Web-based Supplementary Materials for "Estimation of the Optimal Surrogate Based on a Randomized Trial" by Brenda L. Price, Peter B. Gilbert, and Mark J. van der Laan

Web Appendix A: Inference on the clinical treatment effect in a future study based on the previously estimated optimal surrogate, accounting for estimation error and failure of the transportability assumptions

In Sections 3.1 and 3.2 of the main article we showed conditions on the new study P and current study P_0 under which the P_0-optimal surrogate is also the P-optimal surrogate. That is the best possible scenario as statistical inference for the causal effect of treatment on the P_0-optimal surrogate outcome in the new trial corresponds exactly with statistical inference for the causal effect of treatment on the outcome of interest in the new trial. We now consider a situation in which the new study evaluates a new treatment A^* and we are not willing to assume that the intermediate variables S completely block the effect of treatment (current and new) on the outcome; this situation will be very common. Now we can be certain that the P_0-optimal surrogate $E_{P_0}(Y \mid W, A, S)$ is not equal to the P-optimal surrogate, and, since the P_0-optimal surrogate is a function of A which is not measured/evaluated in the new study, one needs to decide how to even define a surrogate for the future study based on $E_{P_0}(Y \mid W, A, S)$. One can imagine that one would use $E_{P_0}(Y \mid W, A, S)$ as a surrogate if we feel that the treatment $A = 1$ in the P_0-study is most comparable with the treatment $A^* = 1$ in the new P-study. Even though we now have no guarantees, $E_{P_0}(Y \mid W, A, S)$ will often be a good candidate surrogate for such a future study (i.e., one that may approximately satisfy the Prentice definition of a valid surrogate in the future P-study), but one needs to be concerned about the difference between $E_P(Y^* \mid W^* = w, A^* = a, S^* = s)$ and $E_{P_0}(Y \mid W = w, A = a, S = s)$ for $a \in \{0, 1\}$.

To address this issue, suppose that ψ_n converges to ψ_0 at a rate $r(n)$ in the sense that

$d_0(\psi_n, \psi_0) = O_P(r(n))$, where $d_{P_0}(\psi, \psi_0) = P_0 L(\psi) - P_0 L(\psi_0)$ is the loss-based dissimilarity. We also have that $d_0(\psi_n, \psi_0) = O_P(r(n))$, but for simplicity we work with ψ_n. For concreteness, let us consider the squared error loss $L(\psi)(O) = (Y - \psi(W, A, S))^2$. Consider a new study P with data structure (W^*, A^*, S^*, Y^*) for which the Equal Conditional Means condition holds, and for which we only collect the surrogate $\psi_n(W^*, A^*, S^*)$ instead of Y^*.

In this new study the data structure is $(W^*, A^*, S^*, \psi_n(W^*, A^*, S^*))$ and one would target the parameter $\theta^*_P = \theta^*_\psi(P) = E_P\left[E_P(\psi_n(W^*, 1, S^*) \mid A^* = 1, W^*)\right] - E_P\left[E_P(\psi_n(W^*, 0, S^*) \mid A^* = 0, W^*)\right]$. The clinical treatment effect target parameter of this study is $E_P(Y^*_1 - Y^*_0) = \theta^*_P = \theta^*_\psi(P)$

$$E_P\left[E_P(\psi_0(W^*, 1, S^*) \mid A^* = 1, W^*)\right] - E_P\left[E_P(\psi_0(W^*, 0, S^*) \mid A^* = 0, W^*)\right].$$

Suppose that $dP(W^* = w, A^* = a, S^* = s)/dP_0(W = w, A = a, S = s) < M < \infty$ P-a.e. for (w, a, s) in a support of (W^*, A^*, S^*). In that case, it follows that

$$d_P(\psi_n, \psi^*) = \int (\psi_n - \psi^*)^2 (W^*, A^*, S^*) dP(W^*, A^*, S^*) \leqslant M d_{P_0}(\psi_n, \psi^*)$$

where $M d_{P_0}(\psi_n, \psi^*) = O_P(r(n))$. Thus, under this condition, $\theta_{\psi_n}(P) - \theta^*_\psi(P) = O_P(r(n))$.

From this we learn that the estimand defined by the average causal effect of treatment on the surrogate $\psi_n(W^*, A^*, S^*)$ in the future study P will be within distance $O_P(r(n))$ from the desired average causal effect of treatment on the actual outcome Y^*. Suppose that one is only interested in picking up causal effects on Y^* that are larger than some minimal value δ^*. Then, one would want to make sure this remainder $O_P(r(n)) < \delta^*$ so that $\mid \theta_{\psi_n}(P) - \theta^*_\psi(P) \mid < \delta^*$. The difference $\theta_{\psi_n}(P) - \theta^*_\psi(P)$ equals $[\theta_{\psi_n}(P) - \theta_{\psi_0}(P_0)] + [\theta_{\psi_0}(P_0) - \theta^*_\psi(P)]$, showing that the first source of the $O_P(r(n))$ remainder is the discrepancy between the estimated optimal surrogate and the true optimal surrogate in the original trial, and the second source is any violations of the Equal Conditional Means condition. Therefore, under this condition, if the original study were very large such that the first discrepancy is negligible, then the surrogate parameter $\theta_{\psi_\infty}(P)$ studied in the new trial equals the target parameter of interest $\theta^*_\psi(P)$. Thus an infinite original study plus Equal Conditional Means implies that point and confidence interval estimates for $\theta^*_\psi(P)$ can be obtained simply by point and confidence interval estimates for the surrogate effect $\theta_{\psi_\infty}(P)$. In addition, for a finite-sample original study, under Equal Conditional Means $[\theta_{\psi_n}(P) - \theta_{\psi_0}(P_0)]$ measures the bias for estimating $\theta^*_\psi(P)$ based on the estimated optimal surrogate instead of on Y^*. Clearly, the idea is that the estimated optimal surrogate must be a good estimate of the true optimal surrogate in the original study, and $E_{P_0}(Y \mid W = w, A = a, S = s)$ must be a reasonable approximation of $E_P(Y^* \mid W^* = w, A^* = a, S^* = s)$ in the future study, in order to trust our surrogate outcome as a surrogate for the outcome in a future study.

