RESEARCH ARTICLE

AteMeVs: An R package for the estimation of the average treatment effect with measurement error and variable selection for confounders

Li-Pang Chen[1], **Grace Y. Yi**[2]*

1 Department of Statistics, National Chengchi University, Taipei, Taiwan, ROC, **2** Department of Statistical and Actuarial Sciences, Department of Computer Science, University of Western Ontario, London, Canada

* gyi5@uwo.ca

OPEN ACCESS

Citation: Chen L-P, Yi GY (2024) AteMeVs: An R package for the estimation of the average treatment effect with measurement error and variable selection for confounders. PLoS ONE 19(9): e0296951. https://doi.org/10.1371/journal.pone.0296951

Editor: Suyan Tian, The First Hospital of Jilin University, CHINA

Received: October 16, 2023

Accepted: December 20, 2023

Published: September 27, 2024

Data Availability Statement: The full dataset is available at the Kaggle website: https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric.

Funding: L.-P. Chen: National Science and Technology Council G. Y. Yi: Natural Sciences and Engineering Research Council of Canada; Canada Research Chairs The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

In causal inference, the estimation of the average treatment effect is often of interest. For example, in cancer research, an interesting question is to assess the effects of the chemotherapy treatment on cancer, with the information of gene expressions taken into account. Two crucial challenges in this analysis involve addressing measurement error in gene expressions and handling noninformative gene expressions. While analytical methods have been developed to address those challenges, no user-friendly computational software packages seem to be available to implement those methods. To close this gap, we develop an R package, called **AteMeVs**, to estimate the average treatment effect using the inverse-probability-weighting estimation method to handle data with both measurement error and spurious variables. This developed package accommodates the method proposed by Yi and Chen (2023) as a special case, and further extends its application to a broader scope. The usage of the developed R package is illustrated by applying it to analyze a cancer dataset with information of gene expressions.

1 Introduction

Bioinformatics has revealed that cancer stems from a genetic disorder, deiven by genetic variations that lead to the abnormally dysfunction of genes and their altered expressions (e.g., [1]). Accurate assessment of gene expression levels becomes crucial for cancer diagnosis and treatment. Chemotherapy is a commonly used approach in cancer treatment as it often effectively eradicates malignant cells. In particular, the integration of targeted therapy with chemotherapy is frequently used to control the growth, division, and spread of cancer cells (e.g., [2]). However, due to its lack of specificity in targeting cancer cells, chemotherapy drugs can impact both cancer cells and healthy cells, which leads to significant side effects. One concern is whether employing chemotherapy is more beneficial than taking alternative treatments that avoid it. We are interested in studying whether taking chemotherapy has a causal effect on increasing the survival of cancer patients.

Competing interests: The authors have declared that no competing interests exist.

This research is partially motivated by the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database, a Canada-UK Project that includes targeted sequencing data of primary breast cancer samples collected by the Cambridge Research Institute and the British Columbia Cancer Centre in Canada [3]. The dataset with all gene expression names is publicly available on the Kaggle website (https://www.kaggle.com/datasets/raghadalharbi/breast-cancer-gene-expression-profiles-metabric). One interesting question is whether patients taking chemotherapy as a treatment ("hormone_therapy": 1 is yes and 0 is no) can increase the chance of the survival status ("overall_survival": 1 is alive and 0 is dead). The dataset contains Z-scores of m-RNA levels for 331 genes, where the Z-score is defined as

$$\frac{\text{an expression level in the tumor sample} - \text{the mean expression in reference sample}}{\text{standard deviation of expression levels in reference sample}},$$

which is a continuous variable. In addition, some gene expressions may be confounded with the outcome and treatment, shown in Fig 1.

Taking the causal inference paradigm, we formulate the question as the estimation of the average treatment effect (ATE), defined as the difference between potential outcomes under two treatments, where the two treatments refer to taking and not taking chemotheropy, respectively.

Various causal inference methods, accompanied by R packages, are available in the literature. Examples include **iWeigReg** [4], **SVMMatch** [5], **CausalGAM** [6], and **wfe** [7]. The R package **mediation** [8] is developed to conduct mediation analysis. Package **qualCI** [9] is used to analyze causal inference with qualitative and ordinal information on outcomes. **Matching-Frontier** [10] applies the matching method to handle the balance issue in causal inference.

In the framework of causal inference, the inverse probability weighted (IPW) estimation method has been widely used to estimate average treatment effects due to its simplicity and transparent interpretation (e.g., [11–13]). The method adjusts for the effects of measured confounders by re-weighting the data as if the weighted data were collected from randomized

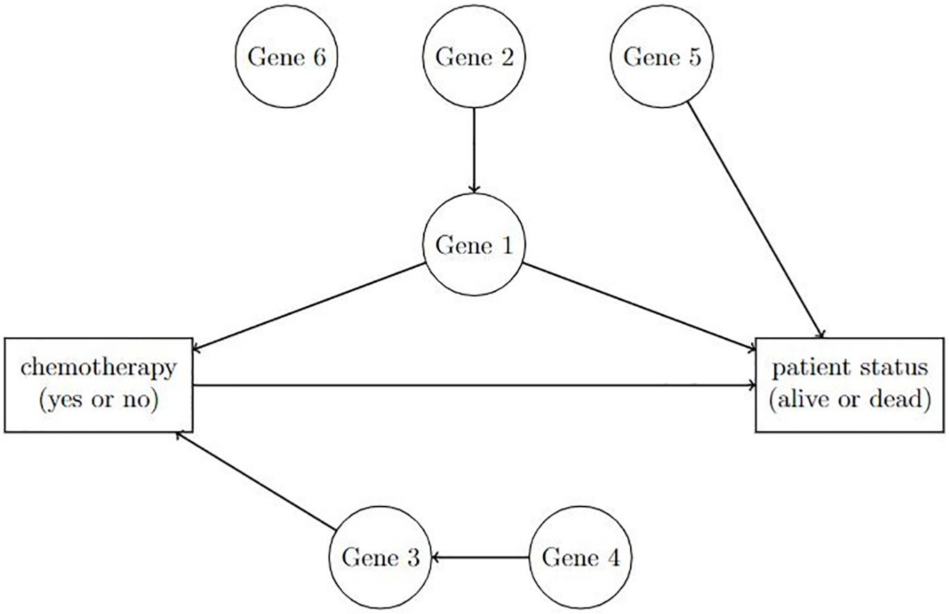

Fig 1. An illustrative diagram of the causal relationship with possible confounders.

https://doi.org/10.1371/journal.pone.0296951.g001 controlled trials. The validity of the method requires two key conditions: (1) the treatment model is correctly specified to consistently estimate propensity scores, and (2) the variables in the treatment model are precisely measured.

In the presence of measurement error, directly applying those methods to the observed data usually yields unreliable estimation results. As illustrated in Fig 1, in the study of the treatment effect on the outcome, confounders (e.g., gene expressions) may possess complex features: some are associated only with the treatment variable, some are solely related to the outcome variable, some are connected to both treatment and outcome variables, and some are not relevant to either treatment or outcome variable. Including irrelevant confounders in the analysis or model building may lead to misleading results.

In the literature, variable selection for causal inference (e.g., [14–16]) or measurement error correction (e.g., [17–22]) are discussed under various settings. However, in the concurrent presence of both features, limited work has been carried out to estimate ATE except for [23]. Moreover, there is a lack of user-friendly R packages designed to facilitate causal estimation for data with both measurement error and spurious variables.

