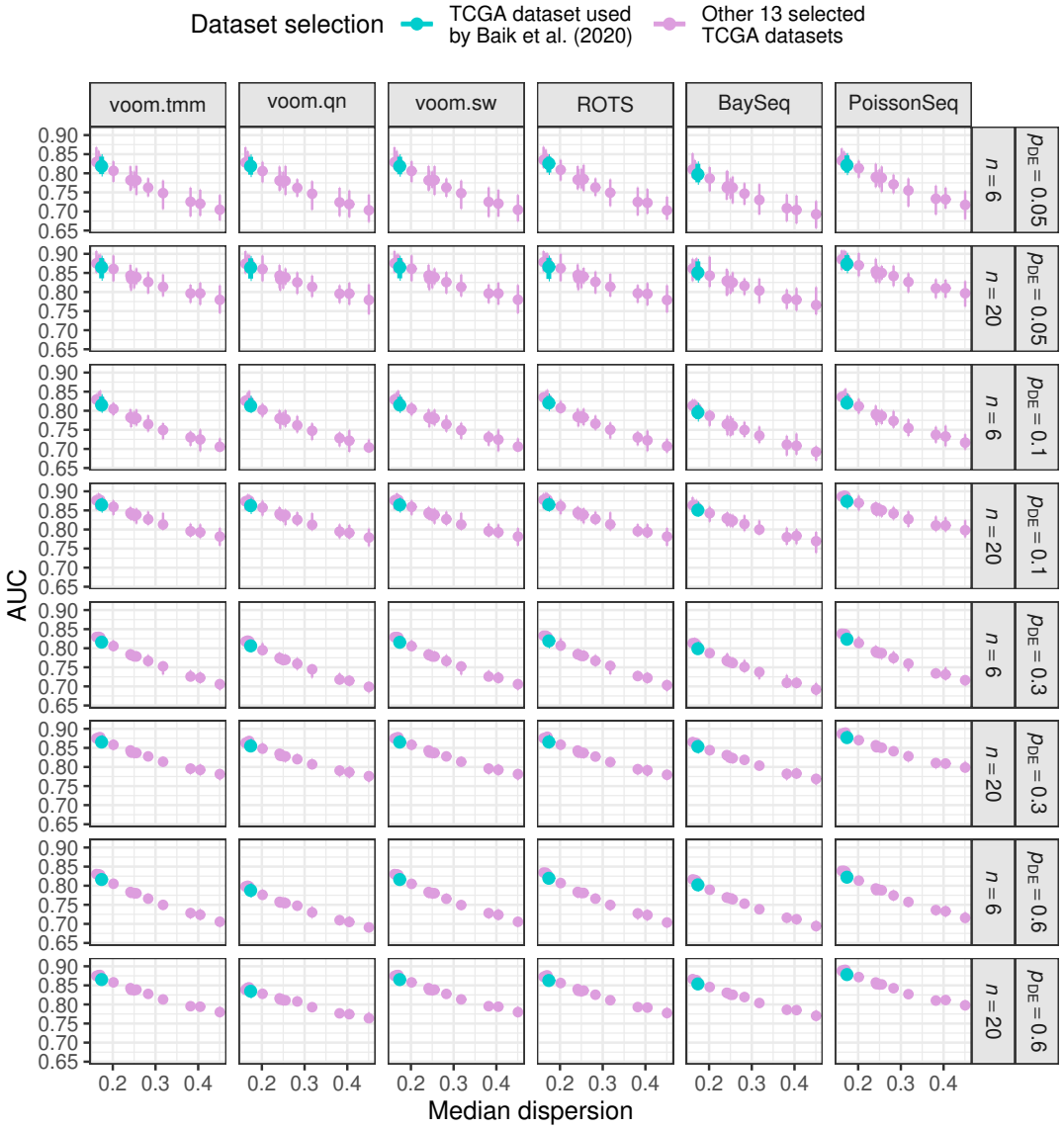

Figure S4: Absolute performance of voom.tmm, voom.qn, voom.sw, ROTS, BaySeq, and PoissonSeq in relation to the median dispersion (averaged across all genes in the real datasets after filtering), across all considered sample sizes ($n \in \{6, 20\}$) and proportions of DE genes ($p_{\text{DE}} \in \{0.05, 0.1, 0.3, 0.6\}$), comparing results based on the KIRC dataset used by Baik et al. (2020) and the results based on 13 other selected TCGA datasets. Each panel displays the median and range of absolute AUC values.

