eAppendix 1: Overview of TMLE, greedy CTMLE, and scalable CTMLE

Here we provide a brief overview of targeted maximum likelihood estimation (TMLE), the initial implementation of collaborative targeted maximum likelihood estimation (CTMLE), and the scalable version of CTMLE discussed in the manuscript.

For the discussion below, we will use the following notation:

Y_1: the potential outcome under the exposed condition
Y_0: the potential outcome under the unexposed condition
A: a binary exposure
Y: the observed outcome (corresponds with either Y_1 or Y_0 depending on whether the individual received exposure or remained unexposed).
X: a set of observed covariates
$E[Y|A,X]$: the expected value of Y given A and X (the outcome model)
$E[A|X]$: the expected value of A given X (the propensity score)
$E[Y|A,X]_n$: the estimated outcome model
$E[A|\, X]_n$: the estimated propensity score

Overview of TMLE

TMLE is a doubly robust substitution estimation method for a causal parameter (e.g., risk difference, risk ratio, odds ratio). TMLE consists of the following steps, which are discussed in detail elsewhere (van der Laan & Rubin, 2006; Gruber & van der Laan, 2009; Pang et al., 2016):

1. Fit initial models for $E[Y|A,X]$ (the outcome model) and $E[A|X]$ (the propensity score), which we will denote as $E[Y|A,X]_n$ and $E[A|\, X]_n$.
2. Use the estimated propensity score to fluctuate, or target, $E[Y|A,X]_n$ in order to obtain an updated estimate for $E[Y|A,X]$ which we will denote as $E[Y|A,X]_n^*$.
    - This targeting step is carried out by regressing the outcome variable, Y, on a function of the estimated propensity score, $h(E[A|\, X]_n)$, while using the logit of the predicted values from the initial estimate for the outcome model (i.e., logit($E[Y|A,X]_n$)), as the offset (for details on the function $h(E[A|\, X]_n)$, we refer the reader to van der Laan & Rubin, 2006). This regression produces a coefficient, which we will denote as ε. This coefficient along with $h(E[A|\, X]_n)$ is referred to as the "clever covariate" and is used for updating, or fluctuating, $E[Y|A,X]_n$. The model for this updated estimate is expressed as:

$$\text{logit}(E[Y|A,X]_n^*) = \text{logit}(E[Y|A,X]_n) + \varepsilon h(E[A|\, X]_n) \qquad [1]$$

3. For each individual in the study population, use the fitted model for $E[Y|A,X]_n^*$ to obtain predicted values for the outcome under the exposed and unexposed condition (i.e., predicted values for the potential outcomes Y_1 and Y_0).

4. Use the predicted values for the potential responses from step 3 to obtain an estimate of the causal parameter of interest. For example, if the risk difference is the parameter of interest then the TMLE estimate would be equivalent to the expression:

$$\frac{1}{n}\sum_{i=1}^{n}(E[Y|1,X]_n^* - E[Y|0,X]_n^*) \qquad [2]$$

It is important to emphasize that the causal parameter that is estimated in step 4 needs to be defined prior to implementing step 2. This is because the targeting that occurs in step 2 is tailored for a specific causal parameter. More specifically, the function $h(E[A|\ X]_n)$ depends on the influence curve for the parameter of interest (Gruber & van der Laan, 2009). If two causal parameters are of interest (e.g., risk difference and risk ratio), then steps 2 through 4 would have to be repeated since the targeting that occurs in step 2 for the risk ratio could be different than the targeting that occurs for the risk difference. The same argument applies when targeting a population of interest. For example, the targeting step that occurs when estimating the average treatment effect in the population (ATE) is not necessarily equivalent to the targeting step when estimating the average treatment effect in the treated (ATT).

TMLE has a number of desirable properties. First, TMLE estimators are doubly robust in that TMLE will produce unbiased estimates if either $E[Y|A,X]$ or $E[A|X]$ is consistently estimated. TMLE estimators are also asymptotically efficient when both $E[Y|A,X]$ and $E[A|X]$ are consistently estimated.

Overview of the original (greedy) CTMLE algorithm

CTMLE extends TMLE by estimating the PS in a collaborative manner with the outcome. In other words, when fitting a PS model, the CTMLE algorithm considers a variable's relationship with treatment in collaboration with the variable's association with the outcome.

The original implementation of CTMLE (greedy CTMLE) uses an iterative forward selection process to construct a series of propensity score models and corresponding candidate TMLE estimators. Each successive propensity score model that is used in the construction of a given TMLE estimator controls for one additional variable. The CTMLE algorithm then uses cross validation to select the TMLE estimator that minimizes a specified loss function for the outcome (e.g., the cross-validated prediction error for the outcome, negative likelihood, etc.).

The objective of CTMLE is to improve a bias/variance tradeoff for the target parameter by identifying the TMLE estimator that only adjusts for the subset of variables that are necessary to control for confounding without over inflating the variance of the estimate. This is achieved by considering many candidate propensity score models and corresponding TMLE estimators, instead of just a single fitted propensity score and corresponding TMLE estimator. A detailed discussion on CTMLE is provided by van der Laan & Gruber, 2010. The basic structure of the CTMLE algorithm consists of the following steps which are outlined below:

Let the variables i, j, and k represent the following:

> i: a flag variable that takes on a value of 0 or 1, indicating whether the "clever covariate" needs to be updated
> j: an integer representing the current iteration ($j = 1, \dots, k$)
> k: the total number of variables in the dataset