In this paper, we develop an R package, called **AteMeVs**, which is desired to estimate the average treatment effect with measurement error and variable selection for confounders. This package, available at CRAN [24], is developed to implement a recent method proposed by [23], which estimates the average treatment effect for noisy data that include both measurement error and spurious variables needed to be excluded. The developed package contains a set of functions that provide a step-by-step estimation procedure, including the correction of the measurement error effects, variable selection for the estimation of propensity scores, and estimation of ATE. Our functions contain multiple options for users to implement, including different ways to correct for the measurement error effects, various penalty functions for variable selection, and different regression models for characterizing propensity scores.

2 Notation and framework

2.1 Propensity score

In contrast to the variables indicated by Fig 1, we now introduce abstract symbols to classify the associated variables differently. Let T denote the observed binary treatment (e.g., chemotherapy) with $T = 1$ if treated and $T = 0$ if untreated. Accordingly, we consider counterfactual responses corresponding to the treatment status. For $t \in \{0, 1\}$, let $Y_{(t)}$ represent the potential outcome of the patient if the patient would have received $T = t$. As described in Section 1, the goal is to assess the causality effect of the treatment (e.g., chemotherapy) on increasing the patient's survival, or equivalently, we are interested in estimating the ATE,

$$\tau_0 \triangleq E(Y_{(1)}) - E(Y_{(0)}).$$

To facilitate an individual's characteristics, we let W denote the p-dimensional vector of pre-treatment confounders for the individual, which as an example, can be understood as gene expressions in Fig 1. To reflect the possible dependence of T on W, we consider the conditional probability

$$\pi \triangleq P(T = 1|W),$$

also called the propensity score for the individual. The introduction of the propensity score allows us to use the observed outcome, denoted by Y, and the treatment information to consistently estimate τ_0 (e.g., [25]), which basically is due to the property (e.g., [23])

$$\tau_0 = E\left(\frac{TY}{\pi}\right) - E\left\{\frac{(1-T)Y}{1-\pi}\right\}. \tag{1}$$

The validity of (1) hinges on the following standard assumptions in the causal inference framework:

1. The strong ignorable treatment assumption (SITA): given the covariates W, potential outcomes $Y_{(0)}$ and $Y_{(1)}$ are independent of T;
2. The stable unit treatment value assumption (SUTVA), also known as the consistency assumption: each subject's potential outcomes are not influenced by the actual treatment assignment of other subjects. Therefore, the observed outcome, Y, for an individual is assumed to be linked with potential outcomes via $Y = TY_{(1)} + (1 - T)Y_{(0)}$;
3. The positivity assumption: the propensity score is between 0 and 1, i.e., $0 < \pi(W) < 1$ for all W.

In applications, π is frequently characterized by a parametric model:

$$g^{-1}(\pi) = W^{\top}\gamma, \tag{2}$$

where $\gamma = (\gamma_0, \gamma_1, \cdots, \gamma_p)^{\top}$ is the vector of regression parameters of dimension $p + 1$, with γ_0 representing the intercept; and $g(\cdot)$ is a link function. For ease of exposition and the inclusion of the intercept, we slightly abuse the notation W in (2) by including 1 to the original p-dimensional vector of confounders here and in the subsequent development. Common choices of $g(\cdot)$ include the logit, probit, and complementary log-log functions, respectively yielding

- *the logistic regression model*:

$$\pi = \frac{\exp(W^{\top}\gamma)}{1 + \exp(W^{\top}\gamma)}, \tag{3}$$

- *the probit regression model*:

$$\pi = \Phi(W^{\top}\gamma), \tag{4}$$

  with $\Phi(\cdot)$ representing the cumulative distribution function of the standard normal distribution, and

- *the complementary log-log regression model*:

$$\pi = 1 - \exp\{-\exp(W^{\top}\gamma)\}. \tag{5}$$

2.2 The IPW estimator

With the setup in Section 2.1, the estimation of τ_0 can be carried out using the measurements of a random sample, say $\{\{T_i, Y_i, W_i\}: i = 1, \ldots, n\}$ of size n, where T_i, Y_i, and W_i represent the corresponding variables for subject i with $i = 1, \ldots, n$.

The estimation of τ_0 basically involves the following two steps. In the first step, we estimate the propensity score $\pi_i = P(T_i = 1|W_i)$ for subject $i = 1, \ldots, n$ based on estimating parameter γ

in model (2). Let $S_i(\gamma; W_i)$ denote the likelihood score function obtained from subject i that is derived from fitting model (2). If the true value of W_i is available, one may solve

$$\sum_{i=1}^{n} S_i(\gamma; W_i) = 0 \tag{6}$$

for γ to obtain a consistent estimate of γ, denoted $\widehat{\gamma}$, provided usual regularity conditions. Then we calculate π_i with γ in (2) replaced by the estimate $\widehat{\gamma}$, and let $\widehat{\pi}_i$ denote the resulting value of π_i.

In the second step, utilizing the property (1) yields a consistent estimate of the ATE τ_0 by the following IPW estimator, as initiated by [25]:

$$\widehat{\tau} = \frac{1}{n}\sum_{i=1}^{n}\frac{T_iY_i}{\widehat{\pi}_i} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-T_i)Y_i}{1-\widehat{\pi}_i}. \tag{7}$$

To mitigate unstable numerical results caused by extreme values of $\widehat{\pi}_i$ that may be close to 0 or 1, [12] proposed a stable version of (7), which also offers a consistent estimator of τ_0:

$$\widehat{\tau} = \left(\sum_{i=1}^{n}\frac{T_i}{\widehat{\pi}_i}\right)^{-1}\sum_{i=1}^{n}\frac{T_iY_i}{\widehat{\pi}_i} - \left(\sum_{i=1}^{n}\frac{1-T_i}{1-\widehat{\pi}_i}\right)^{-1}\sum_{i=1}^{n}\frac{(1-T_i)Y_i}{1-\widehat{\pi}_i}. \tag{8}$$

We use (8) for the following development.

2.3 Irrelevant variables and measurement error

As noted in [23], the validity of (7) or (8) breaks down in the presence of two features of noisy data: measurement error and irrelevant variables.

In applications, some variables (e.g., gene expressions) in W_i can be subject to measurement error. To reflect this feature, we write W_i as $(X_i^\top, Z_i^\top)^\top$ so that all error-prone variables are included in X_i and all precisely measured variables go to Z_i. Let X_i^* denote the observed surrogate measurement of X_i.

To characterize the relationship between X_i^* and X_i, we consider the classical additive error model (e.g., [26, 27])

$$X_i^* = X_i + e_i, \tag{9}$$

where the error term e_i is independent of $\{T_i, X_i, Z_i, Y_i\}$ and follows $N(0, \Sigma_e)$ with covariance matrix Σ_e.

Model (9) is the most commonly used in the literature; it facilitates situations where the observed value fluctuates around the true value with an error term, and the degree of measurement error in X_i^* is reflected by the value of Σ_e. In this paper, we consider the following four cases for Σ_e:

Case 1: Σ_e is known;

Case 2: Σ_e is unknown and estimated from repeated surrogate measurements $\{X_{ij}^* : j = 1, \cdots, n_i; i \in \mathcal{R}\}$ for a subset, say $\mathcal{R}$, of $\{1, 2, \cdots, n\}$ with $|\mathcal{R}| = m$ and $m < n$;

Case 3: Σ_e is unknown and estimated from repeated surrogate measurements $\{X_{ij}^* : j = 1, \ldots, n_i\}$ of X_i for $i = 1, \cdots, n$, where $n_i \geq 2$ that may or may not depend on i;

Case 4: Σ_e is unknown and estimated from an external validation sample $\{\{X_k, X_k^*\} : k \in \mathcal{V}\}$, where $\mathcal{V}$ is index set for the subjects in the validation sample.