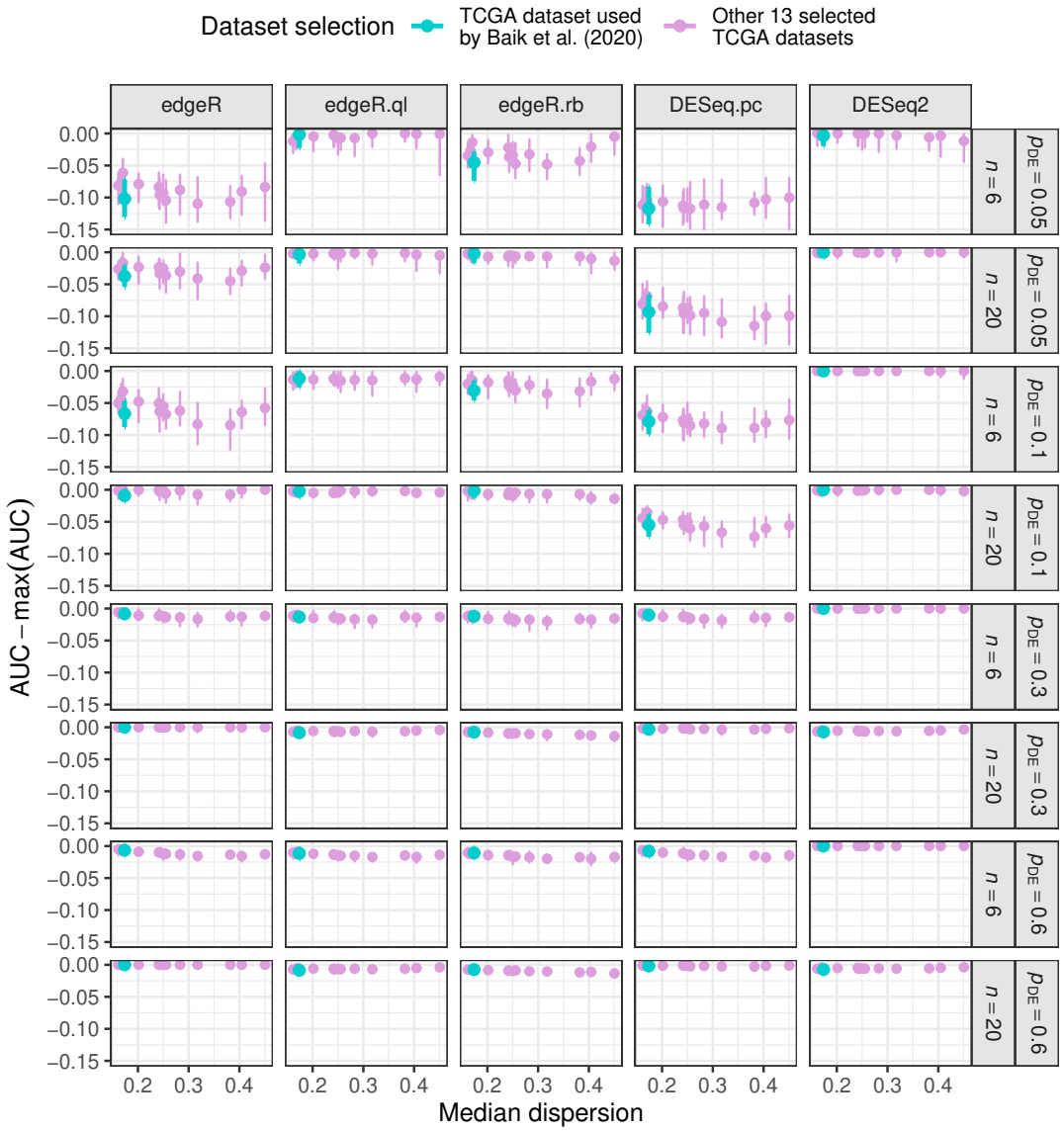

Figure S5: Relative performance of edgeR, edgeR.ql, edgeR.rb, DESeq.pc, and DESeq2 in relation to the median dispersion (averaged across all genes in the real datasets after filtering), across all considered sample sizes ($n \in \{6, 20\}$) and proportions of DE genes ($p_{\text{DE}} \in \{0.05, 0.1, 0.3, 0.6\}$), comparing results based on the KIRC dataset used by Baik et al. (2020) and the results based on 13 other selected TCGA datasets. Each panel displays the median and range of the difference between the AUC and the highest AUC observed within each DGM.