1. Fit an initial outcome model for $E[Y|A, X]$, denoted $E[Y|A, X]_n$.
2. Set $i = 0$ and $j = 1$. If no candidate TMLE estimators have been constructed (i.e., this is the first iteration or equivalently $j = 1$) then proceed to step 2a, otherwise proceed to step 2b.
    a. Construct k propensity score models, where each propensity score model includes one of the k variables in the dataset.
    b. Construct $k - j - 1$ propensity score models. Each propensity score model includes the set of variables from the previously constructed propensity score that corresponds to candidate TMLE estimator number $j - 1$ (candidate TMLE estimator from the previous iteration), and one of the remaining $k - j - 1$ variables in the dataset (note: a total of j variables are included in each fitted propensity score model).
3. For each of the propensity score models constructed in step 2, construct the corresponding TMLE estimator (see discussion on TMLE above).
4. Evaluate the empirical fit for each of the $k - j - 1$ TMLE estimators constructed in step 3 and identify the estimator that best fits the data (i.e., minimizes a specified loss function for the outcome). The next step depends on which of the following conditions is satisfied.
    a. If $j = 1$ (i.e., first iteration) then select this estimator as the first candidate TMLE estimator and evaluate the following 2 conditions.
        i. If the empirical fit of the selected TMLE estimator reduces the chosen loss function relative to the initial outcome model in step 1, then proceed to step 5.
        ii. If the empirical fit of the selected TMLE estimator does not reduce the chosen loss function relative to the initial outcome model in step 1, then use the selected TMLE estimator to update the "clever covariate" (see discussion on TMLE above) and proceed to step 5. (note: the updated clever covariate and selected TMLE estimator will be used when constructing the TMLE estimators in the next iteration).
    b. If $j > 1$ then evaluate the following conditions
        i. If $i = 0$ and the selected TMLE estimator reduces the specified loss function relative to the candidate TMLE estimator from the previous iteration (i.e., iteration $j - 1$), then select this TMLE estimator as the next candidate estimator (candidate estimator number j) and proceed to step 5.
        ii. If $i = 1$, then select this TMLE estimator as the next candidate estimator (candidate estimator number j) and proceed to step 5.
        iii. If $i = 0$ and none of the constructed TMLE estimators from step 3 reduce the loss function relative to the candidate TMLE estimator from the previous iteration (i.e., candidate estimator number $j - 1$), then set $i = 1$ and return to step 3, but now

use the candidate TMLE model from the previous iteration (i.e., candidate estimator $j - 1$) to update the "clever covariate" and construct the TMLE estimators.

5. If $j < k$, then set $j = j + 1$, set $i = 0$, and return to step 2b. Otherwise, proceed to step 6.
6. Steps 1 through 5 result in the construction of k candidate TMLE estimators along with an estimator produced from the initial outcome model from step 1. To select among these candidate estimators, repeat this estimation process using v-fold cross-validation to choose the estimator that minimizes the cross-validated loss function (e.g., prediction error for the outcome, negative likelihood, etc.).

To fix ideas, it is helpful to illustrate the basic outline of the CTMLE algorithm through a simple example. Suppose our dataset consists of a binary exposure, A, an outcome, Y, and three baseline variables, $X = \{X_1, X_2, X_3\}$. The CTMLE algorithm will proceed as follows:

- Fit an initial outcome model. This outcome model could include any combination of the baseline variables.
- Construct three propensity score models: $E[A|X_1]_n$, $E[A|X_2]_n$, and $E[A|X_3]_n$
- Use the fitted propensity score models to update the fit of the initial outcome model to produce three corresponding TMLE estimators which we will denote as $TMLE_{x1,}$ $TMLE_{x2,}$ and $TMLE_{x3}$
- Evaluate the empirical fit for each of the TMLE estimators through a user-specified loss function. For illustrative purposes suppose the selected loss function is the negative likelihood and that $TMLE_{x3}$ minimizes the negative likelihood relative to the other two TMLE estimators. *TMLE*$_{x3}$ is then selected as candidate estimator number 1.
- Now construct two propensity score models $E[A|X_3, X_1\ ]_n$ and $E[A|X_3, X_2\ ]_n$, along with the corresponding TMLE estimators which we will denote as $TMLE_{x3,x1}$ and $TMLE_{x3,x2}$.
- Evaluate the empirical fit for $TMLE_{x3,x1}$ and $TMLE_{x3,x2}$. For illustrative purposes, suppose that neither $TMLE_{x3,x1}$ or $TMLE_{x3,x2}$ reduce the loss function compared to the previous selected candidate estimator ($TMLE_{x3}$). In this case, $TMLE_{x3}$ is used to update the clever covariate. The fitted propensity score models, $E[A|X_3, X_1\ ]_n$ and $E[A|X_3, X_2\ ]_n$, are then used in collaboration with $TMLE_{x3}$ to construct updated versions of $TMLE_{x3,x1}$ and $TMLE_{x3,x2}$ which we will denote as $TMLE^U_{x3,x1}$ and $TMLE^U_{x3,x2}$
- The empirical fit for $TMLE^U_{x3,x1}$ and $TMLE^U_{x3,x2}$ is evaluated and the estimator that results in the greatest reduction in the loss function is selected as candidate estimator number 2. For illustrative purposes, suppose $TMLE^U_{x3,x1}$ is selected.
- Now fit a single propensity score model, $E[A|X_3, X_1, X_2\ ]_n$, and corresponding TMLE estimator $TMLE_{x3,x1,x2}$.
- Evaluate the empirical fit of $TMLE_{x3,x1,x2}$. Suppose $TMLE_{x3,x1,x2}$ reduces the negative likelihood relative to $TMLE_{x3,x1.}$ Then $TMLE_{x3,x1,x2}$ is selected as the third candidate estimator.
- There are now three candidate TMLE estimators along with the estimator from the initial outcome model. The above process is repeated using cross validation to select the estimator that minimizes the cross-validated loss function.

In addition to retaining all of the properties from TMLE, CTMLE has a number of additional advantages. First, CTMLE has a property that has been termed "collaborative double robustness" meaning that the fitted propensity score model only needs to adjust for the residual bias that remains after partial adjustment from the initial fitted outcome model (van der Laan & Gruber, 2010; Gruber, 2010). In other words, it is not necessary for the propensity score or outcome model to fully adjust for confounding by themselves. A consistent estimator can be attained through a collaborative effort of the two, even if neither model by itself is sufficient to produce a consistent estimator. This property makes the CTMLE more robust to model misspecification. Second, since the CTMLE algorithm is able to exclude variables that do not improve confounding control but may increase variability, the CTMLE is more robust in settings involving small samples and rare covariates.