Case 1 is useful for addressing data issues in which the existence of measurement error is acknowledged, yet the magnitude of this error has not been quantified. In this case, we often conduct sensitivity analyses to assess the sensitivity of inference results to varying degrees of measurement error, where user-specified values for Σ_e are typically used to describe different scenarios of measurement error in X_i (e.g., [27, 28]). This case was also considered by [23]. Cases 2 and 3 complement each other in addressing two scenarios with repeated surrogate measurements. In contrast to the availability of replicates, Case 4 assumes the availability of an external validation dataset.

The second issue of noisy data concerns irrelevant variables in the data. As shown in Fig 1, some gene expressions have no connection with chemotherapy or patient's survival status. To reflect this feature, we write $W_i = (W_{\mathrm{I}i}^\top, W_{\mathrm{II}i}^\top)^\top$ for $i = 1, \cdots, n$, where $W_{\mathrm{I}i}$ includes the informative confounders associated with T_i and Y_i, and $W_{\mathrm{II}i}$ contains noninformative confounders. We further write $W_{\mathrm{I}i} = (Z_{\mathrm{I}i}^\top, X_{\mathrm{I}i}^\top)^\top$ and $W_{\mathrm{II}i} = (Z_{\mathrm{II}i}^\top, X_{\mathrm{II}i}^\top)^\top$ so that $Z_i \triangleq (Z_{\mathrm{I}i}^\top, Z_{\mathrm{II}i}^\top)^\top$ represents the subvector of error-free confounders in W_i and $X_i \triangleq (X_{\mathrm{I}i}^\top, X_{\mathrm{II}i}^\top)^\top$ is the subvector of error-prone confounders in W_i. Let p_{Z} and p_{X} denote the dimension of Z_i and X_i, respectively. We let $X_i^* = (X_{\mathrm{I}i}^{*\top}, X_{\mathrm{II}i}^{*\top})^\top$ denote the observed version of X_i, where $X_{\mathrm{I}i}^*$ and $X_{\mathrm{II}i}^*$ are the observed measurements of $X_{\mathrm{I}i}$ and $X_{\mathrm{II}i}$, respectively.

3 Estimation methods

Here we describe methods for estimating the average treatment effect τ_0, with the features of variable selection and measurement error accommodated for each of the four cases described in Section 2.3. The main idea comes from Section 3.1 of [23], which is developed under Case 1 described in Section 2.3. Sections 3.2-3.4 extend the development in Section 3.1 to respectively handle Cases 3.2-3.4 described in Section 2.3.

3.1 Implementation steps for Case 1

First, we describe the algorithm for Case 1 where Σ_e is user-specified. The algorithm of estimating τ_0 contains the five steps, summarized as follows. For details, see Section 3.1 of [23].

Step 1. Simulation:

We simulate a sequence of artificial surrogates, denoted $\{X_i^*(k, \psi) : k = 1, \cdots, K; \psi \in \mathcal{C}; i = 1, \cdots, n\}$, where K is a user-specified positive integer, $\mathcal{C} = \{\psi_1, \psi_2, \ldots, \psi_{\mathrm{M}}\}$ is a sequence of M non-negative values taken from $[0, \psi_{\mathrm{M}}]$ with a given ψ_{M} and $\psi_1 = 0$, and

$$X_i^*(k, \psi) = X_i^* + \sqrt{\psi} e_{ik} \text{ with } e_{ik} \text{ independently generate from } N(0, \Sigma_e). \tag{10}$$

Step 2. Estimation of Treatment Model Parameters:

Parameter γ in model (2) is estimated by solving (6) with X_i replaced by $X_i^*(k, \psi)$, and let $\widehat{\gamma}(k, \psi)$ denote the resulting estimate. Calculate $\widehat{\gamma}(\psi) = K^{-1} \sum_{k=1}^{K} \widehat{\gamma}(k, \psi)$ for $\psi \in \mathcal{C}$.

Step 3. Extrapolation:

For $j = 0, 1, 2, \ldots, p$, let $\widehat{\gamma}_j(\psi)$ denote the jth element of $\widehat{\gamma}(\psi)$; fit a regression model to $\{(\psi, \widehat{\gamma}_j(\psi)) : \psi \in \mathcal{C}\}$ and extrapolate it to $\psi = -1$; and let $\tilde{\gamma}_j$ denote the resulting extrapolated value of γ_j, the jth element of γ. Write $\tilde{\gamma} = (\tilde{\gamma}_0, \tilde{\gamma}_1, \ldots, \tilde{\gamma}_p)^\top$.

Step 4. Variable Selection:

Minimize the penalized quadratic loss function

$$\begin{aligned}\ell_{\mathrm{P}}(\gamma) &\triangleq \ell(\gamma) - n\sum_{j=1}^{p}\rho_{\lambda}(|\gamma_j|)\\ &\triangleq \frac{1}{2}(\gamma-\tilde{\gamma})^{\top}V_n(\gamma-\tilde{\gamma}) - n\sum_{j=1}^{p}\rho_{\lambda}(|\gamma_j|)\end{aligned} \tag{11}$$

with respect to γ, where $\rho_{\lambda}(\cdot)$ is a user-specified penalty function with a tuning parameter λ, and V_n is a user-specified positive definite weight matrix.

Step 5. Estimation of ATE:

Write $\widehat{\gamma} = (\widehat{\gamma}_{\mathrm{I}}^{\top}, \widehat{\gamma}_{\mathrm{II}}^{\top})^{\top}$ with $\widehat{\gamma}_{\mathrm{I}} = (\widehat{\gamma}_{x\mathrm{I}}^{\top}, \widehat{\gamma}_{z\mathrm{I}}^{\top})^{\top}$ and $\widehat{\gamma}_{\mathrm{II}} = (\widehat{\gamma}_{x\mathrm{II}}^{\top}, \widehat{\gamma}_{z\mathrm{II}}^{\top})^{\top}$ corresponding to the non-zero and zero components in $\widehat{\gamma}$, respectively. With unimportant variables $X_{\mathrm{II}i}$ and $Z_{\mathrm{II}i}$ excluded from the initial model (2), the final treatment model is taken as

$$g^{-1}(\pi_i) = W_{\mathrm{I}i}^{\top}\gamma_{\mathrm{I}}, \tag{12}$$

where γ_{I} is the vector of model parameters associated with important covariates $W_{\mathrm{I}i}$.

For $k = 1, \cdots, K$ and $\psi \in \mathcal{C}$, calculate an estimate, say, $\widehat{\tau}(k,\psi)$, of τ_0 using (8) with $\widehat{\pi}_i$ replaced by the propensity score for subject i, determined by the selected treatment model (12) with $X_{\mathrm{I}i}$ replaced by $X_{\mathrm{I}i}^{*}(k,\psi)$, the subvector of $X_i^{*}(k,\psi)$ that corresponds to $X_{\mathrm{I}i}^{*}$. Then calculate

$$\widehat{\tau}(\psi) = K^{-1}\sum_{k=1}^{K}\widehat{\tau}(k,\psi).$$

Finally, fit a regression model to $\{(\psi, \widehat{\tau}(\psi)) : \psi \in \mathcal{C}\}$ and extrapolate it to $\psi = -1$. The resulting value, denoted as $\widehat{\tau}$, is taken an estimate of τ_0.

3.2 Implementation steps for Case 2

Consider Case 2 where repeated measurements $\{X_{ij}^{*} : j = 1, \cdots, n_i; i \in \mathcal{R}\}$ of X_i are available for $|\mathcal{R}| \triangleq m$ subjects, and surrogates X_{ij}^{*} and X_i are linked via the measurement error model

$$X_{ij}^{*} = X_i + e_{ij} \quad \text{for} \quad i \in \mathcal{R} \quad \text{and} \quad j = 1, \cdots, n_i,$$

where for $i \in \mathcal{R}$, $n_i \geq 2$; e_{ij} follows $N(0, \Sigma_e)$ with unknown covariance matrix Σ_e; and the e_{ij} are independent of $\{T_i, X_i, Z_i, Y_{i(1)}, Y_{i(0)}\}$.