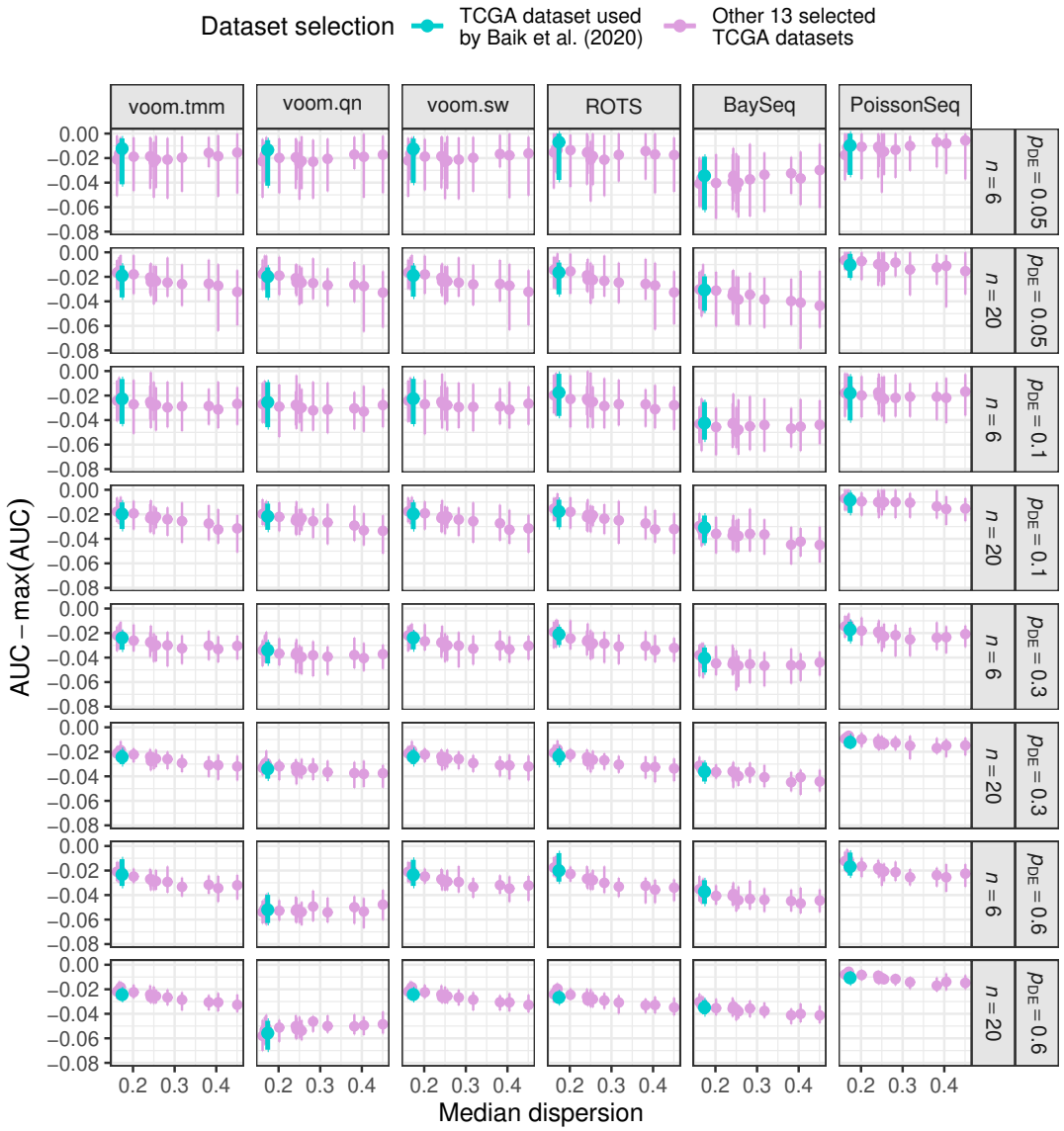

Figure S6: Relative performance of voom.tmm, voom.qn, voom.sw, ROTS, BaySeq, and PoissonSeq in relation to the median dispersion (averaged across all genes in the real datasets after filtering), across all considered sample sizes ($n \in \{6, 20\}$) and proportions of DE genes ($p_{\text{DE}} \in \{0.05, 0.1, 0.3, 0.6\}$), comparing results based on the KIRC dataset used by Baik et al. (2020) and the results based on 13 other selected TCGA datasets. Each panel displays the median and range of the difference between the AUC and the highest AUC observed within each DGM.

Nutzung von großen Sprachmodellen

Zur Anfertigung dieser Dissertation wurden große Sprachmodelle (Large Language Models) genutzt. Diese wurden ausschließlich herangezogen, um Vorschläge für sprachliche Korrekturen auf Basis bereits verfasster Inhalte zu generieren. Hierbei wurden die Modelle GPT-4o (OpenAI) und GPT-5 (OpenAI) verwendet.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 16.01.2026 _____

Christina Sauer