Overview of the scalable CTMLE algorithm

The original implementation of the CMTLE algorithm (greedy CTMLE) is very computationally intensive and not scalable to large healthcare databases. As outlined above, when constructing each successive candidate estimator, the CTMLE algorithm searches through each of the variables in the dataset that were not selected in the previous iteration. With healthcare databases containing thousands of variables that are potentially available for adjustment, this iterative process is not computationally practical.

To make the CTMLE algorithm scalable to large data, the portion of the algorithm that requires iteratively searching through each variable can be modified. This is done by requiring the algorithm to select covariates in a predefined order that is specified by the user. The modified algorithm then constructs propensity score models, and corresponding candidate TMLE estimators, according to the predefined order.

The modified algorithm still produces a total of k candidate TMLE estimators. However, each iteration of the scalable version of the CTMLE algorithm only requires a single propensity score model and corresponding TMLE estimator to be constructed, rather than multiple propensity score models and corresponding TMLE estimators. The steps for the modified algorithm that include a preordering for the variables are outlined below and a detailed discussion on the modified CTMLE is provided by Ju et al. (2016):

1. Fit an initial outcome model for $E[Y|A,X]$.
2. Preorder the variables in the dataset based on some user-specified ordering.
3. Set $i = 0$ and $j = 1$
4. Construct a single propensity score model that includes the first j ordered variables in the dataset.
5. Construct the corresponding TMLE estimator (see discussion on TMLE above).
6. Evaluate the empirical fit for the TMLE estimator from step 5.
    a. If $j = 1$ (i.e., first iteration) then select this estimator as the first candidate TMLE estimator and assess which of the following two conditions are satisfied.
        i. If the empirical fit of the TMLE estimator reduces the chosen loss function relative to the fitted initial outcome model in step 1, then proceed to step 7.
        ii. If the empirical fit of the TMLE estimator does not reduce the chosen loss function relative to the fitted initial outcome model in step 1, then use this TMLE estimator to update the “clever covariate” (see discussion on TMLE above) and proceed to

step 7. (note: the updated clever covariate along with the candidate TMLE estimator will be used for constructing the next TMLE estimator in step 5).

- b. If $j > 1$ then evaluate the following conditions
    - i. If $i = 0$ and the TMLE estimator reduces the specified loss function relative to the candidate TMLE estimator from the previous iteration (i.e., iteration $j - 1$), then select this TMLE estimator as the next candidate estimator (candidate estimator number j) and proceed to step 7.
    - ii. If $i = 1$, then select this TMLE estimator as the next candidate estimator (candidate estimator number j) and proceed to step 7.
    - iii. If $i = 0$ and the current TMLE estimator does not reduce the loss function relative to the candidate estimator from the previous iteration (i.e., candidate estimator number $j - 1$), then set $i = 1$ and return to step 5 using the candidate TMLE model from the previous iteration (i.e., candidate estimator $j - 1$) in the construction of the updated "clever covariate" and TMLE estimator.

7. If $j < k$ then set $j = j + 1$, set $i = 0$, and return to step 4. Otherwise, proceed to step 8.
8. Steps 1 through 7 result in the construction of k candidate TMLE estimators in addition to the estimator produced from the initial outcome model from step 1. To select among these candidate estimators, repeat this estimation process using v-fold cross-validation to choose the estimator that minimizes the cross-validated loss function (e.g., prediction error for the outcome, negative likelihood).

To reduce computation time even further, the above algorithm can be modified to include an early stopping rule. This early stopping rule is defined by an argument called "patience", which is an integer that is specified by the user and is applied in step 8 of the above algorithm. If the cross-validated loss function does not improve after considering a certain number of candidate TMLE estimators (number is specified in the patience setting) then the modified algorithm will stop and will not consider any of the remaining variables in the covariate set. For example, if the patience parameter is set at 10, the modified CTMLE algorithm will stop constructing TMLE estimates if the cross validated prediction error for the outcome does not improve after 10 additional variables are considered.

It is important to emphasize that as long as there is no early stopping, the scalable CTMLE has the same asymptotic behavior as greedy CTMLE (Ju, et al. 2016). However, the scalable CTMLE can have weaker finite sample performance if the pre-ordering strategy is not properly designed. In other words, the scalable CTMLE can be very sensitive to the preordering of variables in finite samples. If variables that are not important confounders appear early in the list (e.g., instrumental variables), the modified CTMLE may not perform optimally.

eAppendix2: Description of Study Cohorts Used in Plasmode Simulations

1. ***Cohort 1 (NSAID dataset)***: Data consists of 49,653 Medicare beneficiaries ages 65 and older who were enrolled in the Pharmaceutical Assistance Contract for the Elderly (PACE) program provided by the state of Pennsylvania. The study cohort included individuals who initiated either a Cox-2 inhibitor or a nonselective NSAID between 1999-2002. Initiation of NSAID or Cox-2 inhibitor was defined as filling at least one prescription for an NSAID, or Cox-2 inhibitor, between January 1, 1999, and December 31, 2002, without any prescription of an NSAID, or Cox-2 inhibitor, during the 18 months prior to the index date (new-user design). Individuals were followed for 180 days after initiation of therapy with the study outcome being gastrointestinal complications. The data consists of 9,470 unique codes within eight data dimensions (prescription drugs, ambulatory diagnoses, hospital diagnoses, nursing home diagnoses, ambulatory procedures, hospital procedures, physician diagnoses, and physician procedures). Each code represents the frequency of times the code appeared in the pre-specified washout period (18 months prior to the index date). These codes were used to create HDPS generated variables as described in the manuscript.
2. ***Cohort 2 (NOAC dataset)***: Data were collected by United Healthcare and consists of 18,447 individuals who initiated either a novel oral anticoagulant (NOAC) or warfarin between October 2009 and December 2012 after a 1 year washout period with no prescription for any oral anticoagulant (new-user design). Data were collected to compare the effect of warfarin versus NOAC on a number of outcomes, including a combined outcome of all-cause mortality, ischemic stroke, and hemorrhagic stroke. Data consisted of 23,531 unique claims codes within 5 data dimensions (inpatient procedures, outpatient procedures, inpatient diagnoses, outpatient diagnoses, and prescription drugs). Each code represents the frequency of times the code appeared in the pre-specified washout period (12 months prior to the index date). These codes were used to create HDPS generated variables as described in the manuscript.
3. ***Cohort 3 (Statin dataset)***: Data consists of a population of elderly adults (>65 years) who were enrolled in Pennsylvania's Pharmaceutical Assistance Contract for the Elderly (PACE) program provided by the state of Pennsylvania. The dataset consists of 36,122 individuals who initiated either a statin or glaucoma medication after no prescription of either a statin or glaucoma medication within the 12 months prior to the index date (new-user design). Individuals were followed to evaluate the protective effect of initiating statin therapy on all-cause mortality. The data consists of 9,470 unique codes within eight data dimensions (prescription drugs, ambulatory diagnoses, hospital diagnoses, nursing home diagnoses, ambulatory procedures, hospital procedures, physician diagnoses, and physician procedures). Each code represents the frequency of times the code appeared in the pre-specified washout period (12 months prior to the index date). These codes were used to create HDPS generated variables as described in the manuscript.