With the repeated measurements, using the method of moments, we estimate Σ_e by

$$\widehat{\Sigma}_e = \frac{\sum_{i\in\mathcal{R}}\sum_{j=1}^{n_i}(X_{ij}^{*} - \overline{X}_i^{*})(X_{ij}^{*} - \overline{X}_i^{*})^{\top}}{\sum_{i\in\mathcal{R}}(n_i - 1)}, \tag{13}$$

where $\overline{X}_i^{*} = n^{-1}\sum_{j=1}^{n_i} X_{ij}^{*}$.

To estimate τ_0, we repeat the five steps described in Section 3.1 with Σ_e in (10) replaced by (13).

3.3 Implementation steps for Case 3

Now we consider Case 3 described in Section 2.3, where Σ_e is unknown but repeated surrogate measurements $\{X_{ij}^* : j = 1, \cdots, n_i\}$ are available for all the subjects in the sample, with $n_i \geq 2$ for $i = 1, \cdots, n$.

Adapting the development in [27, 29] (p.107), we modify Step 1 in Section 3.1 as follows. For any $\psi \in \mathcal{C}$ and $i = 1, \cdots, n$, we generate n_i variates independently from the standard normal distribution, and let $\{d_{ij}(\psi): j = 1, \cdots, n_i\}$ denote them. Calculate $\overline{d}_i(\psi) = \frac{1}{n_i}\sum_{j=1}^{n_i} d_{ij}(\psi)$ and

$$c_{ij}(\psi) = \frac{d_{ij}(\psi) - \overline{d}_i(\psi)}{\sqrt{\sum_{l=1}^{n_i}\{d_{il}(\psi) - \overline{d}_i(\psi)\}^2}}.$$

Then for $k = 1, \cdots, K$, we define

$$X_i^{**}(k, \psi) = \overline{X}_i^* + \sqrt{\frac{\psi}{n_i}}\sum_{j=1}^{n_i} c_{ij}(\psi)X_{ij}^*, \tag{14}$$

where $\overline{X}_i^* = n_i^{-1}\sum_{j=1}^{n_i} X_{ij}^*$. Estimation of τ_0 can then be proceeded following the five steps in Section 3.1, with $X_i^*(k, \psi)$ in (10) replaced by $X_i^{**}(k, \psi)$ in (14).

3.4 Implementation steps for Case 4

We now consider Case 4 described in Section 2.3. In this case, we have the main study data, given by $\left\{\{Y_i, T_i, Z_i, X_i^*\} : i \in \mathcal{M}\right\}$ with $\mathcal{M} = \{1, \cdots, n\}$, and an external validation sample $\{\{X_k, X_k^*\} : k \in \mathcal{V}\}$ with size $|\mathcal{V}| \triangleq m$, where the index sets $\mathcal{M}$ and $\mathcal{V}$ do not overlap. We assume that X_k^* and X_k are related via (9) for $k \in \mathcal{V}$. Further, we make the transportability assumption, considered in [30].

With the availability of X_k and X_k^* in $\mathcal{V}$, we can empirically estimate $\Sigma_X = \text{var}(X_k)$ and $\Sigma_{X^*} = \text{var}(X_k^*)$, and denote the resulting estimators by

$$\widehat{\Sigma}_X = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}(X_i - \overline{X}_i)(X_i - \overline{X}_i)^\top$$

and

$$\widehat{\Sigma}_{X^*} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\left(X_i^* - \overline{X}_i^*\right)\left(X_i^* - \overline{X}_i^*\right)^\top,$$

respectively, where $\overline{X}_i = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} X_i$ and $\overline{X}_i^* = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} X_i^*$.

Consequently, by the additivity of covariance matrices in (9), we estimate Σ_e by

$$\widehat{\Sigma}_e = \widehat{\Sigma}_{X^*} - \widehat{\Sigma}_X. \tag{15}$$

Then estimation of τ_0 is carried out following the five steps in Section 3.1, with Σ_e in (10) replaced by the estimator $\widehat{\Sigma}_e$ in (15).

4 Implementation details

To implement the estimation procedures described in Section 3, we need to first decide the choice of relevant quantities, including K and $\mathcal{C}$ in Step 1, the regression model for extrapolation in Steps 3 and 5, and the penalty function together with the tuning parameter in Step 4. In the following subsections, we discuss the choice of each quantity individually.

4.1 Choice of K and $\mathcal{C}$ in step 1

Integer K determines the repetition of simulated data $X_i^*(k, \psi)$ for each given ψ. To reduce Monte Carlo errors, a larger value of K is expected to produce a more stable result of $\widehat{\gamma}(\psi)$. On the other hand, a larger value of K requires a substantially longer computational time. Therefore, a suitable choice of K is driven by the trade-off between the computation time and the accuracy of the results. Empirical experience (e.g., [23, 31–33]) suggests setting K to be 50, 100, 200, or 500 may be reasonable for many applications.

Regarding the choice of $\mathcal{C}$, one may take ψ_{M} to be 1 or 2, and divide the interval $[0, \psi_{\mathrm{M}}]$ equally into M sub-intervals, where M may be taken as 5, 10, or other positive integers. Then $\mathcal{C}$ is the set of the resulting cut points.

4.2 Choice of extrapolation function in steps 3 and 5

[26] (Section 5.3.2) suggests to use one of the following functions, denoted by $\varphi(u)$ with parameters β_0, β_1, β_2, and β_3, to approximate the true extrapolation functions in implementing Steps 3 and 5:

1. the quadratic function

$$\varphi(u) = \beta_0 + \beta_1 u + \beta_2 u^2; \tag{16}$$

2. the linear function

$$\varphi(u) = \beta_0 + \beta_1 u; \tag{17}$$

3. the rational linear function

$$\varphi(u) = \beta_0 + \frac{\beta_1}{\beta_2 + u}. \tag{18}$$

To increase flexibility, we add the following function to approximate the extrapolation functions for the implementation of Steps 3 and 5:

4. the cubic function

$$\varphi(u) = \beta_0 + \beta_1 u + \beta_2 u^2 + \beta_3 u^3. \tag{19}$$

4.3 Choices of penalty function in step 4

In implementing Step 4 in Section 3.1, we consider the following commonly used penalty functions $\rho_\lambda(u)$ that are included in the R package `ncvreg`:

1. the least absolute shrinkage and selection operator (LASSO) penalty [34]:
$$\rho_\lambda(u) = \lambda|u|, \tag{20}$$

2. the smoothly clipped absolute deviation (SCAD) penalty [35]:
$$\rho'_\lambda(u) = \lambda\{I(u \leq \lambda) + \frac{(a\lambda - u)_+}{(a-1)\lambda} \cdot I(u > \lambda)\}, \tag{21}$$
where $I(\cdot)$ is the indicator function, $u_+ = \max\{u, 0\}$, and $a = 3.7$. Here $\rho'_\lambda(u)$ is the first order derivative of the penalty function $\rho_\lambda(u)$ with tuning parameter λ.

3. the minimax concave penalty (MCP) function proposed by [36]:
$$\rho'_\lambda(u) = (\lambda - u/a)_+ \tag{22}$$
with $a = 3$.

4. the Elastic Net [37]:
$$\rho_\lambda(u) = \lambda\{(1-\alpha)u^2 + \alpha|u|\}, \tag{23}$$
with $\alpha \in [0, 1]$. If $\alpha = 1$, then (23) reduces to (20); when $\alpha = 0$, then (23) gives the L_2-norma penalty for the ridge regression.