eAppendix 3: Imputing Potential Outcomes from Simulated Data

Recall that for the plasmode simulations constructed in the manuscript, treatment assignment was not simulated and was a function of the true underlying covariate associations in the data. However, the outcome variable was simulated and was a function of a known parametric logistic model. Because the "true" outcome model was known, this model could be used to impute the potential outcomes for each individual under alternative treatment conditions. These potential outcomes, in turn, could be used to calculate the value of the true causal parameter of interest (e.g., risk difference).

Here we provide a simple example of imputing potential outcomes from the "true" outcome model to obtain the value of the causal parameter of interest.

For the discussion below, we will use the following notation:

Y_1: the potential outcome under the exposed condition
Y_0: the potential outcome under the unexposed condition
A: a binary exposure
Y: the observed outcome (corresponds with either Y_1 or Y_0 depending on whether the individual received exposure or remained unexposed).
X: a set of observed covariates
$E[Y|A, X]$: the expected value of Y given A and X (the outcome model)

Suppose our dataset consists of a binary exposure, A, an outcome, Y, and three baseline variables, $X = \{X_1, X_2, X_3\}$. Further, suppose that the true data generating model for outcome probabilities can be expressed as:

$$logit(E[Y|A, X_1, X_2, X_3]) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_A A \qquad [1]$$

The model expressed in Equation 1 can be used to produce outcome probabilities for each individual under the exposed ($A = 1$) and unexposed ($A = 0$) condition. For example, suppose the beta coefficients in Equation 1 are equal to the following:

$$\beta_0 = 0$$
$$\beta_1 = 0.2$$
$$\beta_2 = 0.1$$
$$\beta_3 = 0.4$$
$$\beta_A = -0.3$$

The "true" outcome probability for each individual can be computed by simply plugging in their specific covariate values and exposure status into Equation 1. For example, suppose that the first subject in the dataset has $X_1 = 1$, $X_2 = 1$, $X_3 = 0$, and $A = 1$. Then the predicted outcome probability for subject 1 would be:

$$E[Y|A = 1, X_1 = 1, X_2 = 1, X_3 = 0] = E[Y_1|X_1 = 1, X_2 = 1, X_3 = 0]$$

$$= \text{expit}[0.2(1) + 0.1(1) - 0.3(1)] = 0.5$$

Because subject 1 received exposure, their observed outcome probability is equivalent to their potential outcome probability under the exposed condition as illustrated above. To obtain their counterfactual outcome probability under no exposure, we would simply plug in a value of 0 for exposure status:

$$E[Y_0|X_1 = 1, X_2 = 1, X_3 = 0] = \text{expit}[0.2(1) + 0.1(1)] \approx 0.57$$

Table 1 below shows the counterfactual outcome probabilities for three individuals in a hypothetical dataset, along with their observed outcome probability and baseline covariate values.

Table 1. Counterfactual outcome probabilities for a hypothetical dataset.

Subject	X_{1i}	X_{2i}	X_{3i}	A	$E[Y|A_i, X_{1i}, X_{2i}, X_{3i}]$	$E[Y_0|X_{1i}, X_{2i}, X_{3i}]$	$E[Y_1|X_{1i}, X_{2i}, X_{3i}]$
1	1	1	0	1	0.5	0.57	0.5
2	1	0	0	0	0.55	0.55	0.48
3	0	1	1	0	0.62	0.62	0.55
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The counterfactual outcome probabilities can then be used to generate counterfactual outcomes under exposed and unexposed conditions for each individual. These counterfactual outcomes can then be used to calculate the unbiased or “true” value of the causal parameter of interest. For example, if the parameter of interest is the risk difference, this would be obtained through the expression:

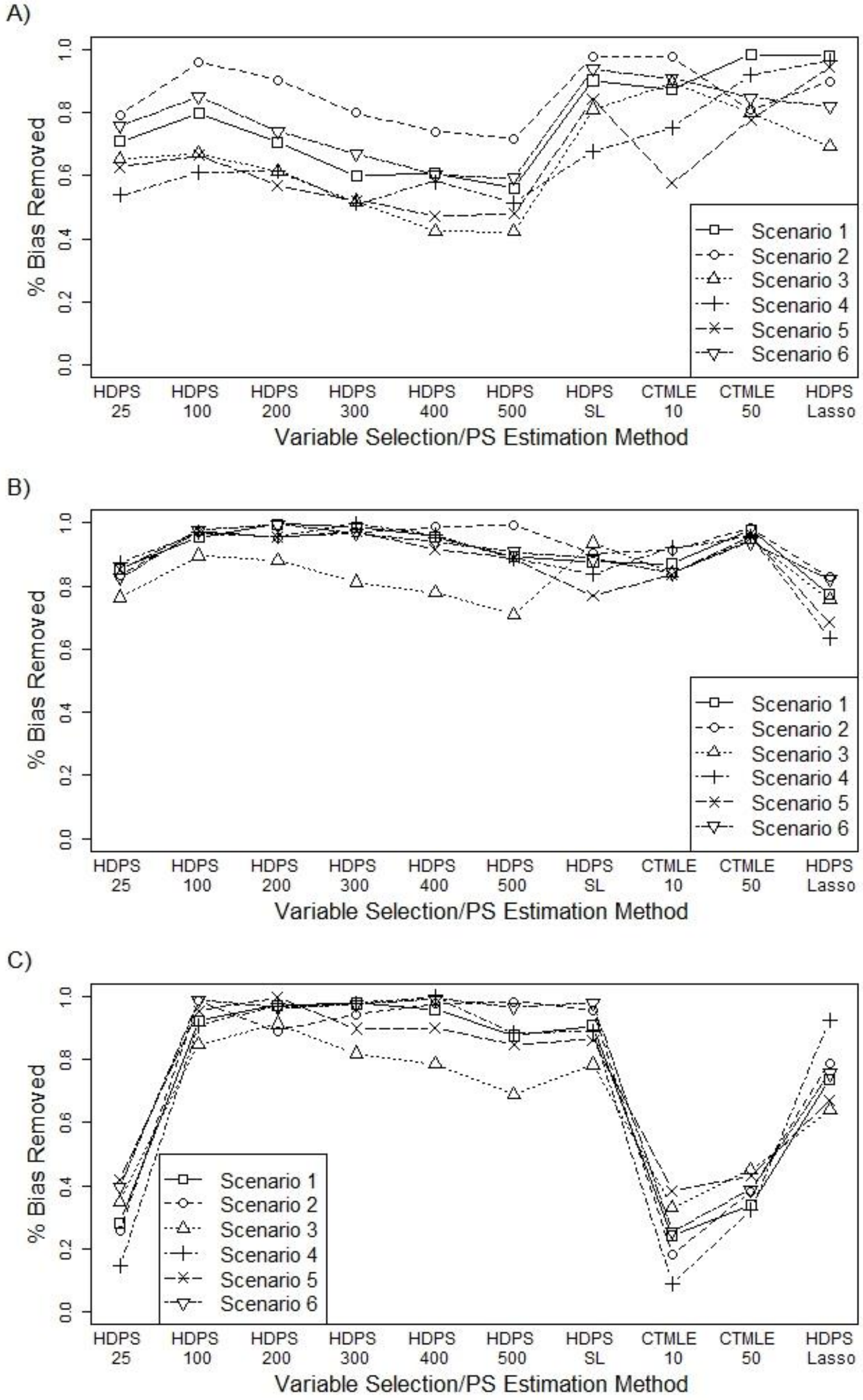
$$\frac{1}{n}\sum_{i=1}^{n}(Y_{1i} - Y_{0i})$$

Where n is the number of individuals in the dataset. Equivalently, the counterfactual outcome probabilities could be used directly in the calculation of the value for the causal parameter of interest. For example, if the risk difference is again the desired parameter, this would be obtained through the expression:

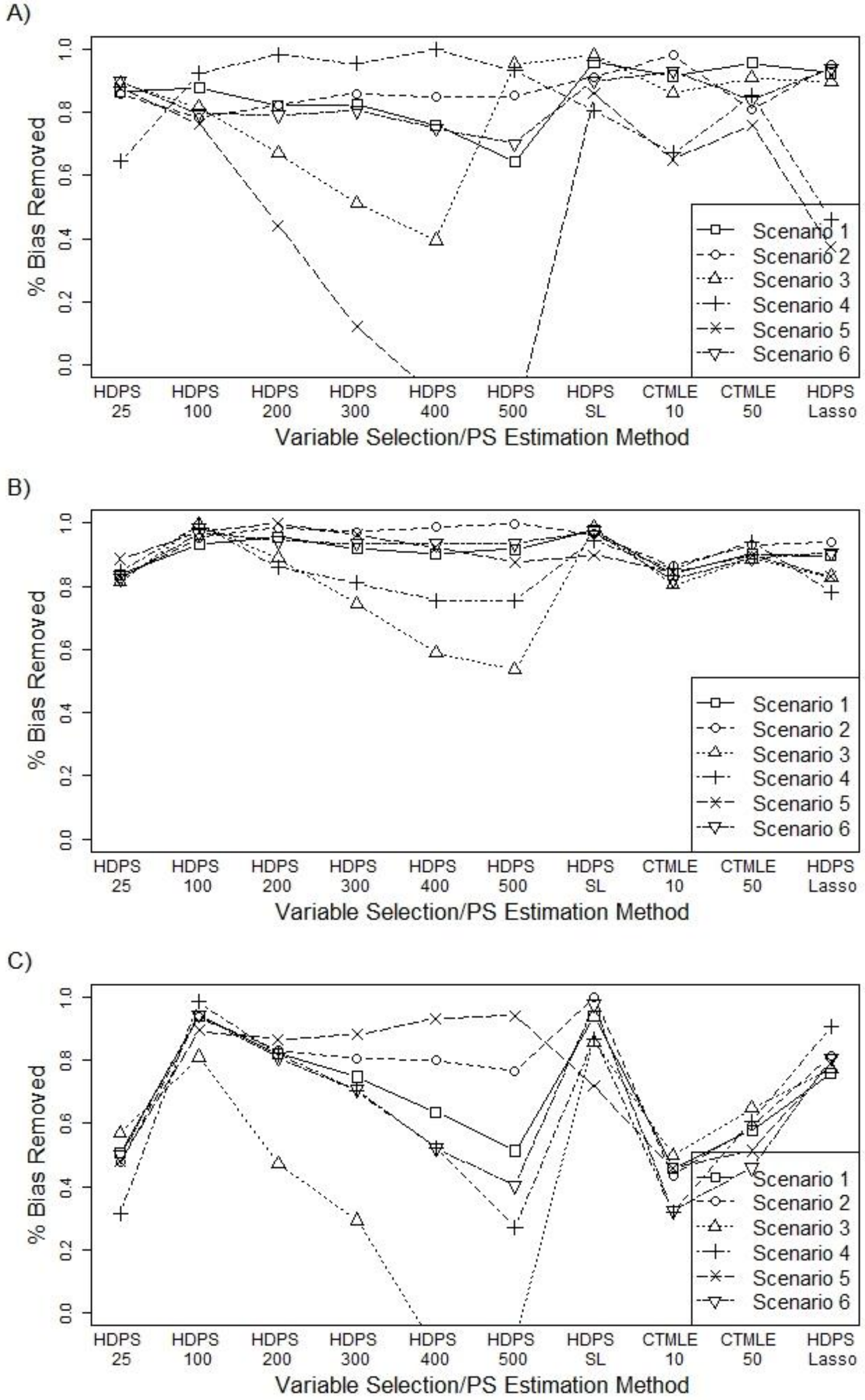
$$\frac{1}{n}\sum_{i=1}^{n}(E[Y_1|X_{1i}, X_{2i}, X_{3i}] - E[Y_0|X_{1i}, X_{2i}, X_{3i}])$$