4.4 Determination of tuning parameter

To achieve satisfactory performance of the selection procedure, we may consider one of the following criteria for choosing a suitable value for the tuning parameter λ:

1. Bayesian Information Criterion (BIC)
Given a grid Λ of possible values for the tuning parameter λ, and for $\lambda \in \Lambda$, let
$$\widehat{\gamma}(\lambda) = \text{argmin}_\gamma \ell_P(\gamma)$$
and let df_λ denote the number of non-zero elements of $\widehat{\gamma}(\lambda)$. We define
$$BIC(\lambda) = -2\ell(\widehat{\gamma}(\lambda)) + 2(\log n)df_\lambda.$$
Then the optimal tuning parameter λ^* is chosen as the minimizer of $BIC(\lambda)$:
$$\lambda^* = \text{argmin}_{\lambda\in\Lambda} BIC(\lambda).$$

2. V-fold cross validation (CV)
The original dataset is first divided into V subsamples with an equal size, where V is a user-specified positive integer, such as $V = 5$. The rth subsample is taken as the testing set and the remaining $(V - 1)$ subsamples are merged as the training set.
Applying (11) to the training set gives us an estimator of γ, denoted $\gamma^{(-r)}(\lambda)$. Then we evaluate $\ell(\gamma)$ in (11) at $\gamma = \gamma^{(-r)}(\lambda)$ based on the rth testing set, and let $\ell^{(r)}(\gamma^{(-r)}(\lambda))$ denote the resulting value. Finally, we compute
$$CV(\lambda) = \frac{1}{V}\sum_{r=1}^{V} \ell^{(r)}(\gamma^{(-r)}(\lambda)).$$

The optimal tuning parameter λ^* is then determined by the minimizer of $CV(\lambda)$:

$$\lambda^* = \mathrm{argmin}_{\lambda \in \Lambda} CV(\lambda).$$

This approach can be realized by employing the function `cv.ncvreg` in the R package `ncvreg`.

5 Syntax of R package

In this section, we present our developed R package, **AteMeVs**, which implements the estimation procedures described in Section 3, together with the details in Section 4. The developed R package utilizes the available R packages: **MASS** and **ncvreg**. The former is used to generate a multivariate normal distribution to address (10), and the latter is used to implement penalty functions as outlined in Section 4. Below, we describe the syntax of the developed functions that implement the step-by-step estimation procedure in Section 3.

SIMEX_EST

The function `SIMEX_EST` implements Steps 1-3 in Section 3.1, given by

```
  SIMEX_EST(data, PS = "logistic", Psi = seq(0,1,length = 10),
px = p, K = 200, extrapolate="quadratic", Sigma_e, replicate =
"FALSE", RM = rep(0,px)).
```

The arguments in this function include

- `data`: an $n \times (p + 2)$ matrix of a dataset. The first column records the observed outcome, the second column displays the values for the binary treatment, and the remaining columns store the observed measurements for the confounders.
- `PS`: a specification of a link function $g(\cdot)$ in (2). `logistic` refers to the logistic regression function (3), `probit` reflects the probit model (4), and `cloglog` gives the complementary log-log regression model (5).
- `Psi`: the specification of $\mathcal{C}$ in Step 1.
- `px`: the dimension of X.
- `K`: a positive integer K in Step 1.
- `extrapolate`: the extrapolation function in Step 3. `quadratic` reflects the quadratic polynomial function (16), `linear` gives the linear polynomial function (17), `RL` is the rational linear function (18), and `cubic` refers to the cubic polynomial function (19).
- `Sigma_e`: the covariance matrix Σ_e for the measurement error model (9).
- `replicate`: the identification of the availability of repeated measurements in the confounders. `replicate = "FALSE"` represents no repeated measurements and `replicate = "TRUE"` indicates that repeated measurements exist in the dataset. The default is set as `replicate = "FALSE"`.
- `RM`: a p_X dimensional user-specified vector with each entry representing the number of repetitions for the respective confounder. For example, `RM = c(2,2,3)` indicates that three confounders in X have repeated measurements, where the first and second confounders have two repetitions and the third one has three repetitions. The default of `RM` is set as the p_X-dimensional zero vector, i.e., `RM = rep(0,px).`

In the argument data, the potential outcome can be continuous or binary and the treatment in the second column is designed to be binary. For the columns of confounders, one should place error-prone confounders from the third to the (px+2)-th column, and the remaining columns record precisely-measured confounders in Z. The argument PS is used to specify a link function $g(\cdot)$ that is used to characterize the propensity score in model (2), where the logistic regression model (3) is taken as the default. Two arguments Psi and K are used to generate the working data $X_i^*(k, \psi)$ in Step 1 in Section 3.1. The default of Psi is given by seq(0,1,length = 10), i.e., an interval [0, 1] with equal width divided into $M = 10$ subintervals, and the default of K is set as 200. px reflects the dimension of error-prone confounders, with the default value set as the dimension of all confounders W_i, revealing that all confounders may be subject to measurement error. Setting px = 0 accommodates the situation where all confounders are precisely measured and there is no need to correct the measurement error effects. On the contrary, if confounders do involve measurement error, specifying px = 0 yields the *naive* estimate of τ_0 which ignores the measurement error effects. The argument extrapolate contains commonly used working functions for extrapolation listed in Section 4.2. The default of the working extrapolation function is taken as the quadratic function.

Finally, Sigma_e records the covariance matrix Σ_e, which can be user-specified or estimated by using auxiliary information, as described in Sections 3.2-3.4. Two arguments replicate and RM are used to indicate the availability of repeated measurements and the way of generating working data. Specifically, when replicate = "FALSE", then (10) is implemented to generate the working variables in the main dataset for Cases 1, 2, and 4, respectively described in Sections 3.1, 3.2, and 3.4. On the other hand, the argument replicate = "TRUE" reflects Case 3 described in Section 3.3, which uses (14) to replace Step 1 in Section 3.1. The argument RM is in use to accompany with the argument replicate. When replicate = "FALSE", no repeated measurements are available in the sample, and RM should be set as RM = rep(0,px). In contrast, if replicate = "TRUE", there are repeated measurements for the confounders, and in this case, users should specify the number of repeated measurements for each confounder by setting a proper value for RM. For example, setting RM = c(2,2,3) represents that three confounders have repeated surrogate measurements, having 2, 2, and 3 replicates, respectively. When all arguments are specified, the output of this function gives a vector $\tilde{\gamma}$ as defined in Step 3.

VSE_PS

The function VSE_PS, reflecting *variable selection and estimation of propensity scores*, is used to implement Step 4 in Section 3.1. The input function is given by

```
VSE_PS(V, y, method="lasso", cv="TRUE", alpha = 1),
```

with the following arguments:

- V: a $(p+1)\times(p+1)$ matrix V_n in (11).
- y: a $(p+1)$-dimensional vector $\tilde{\gamma}$ in (11).
- method: it reflects the penalty function $\rho_\lambda(\cdot)$ in (11) with choices presented in Section 4.3, where "lasso", "scad" and "mcp" are given by (20), (21) and (22), respectively.
- cv: the method for choosing the tuning parameter λ. cv="TRUE" suggests the use of the cross-validation method and cv="FALSE" allows the use of the BIC, described in Section 4.4.
- alpha: a constant $\alpha \in [0, 1]$ in (23).

The argument `V` is a user-specified matrix, with the default set as the identity matrix, and `y` represents a vector derived by the output of `SIMEX_EST`. The argument `method` provides the penalty functions in Section 4.3 that are implemented in the R package **ncvreg**. The argument `cv` gives two choices to determine the optimal tuning parameter, respectively determined by cross-validation and BIC. Finally, `alpha` reflects a user-specified value α in (23), with the default value `alpha = 1` that recovers the lasso method.