The calculated value for the causal parameter of interest can be considered the “truth” and used as the benchmark when calculating bias and mean squared error in simulations that evaluate the performance of various statistical models.

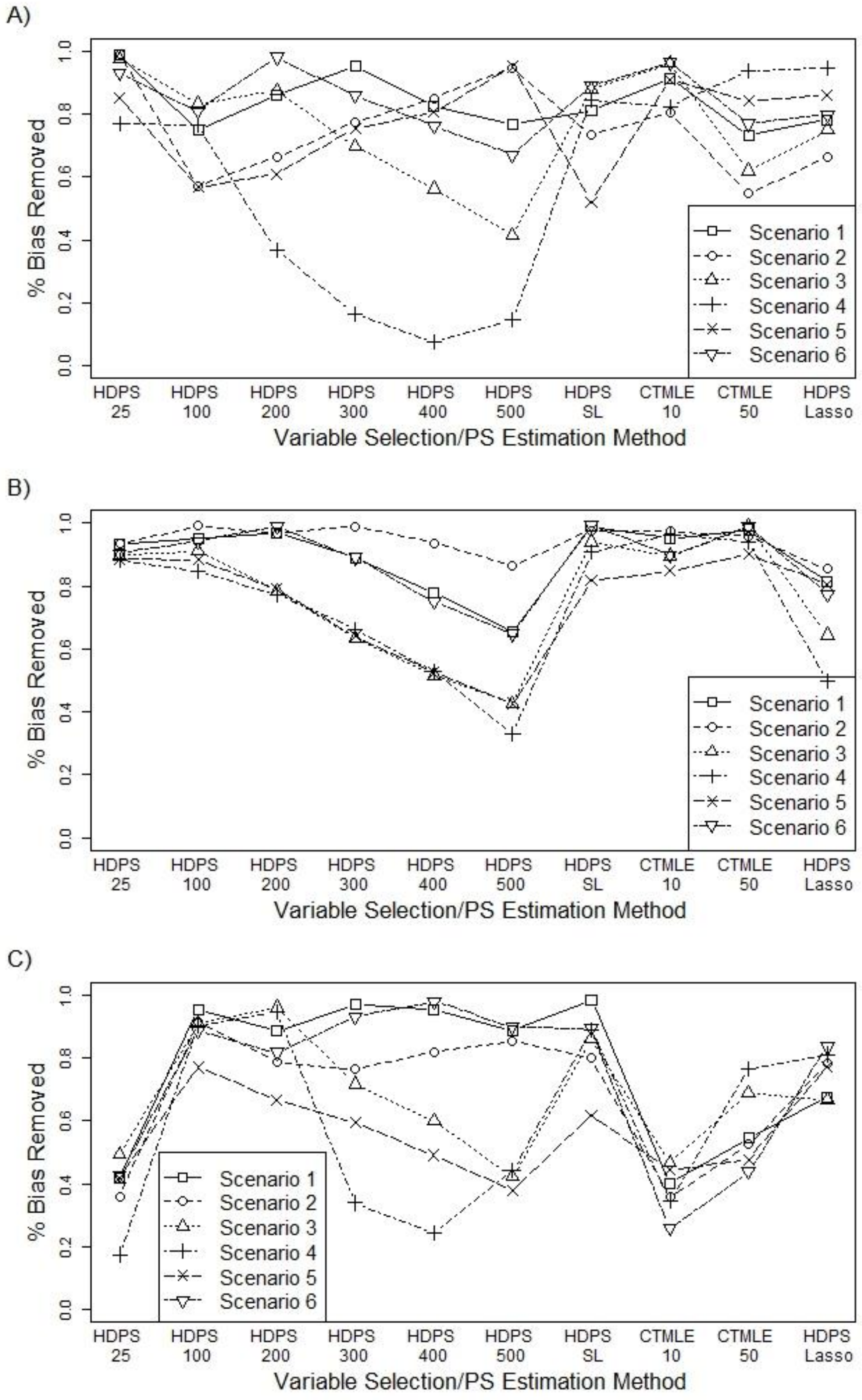
eAppendix 4: Supplemental Figures



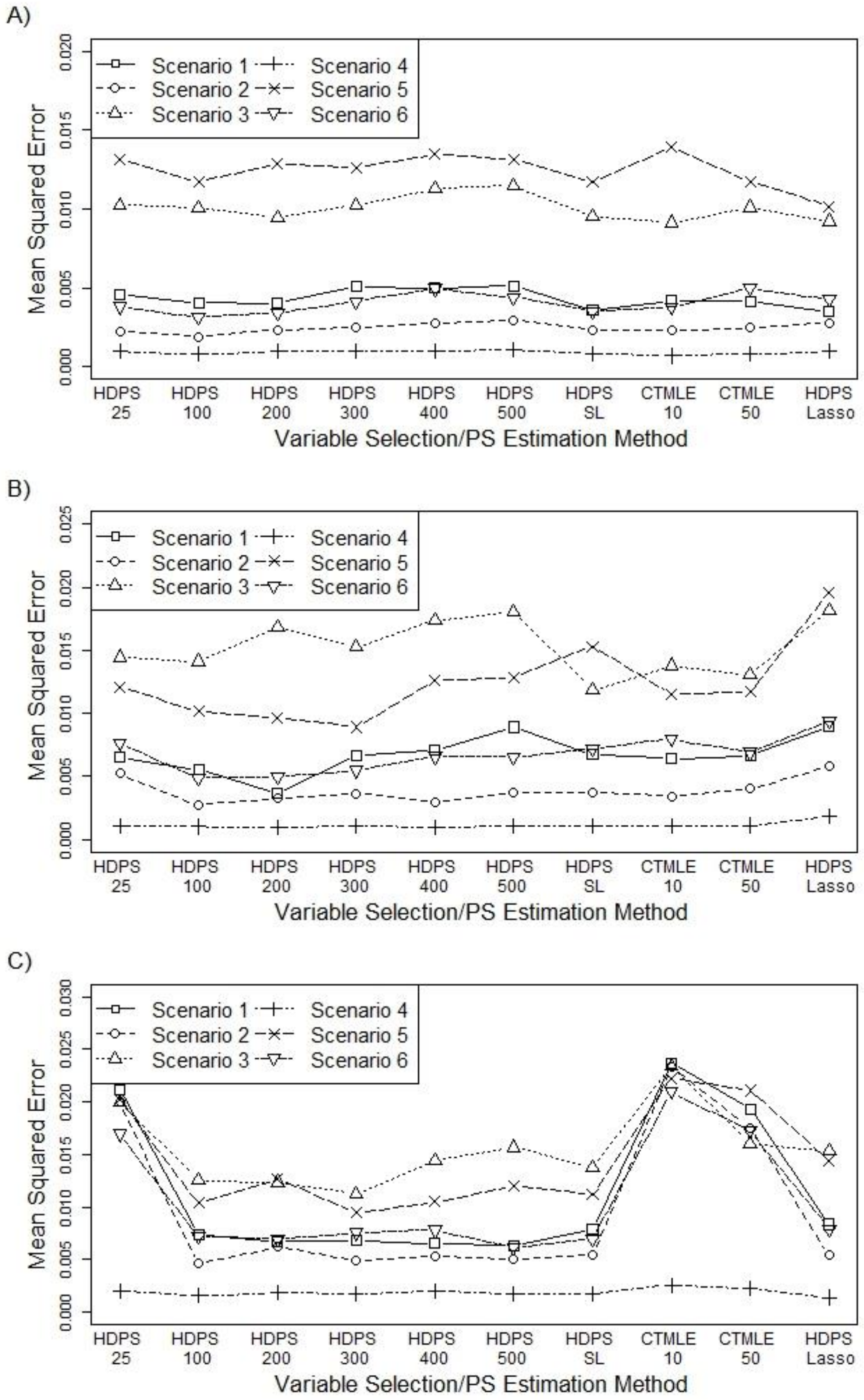
eFigure 1. Percent bias removed for each scenario and variable selection method when matching on the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.



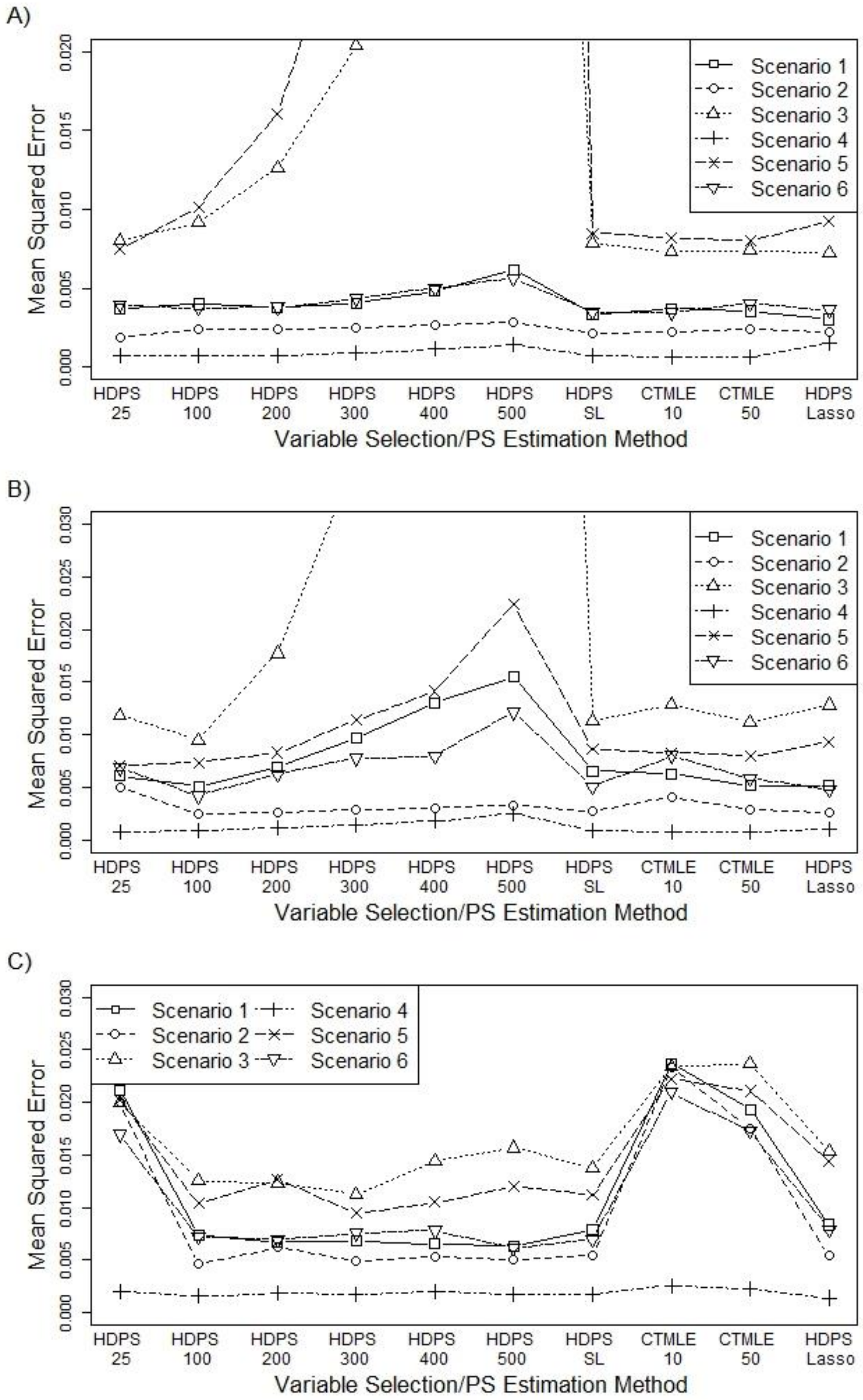
eFigure 2. Percent bias removed for each scenario and variable selection method when using IPTW to implement the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.