The output of this function gives a $(p + 1)$-dimensional vector of the estimator of γ. In this vector, components with zero values represent confounders that are unimportant and should be excluded; components with nonzero values identify important confounders entering the treatment model (12).

EST_ATE

Upon the implementation of Steps 1-4, we then use the function `EST_ATE` to estimate ATE, as discussed in Step 5. The implementation is given by

```
  EST_ATE(data, PS = “logistic”, Psi = seq(0,1,length = 10),
K = 200, gamma, px = p, extrapolate=“quadratic”, Sigma_e,
replicate = “FALSE”, RM = 0, bootstrap = 100).
```

All the arguments in this function are the same as those in `SIMEX_EST`, except for the argument `gamma`. The argument `gamma` records the estimate obtained from the implementation of Steps 1-4, and is used to estimate the propensity score $\widehat{\pi}_i(k, \psi)$ in Step 5. The function `EST_ATE` provides the final estimate of ATE.

Furthermore, to provide a variance estimate and the resulting p-value for the estimated ATE, we employ the bootstrap algorithm by repeatedly running the proposed procedure to a sequence of bootstrap samples; then using the resulting estimates of ATE, we compute the sample variance of those estimates; taking this as a bootstrap variance for the initially obtained estimate of ATE, we calculate an associated p-value. The argument `bootstrap` is used to specify the number of bootstrap samples the user wishes to consider; its default value is set as 100. Function `EST_ATE` outputs values with headings `estimate`, `variance`, and `p-value`, which are a point estimate, the associated variance estimate, and the resulting p-value of τ_0, respectively.

6 Numerical studies

6.1 Implementation of AteMeVs

In this section, we implement the R package **AteMeVs** to the METABRIC data described in Section 1. The dataset contains the information for 1422 patients, together with 331 gene expressions. Following the notation in Section 2, we define Y and T as “overall_survival” and “hormone_therapy”, respectively. We let W denote those gene expressions. The following code is used to prepare for the dataset.

```
read.table(“C://METABRIC_causal.csv”, sep = “,”, header = TRUE) ->
data_METABRIC

library(MASS)
library(ncvreg)
library(AteMeVs)

data = data_METABRIC[, 1:280]
gene = colnames(data_METABRIC)[3:280]
```

```
set.seed(20651252)
n = dim(data)[1]
p = dim(data)[2] - 2
```

We now demonstrate Steps 1-3 by using the function `SIMEX_EST` and by setting $K = 10$ and $\mathcal{C}$ to include the cutpoints equally dividing the interval [0, 2] into $M = 10$ subintervals. Since there is no additional information to estimate Σ_e, we follow Case 1 in Section 2.3 to specify Σ_e as a diagonal matrix with common value `s2 = 0.2`. As noted in [32], measurements of gene expressions are subject to measurement error, and therefore, we specify `px = p`, which is 331. The implementation is given below:

```
Psi = seq(0, 2, length = 10)
K = 10
s2 = 0.2

y = as.vector(SIMEX_EST(data, Psi, K, px = p, Sigma_e = diag(s2, p)))
matrix(y,ncol=2)
              [,1]         [,2]
 [1,] -0.132002775 -0.098403692
 [2,]  0.192014131 -0.247497662
 [3,]  0.333889793  0.153789525
 [4,] -0.470314553 -0.486142020
 [5,] -0.164594040 -0.040468911
 [6,] -0.190031631  0.041317059
 [7,]  0.443131214  0.108216814
 [8,] -0.612154496  0.014734219
 [9,]  0.201441389 -0.012663900
[10,]  0.611405760 -0.242673669
```

Due to the space constraint, we report partial results for the output `y` above to show the estimate $\tilde{\gamma}$.

Next, we use $\tilde{\gamma}$ to demonstrate variable selection in Step 4. Since V_n in (11) is user-specified, we follow [23] and set V_n as the identity matrix. To see the impact of variable selection by different methods, we examine three penalty functions (20), (21), and (22). Detailed demonstrations with an application of the function `VSE_PS` are given below. We also display some results in the command "`VS`" as follows.

```
V = diag(1, length(y), length(y))
est_lasso_cv = VSE_PS(V, y, method = “lasso”, cv = “TRUE”)
est_scad_cv = VSE_PS(V, y, method = “scad”, cv = “TRUE”)
est_mcp_cv = VSE_PS(V, y, method = “mcp”, cv = “TRUE”)
cbind(est_lasso_cv,
      est_scad_cv,
      est_mcp_cv) -> VS
rownames(VS) = gene
```

```
VS
       est_lasso_cv est_scad_cv est_mcp_cv
brca1     0.0000000   0.0000000  0.0000000
brca2     0.0000000   0.0000000  0.0000000
palb2     0.0000000   0.0000000  0.0000000
pten      0.0000000   0.5182603  0.5230828
tp53      0.0000000   0.0000000  0.0000000
atm       0.0000000   0.0000000  0.0000000
cdh1      0.0000000   0.0000000  0.0000000
chek2     0.5457605   0.6601017  0.6649249
nbn       0.0000000   0.0000000  0.0000000
nf1      -0.5277423  -0.5634578 -0.5586342
stk11     0.0000000   0.0000000  0.0000000
bard1     0.0000000   0.0000000  0.0000000
mlh1      0.0000000   0.0000000  0.0000000
msh2      0.5342537   0.6485958  0.6534201
msh6      0.0000000   0.0000000  0.0000000
pms2      0.0000000   0.0000000  0.0000000
epcam     0.0000000   0.0000000  0.0000000
rad51c    0.0000000   0.0000000  0.0000000
rad51d    0.0000000   0.0000000  0.0000000
rad50     0.0000000   0.0000000  0.0000000
rb1      -0.8071897  -0.8429033 -0.8380785
rbl1      0.0000000   0.0000000  0.0000000
rbl2      0.0000000   0.0000000  0.0000000
ccna1     0.0000000   0.0000000  0.0000000
ccnb1     0.0000000   0.0000000  0.0000000
cdk1      0.0000000   0.0000000  0.0000000
ccne1     0.0000000   0.0000000  0.0000000
cdk2      0.0000000   0.0000000  0.0000000
cdc25a   -0.5527873  -0.5885001 -0.5836741
ccnd1     0.5691736   0.6835180  0.6883442
cdk4      0.5165555   0.6309001  0.6357266
cdk6     -0.7121596  -0.7478720 -0.7430453
ccnd2     0.0000000   0.0000000  0.0000000
cdkn2a   -0.5822496  -0.6179617 -0.6131343
cdkn2b    0.0000000   0.0000000  0.0000000
myc      -0.6168412  -0.6525533 -0.6477256
cdkn1a    0.0000000   0.0000000  0.0000000
cdkn1b    0.0000000   0.0000000  0.0000000
e2f1      0.0000000   0.0000000  0.0000000
e2f2      0.0000000  -0.5323233 -0.5274951
```

Variable selection shows that zero values correspond to unimportant gene expressions and nonzero ones suggest important gene expressions. The results show that informative gene expressions are sparse regardless of variable selection methods. In the partial results displayed here, we observe that some gene expressions, such as "e2f2" and "pten", are selected by (21) and (22) but not by (20). Moreover, selected gene expressions contain "cdk4", "cdk6", and "ccnd1", which is consistent with the findings of [38, 39]; they found that "ccnd1" was associated with a good breast cancer prognosis and cdk6 has been shown to be regulated and influenced by several mitogenic signaling pathways in breast cancer.

Finally, using the selected gene expressions, we estimate ATE by using the function `EST_ATE`; the estimation result is shown as follows:

```
ate_lasso_cv = EST_ATE(data,
                        gamma = est_lasso_cv,
                        px = p,
                        Sigma_e = diag(s2, px))
ate_scad_cv = EST_ATE(data,
                       gamma = est_scad_cv,
                       px = p,
                       Sigma_e = diag(s2, px))
ate_mcp_cv = EST_ATE(data,
                      gamma = est_mcp_cv,
                      px = p,
                      Sigma_e = diag(s2, px))
> ate_mcp_cv
     estimator   variance      p-value
[1,]    1.4773 0.04885097 2.326125e-11
> ate_scad_cv
     estimator   variance       p-value
[1,] 1.631049  0.04049768 5.275382e-16
> ate_lasso_cv
     estimator  variance   p-value
[1,] 1.116406  0.0357837 3.597e-09
```

An estimate of ATE is 1.4773, 1.631049, and 1.116406, respectively corresponding to the estimates of γ derived from (20), (21), and (22); and associated variance estimates derived by the three methods are 0.04885097, 0.04049768, and 0.0357837, respectively. All the three resulting p-values are smaller than the significance level 0.05, suggesting that the chemotherapy treatment has a positive causal effect on increasing the survival of a patient. These results are derived by accommodating the effects of informative gene expressions, including "ccnd1", "cdk4", and "cdk6", which are also identified to be informative by [9].

6.2 Comparisons of AteMeVs with other methods

To highlight the advantages of the package **AteMeVs** and underscore the importance of addressing issues of measurement error and variable selection, we consider two additional scenarios: (i) using **AteMeVs** without implementing `VSE_PS`, and (ii) using existing packages **iWeigReg** and **CausalGAM**, in comparison to the use of **AteMeVs** with different penalty functions. In Scenario (i), we correct for measurement error in confounders but do not address the exclusion of irrelevant confounders; in Scenario (ii), we aim to estimate ATE without taking measurement error and variable selection into account. We summarize the numerical results in Table 1, where 'EST' represents the estimate of ATE, 'VAR' is the variance associated with the estimated ATE derived from the packages, and 'p-value' is the p-value derived from testing the null hypothesis H_0: $\tau_0 = 0$.

The results show that the package **AteMeVs** performs stably, irrespective to the degree of measurement error and the choice of the penalty function; significant causal effects are

Table 1. Comparisons of estimation methods. LASSO(x) is the usage of the package AteMeVs with the LASSO penalty function, SCAD(x) is the usage of the package AteMeVs with the SCAD penalty function, MCP(x) is the usage of the package AteMeVs with the MCP penalty function, Full(x) refers to Scenario (i), where x = 0.2, 0.5 or 0.7, representing identical diagonal elements in Σ_e. The usage of iWeigReg and CausalGAM refer to Scenario (ii).

	Method	**Estimator of ATE**		
		EST	**VAR**	**p-value**
AteMeVs	LASSO(0.2)	1.477	0.049	2.326e-11
	LASSO(0.5)	1.739	0.043	6.83e-17
	LASSO(0.7)	1.590	0.071	2.525e-09
	SCAD(0.2)	1.631	0.040	5.275e-16
	SCAD(0.5)	1.115	0.102	0.000
	SCAD(0.7)	0.661	0.097	0.034
	MCP(0.2)	1.116	0.036	3.597e-19
	MCP(0.5)	1.274	0.066	7.084e-07
	MCP(0.7)	1.440	0.099	4.599e-06
Scenario (i)	Full(0.2)	0.027	0.031	0.878
	Full(0.5)	0.300	0.079	0.285
	Full(0.7)	0.005	0.075	0.986
Scenario (ii)	**iWeigReg**	3.168	0.741	0.000
	CausalGAM	0.121	0.038	0.534

https://doi.org/10.1371/journal.pone.0296951.t001

revealed by the use of **AteMeVs**. In contrast, Scenario (i) shows insignificant causal effects with p-values greater than 0.05, which might be caused by the involvement of irrelevant confounders, even though measurement error correction is taken into account. Moreover, with the ignorance of measurement error effects and variable selection, it is interesting that the package **iWeigReg** shows evidence for the significance of the causal effects, but the EST and VAR are greater than those derived by the package **AteMeVs**. On the other hand, **CausalGAM** does not provide evidence for suggesting ATE differs zero.

7 Discussion

The inverse-probability-weighting estimation method and its variants have proved to be useful for estimating the average treatment effect within the causal inference framework. However, their applications are hindered by two critical conditions. The validity of those methods relies on the proper determination of propensity scores and the use of the precise measurements of the covariates. When data lack these features [23], introduced a simulation-based method that adapts the inverse-probability-weighting scheme to accommodate measurement error effects as well as variable selection for calculating propensity scores. In this paper, we develop an R package, called **AteMeVs**, to extend the method proposed by [23]. This package provides analysts a user-friendly tool for estimating the average treatment effect when working with error-contaminated data and inconsequential confounders.

As the package **AteMeVs** is designed to handle classical measurement error model (9), biased estimation is anticipated when model (9) is not feasible; the impact of the violation of the model (9) was explored by [40] for survival analysis with covariate measurement error. Further, the development of the package **AteMeVs** lies on the correct parametric modelling for the propensity score. When such an assumption is untrue, estimation results obtained from using **AteMeVs** may become invalid. However, in applications, the relationship between the treatment and the confounders can be complex, making it difficult to have straightforward representation through a convenient parametric model. It is useful to introduce semiparametric models to characterize propensity scores.

While the package **AteMeVs** offers flexibility in handling measurement error and variable selection, it has limitations. Currently, the package focuses on continuous error-prone random variables, and it cannot handle error-contaminated confounders that are mixed with both continuous and discrete variables. Another notable issue concerns the size of data. The **AteMeVs** is basically developed for settings where the number of confounders is smaller than the sample size, which is driven by the setup considered in [23]. It is interesting to generalize the method in [23] to handle high-dimensional error-prone data, where the dimension of confounders can be diverging as the sample size approaches infinity. Creating R packages to conduct causal inference about such data can be useful.

Finally, the package **AteMeVs** developed here can only handle outcomes with complete observations. In the presence of incomplete responses with error-contaminated covariates, such as survival data with covariate measurement error, it is important to address both the censoring effects (e.g., [41]) and the measurement error effects when estimating causal effects. It is interesting to devise causal inference methods to handle such data and then develop R packages accordingly to extend the application scope of the package **AteMeVs**.

Acknowledgments

The authors thank two referees for their useful comments to significantly improve the presentation of the initial manuscript.

Author Contributions

Conceptualization: Li-Pang Chen, Grace Y. Yi.

Methodology: Li-Pang Chen, Grace Y. Yi.

Software: Li-Pang Chen.

Supervision: Grace Y. Yi.

Visualization: Li-Pang Chen, Grace Y. Yi.

Writing – original draft: Li-Pang Chen, Grace Y. Yi.