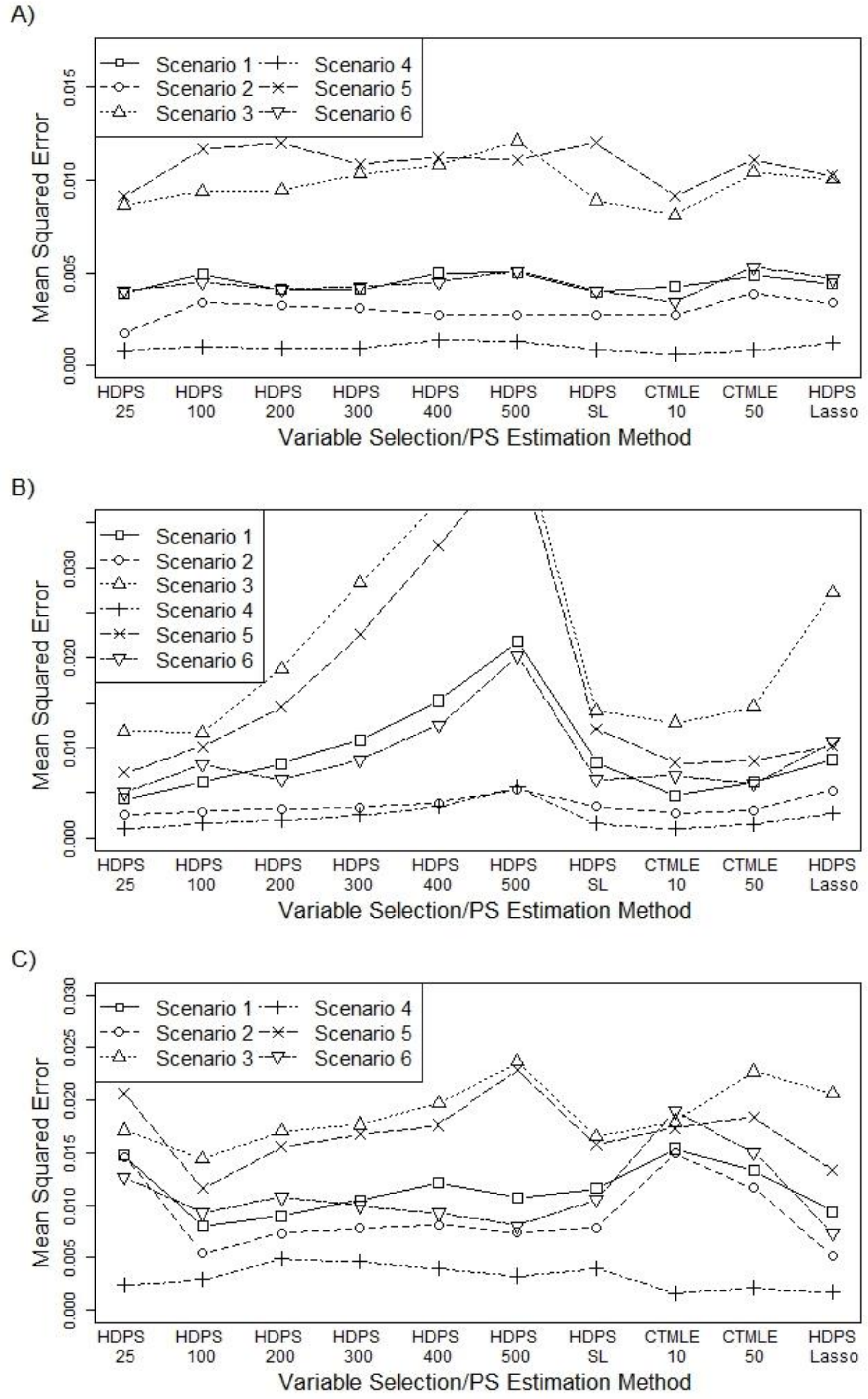
eFigure 3. Percent bias removed for each scenario and variable selection method when using TMLE to implement the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.



eFigure 4. Mean squared error (MSE) for each scenario and variable selection method when matching on the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.



eFigure 5. Mean squared error (MSE) for each scenario and variable selection method when using IPTW to implement the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.



eFigure 6. Mean squared error (MSE) for each scenario and variable selection method when using TMLE to implement the estimated propensity scores. Plots A, B, and C show results for plasmode simulations based on the NSAID, NOAC, and Statin datasets, respectively.

References

Van der Laan MJ, Rubin D. Targeted Maximum Likliehood Learning. Int J Biostat. 2006;2(1).

Gruber S, van der Laan MJ. Targeted maximum likelihood estimation: A gentle introduction. *U.C. Berkeley Division of Biostatistics Working Paper Series.* 2009;Working Paper 252

Pang M, Schuster T, Filion KB, Eberg M, Platt RW. Targeted maximum likelihood estimation for pharmacoepidemiologic research. *Epidemiology*. 2016;27:570-577

Van Der Laan MJ, Gruber S. Collaborative double robust targeted maximum likelihood estimation. Int J Biostat. 2010; 6: 1–68.

Ju C, Gruber S, Lendle SD, Franklin JM, Wyss R, Schneeweiss S, van der Laan MJ. Scalable collaborative targeted learning for large scale and high-dimensional data. *U.C. Berkeley Division of Biostatistics Working Paper Series.* 2016;Working Paper 352