References

1. Narrandes S. and Xu W. (2018). Gene expression detection assay for cancer clinical use. *Journal of Cancer*, 9, 2249–2265. https://doi.org/10.7150/jca.24744 PMID: 30026820
2. Mulford A. J., Wing C., Dolan M. E., and Wheeler H. E. (2021). Genetically regulated expression underlies cellular sensitivity to chemotherapy in diverse populations. *Human Molecular Genetics*, 30, 305–317. https://doi.org/10.1093/hmg/ddab029 PMID: 33575800
3. Pereira B., Chin SF., Rueda O. et al. (2016). The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications* 7, 11479. https://doi.org/10.1038/ncomms11479 PMID: 27161491
4. Tan, Z. and Shu, H. (2013). iWeigReg: Improved methods for causal inference and missing data problems. R package version 1.0.
5. Ratkovic, M (2015). SVMMatch: Causal effect estimation and diagnostics with support vector machines. R package version 1.1.
6. Glynn, A. and Quinn, K. (2010). CausalGAM: Estimation of causal effects with generalized additive models. R package version 0.1-3.
7. Kim, I.S. and Imai, K. (2014). wfe: Weighted linear fixed effects regression models for causal inference. R package version 1.3.
8. Tingley D., Yamamoto T., Hirose K., Keele L., and Imai K. (2014). "mediation: R package for causal mediation analysis. *Journal of Statistical Software*, 59, 1–38. https://doi.org/10.18637/jss.v059.i05
9. Kato S., Okamura R., Adashek J.J., Khalid N., Lee S., Nguyen V., et al. (2021). Targeting G1/S phase cell-cycle genomic alterations and accompanying co-alterations with individualized CDK4/6 inhibitor-

based regimens. *JCI Insight*, 11; 6(1):e142547. https://doi.org/10.1172/jci.insight.142547 PMID: 33427211

10. King, G., Lucas, C., and Nielsen, R. (2015). MatchingFrontier: R package for computing the matching frontier. R package version 1.0.0.

11. Bang H. and Robins J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61, 962–973. https://doi.org/10.1111/j.1541-0420.2005.00377.x PMID: 16401269

12. Lunceford J. K. and Davidian M. (2004). Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study. *Statistics in Medicine*, 23, 2937–2960. https://doi.org/10.1002/sim.1903 PMID: 15351954

13. Rosenbaum P. R. and Rubin D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79, 516–524. https://doi.org/10.1080/01621459.1984.10478078

14. Ertefaie A., Asgharian M., and Stephens D. A. (2018). Variable selection in causal inference using a simultaneous penalization method. *Journal of Causal Inference*, 20170010. https://doi.org/10.1515/jci-2017-0010

15. Koch B., Vock D. M., Wolfson J., and Vock L. B. (2020). Variable selection and estimation in causal inference using Bayesian spike and slab priors. *Statistical Methods in Medical Research*, 29, 2445–2469. https://doi.org/10.1177/0962280219898497 PMID: 31939336

16. Shortreed S. M. and Ertefaie A. (2017). Outcome-adaptive lasso: Variable selection for causal inference. *Biometrics*, 73, 1111–1122. https://doi.org/10.1111/biom.12679 PMID: 28273693

17. Edwards J., Cole S. R., and Westreich D. (2015). All your data are always missing: incorporating bias due to measurement error into the potential outcomes framework. *International Journal of Epidemiology*, 44, 1452–1459. https://doi.org/10.1093/ije/dyu272 PMID: 25921223

18. Imai K. and Yamamoto T. (2010). Causal inference with differential measurement error: nonparametric identification and sensitivity analysis. *American Journal of Political Science*, 54, 543–560. https://doi.org/10.1111/j.1540-5907.2010.00446.x

19. Kyle R. P., Moodie E. E. M., Klein M. B., and Abrahamowicz M. (2016). Correcting for measurement error in time-varying covariates in marginal structural models. *American Journal of Epidemiology*, 184, 249–258. https://doi.org/10.1093/aje/kww068 PMID: 27416840

20. McCaffrey D. F., Lockwood J. R., Setodji C. M. (2013). Inverse probability weighting with error-prone covariates. *Biometrika*, 100, 671–680. https://doi.org/10.1093/biomet/ast022 PMID: 24795484

21. Shu D. and Yi G. Y. (2019a). Causal inference with measurement error in outcomes: Bias analysis and estimation methods. *Statistical Methods in Medical Research*, 28, 2049–2068. https://doi.org/10.1177/0962280217743777 PMID: 29241426

22. Shu D. and Yi G. Y. (2019b). Inverse-probability-of-treatment weighted estimation of causal parameters in the presence of error-contaminated and time-dependent confounders. *The Biometrical Journal*, 61, 1507–1525. https://doi.org/10.1002/bimj.201600228 PMID: 31449324

23. Yi G. Y. and Chen L.-P. (2023). Estimation of the average treatment effect with variable selection and measurement error simultaneously addressed for potential confounders. *Statistical Methods in Medical Research*, 32, 691–711. https://doi.org/10.1177/09622802221146308 PMID: 36694932

24. Chen, L.-P. and Yi, G. Y. (2023). AteMeVs: average treatment effects with measurement error and variable selection for confounders. https://cran.r-project.org/web/packages/AteMeVs/index.html. R package version 0.1.0

25. Rosenbaum P. R. and Rubin D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70, 41–55. https://doi.org/10.1093/biomet/70.1.41

26. Carroll R. J., Ruppert D., Stefanski L. A., and Crainiceany C. M. (2006). *Measurement Error in Nonlinear Models*, 2nd ed., Chapman & Hall.

27. Yi G. Y. (2017). *Statistical Analysis with Measurement Error or Misclassification: Strategy*, *Method and Application*, Springer.

28. Yi G. Y., Delaigle A., and Gustafson P. (2021). *Handbook of Measurement Error Models*. Chapman & Hall/CRC, Boca Raton, FL.

29. Devanarayan V. and Stefanski L. A. (2002). Empirical simulation extrapolation for measurement error models with replicate measurements. *Statistics & Probability Letters*, 59, 219–225. https://doi.org/10.1016/S0167-7152(02)00098-6

30. Yi G. Y., Ma Y. Spiegelman D., and Carroll R. J. (2015). Functional and structural methods with mixed measurement Error and misclassification in covariates. *Journal of the American Statistical Association*, 110, 681–696. https://doi.org/10.1080/01621459.2014.922777 PMID: 26190876

31. Carroll R. J., Lombard F., Küchenhoff H., and Stefanski L. A. (1996). Asymptotics for the SIMEX estimator in structural measurement error models. *Journal of the American Statistical Association*, 91, 242–250. https://doi.org/10.1080/01621459.1996.10476682

32. Chen L.-P. and Yi G. Y. (2021). Analysis of noisy survival data with graphical proportional hazards measurement error models. *Biometrics*, 77, 956–969. https://doi.org/10.1111/biom.13331 PMID: 32687216

33. Yi G. Y., Tan X., and Li R. (2015). Variable selection and inference procedures for marginal analysis of longitudinal data with missing observations and covariate measurement error. *Canadian Journal of Statistics*, 43, 498–518. https://doi.org/10.1002/cjs.11268 PMID: 26877582

34. Tibshirani R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *Series B*, 58, 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

35. Fan J. and Li R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96, 1348–1360. https://doi.org/10.1198/016214501753382273

36. Zhang C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38, 894–942. https://doi.org/10.1214/09-AOS729

37. Zou H., and Hastie T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society*, *sSeries B*, 67, 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

38. Nebenfuehr S., Kollmann K., and Sexl V. (2020). The role of CDK6 in cancer. *International Journal of Cancer*, 147, 2988–2995. https://doi.org/10.1002/ijc.33054 PMID: 32406095

39. Peurala E., Koivunen P., Haapasaari K.-M. Bloigu R., and Jukkola-Vuorinen A. (2013). The prognostic significance and value of cyclin D1, CDK4 and p16 in human breast cancer. *Breast Cancer Research*, 15, Article number: R5. https://doi.org/10.1186/bcr3376 PMID: 23336272

40. Yi G. Y. and He W. (2012). Bias analysis and the simulation-extrapolation method for survival data with covariate measurement error under parametric proportional odds models. *Biometrical Journal*, 54, 343–360. https://doi.org/10.1002/bimj.201100037 PMID: 22685001

41. Cheng Y.-J. and Wang M.-C. (2012) Estimating propensity scores and causal survival functions using prevalent survival data. *Biometrics*, 68, 707–716. https://doi.org/10.1111/j.1541-0420.2012.01754.x PMID: 22834993