Future work is needed to obtain confidence intervals for $\theta^*_\psi(P) = E_P(Y^*_1 - Y^*_0)$ based on the estimated optimal surrogate instead of on the true optimal surrogate. This problem is readily solved if ψ_n were estimated using a parametric model, in which case the delta method would yield a confidence interval for ψ_0 and for $\theta^*_\psi(P)$, and this parametric model could be selected data-adaptively. However, obtaining a confidence interval when estimating ψ_n nonparametrically through super-learning as we do is much harder, because ψ_0 is a function that is not estimable at root-n rate. For example, the nonparametric bootstrap theoretically fails for machine learning based estimators because of their slower than root-n rate.

Web Appendix B: Connection of the optimal surrogate framework to other surrogate frameworks

Joffe and Greene (2009) classified statistical methods for evaluating the validity of candidate surrogate endpoints into four frameworks, which may be referred to as the Prentice replacement endpoint, controlled direct effects, principal stratification, and meta-analysis frameworks. Prentice (1989) catalyzed the field with his definition of a valid surrogate endpoint and operational criteria, as “a response variable for which a test of the null hypothesis of no relationship to the treatment groups under comparison is also a valid test of the corresponding null hypothesis based on the true endpoint." Prentice (1989) also provided operational criteria for checking whether an intermediate endpoint satisfies this definition, the most important being the 'full mediation' criterion that the distribution of the clinical endpoint conditional on the surrogate is the same as the distribution of the clinical endpoint conditional on the surrogate and treatment, and many subsequent papers developed methods for evaluating these criteria or related criteria [e.g., Freedman et al. (1992); Lin, Fleming, and DeGruttola (1997); Wang and Taylor (2002); Buyse and Molenberghs (1998); Alonso (2006); Weir and Walley (2006); Kobayashi and Kuroki (2014)]. Noting that the Prentice approach is based purely on statistical parameters and the study of associations between observable random variables, Joffe and Greene (2009) suggested an alternative framework based on controlled direct and indirect causal parameters that assume experimental manipulation of the hypothesized surrogate, a framework also studied by Robins and Greenland (1992) and Pearl (2001). While this controlled effects framework has major advantage to address questions about how interventions on the surrogate causally effect the clinical outcome, it is challenged by questions of conceivability of the causal target parameters in some settings (Gilbert, Hudgens, and Wolfson, 2011) and by difficulties in justifying assumptions used to identify the causal parameters.

Observing that many early methods for assessing the Prentice criteria did not account for the fact that baseline predictors of both the surrogate and clinical outcome must be correctly controlled for, Frangakis and Rubin (2002) introduced the principal stratification framework that studies how the clinical treatment effect varies over principal strata subgroups defined by the potential surrogate endpoints under each of the two treatment assignments. Many statistical methods papers in this framework have followed, including Gilbert and Hudgens (2008), including Taylor, Wang, and Thiebaut (2005), Gilbert and Hudgens (2008), van der

Weele (2008), Li, Taylor, and Elliott (2010), Huang, Gilbert, and Wolfson (2013), and Gilbert et al. (2015). The meta-analysis framework studies the association of treatment effects on the surrogate outcome with treatment effects on the clinical outcome [e.g., Daniels and Hughes (1997), Buyse et al. (2000), and Gail et al. (2000)], with advantage that the treatment effects are causal effects based on the randomization and are estimable from standard assumptions.

VanderWeele (2013) reviewed how these four frameworks relate to criteria for guaranteeing a consistent surrogate, and Gilbert et al. (2015) studied relationships between principal stratification criteria and the Prentice definition. Except for a segment of the meta-analysis literature, there is quite limited surrogate endpoint evaluation literature on methods for applying and assessing the validity of a surrogate endpoint in a new trial for inferring the causal treatment effect in that trial without including clinical endpoint data (Gilbert et al., 2015). The small size of this literature may be surprising given the centrality of this objective in biomedical applications. Pointing to this gap in the literature, Pearl and Bareinboim (2011, 2012) introduced the causal selection diagram approach, to estimation and testing of the clinical treatment effect in a new setting based on a surrogate and baseline covariates, which may be viewed as a fifth framework for surrogate endpoint evaluation.

Our newly proposed approach does not fit squarely into any of the five frameworks, thereby constituting a sixth framework that we name the optimal surrogate approach. It departs from the principal stratification and controlled effects frameworks, aligning more closely with the other three, in being based purely on statistical parameters that are estimable under the basic assumptions typically made in randomized clinical trials. In particular, it aligns with the Prentice framework by taking as its starting point the excellent Prentice definition of a valid surrogate endpoint. In fact, the optimal surrogate is constructed to guarantee satisfaction of the Prentice definition, a unique advantage compared to previous approaches. Our approach also departs from previous approaches by defining the optimal surrogate as an unknown parameter, such that its predicted values are used as the surrogate endpoint. Because this estimated optimal surrogate is consistent under standard assumptions, in trials with large sample sizes it approximately satisfies the Prentice definition.

The optimal surrogate approach is related to Prentice's (1989) operational criteria. First, the best optimal surrogate will have treatment and candidate surrogate separately highly predictive of the final outcome, similar to the first two Prentice criteria. Second, it posits a no direct effect criterion for licensing correct inferences on the clinical treatment effect in the new trial, which is a conditional mean version of Prentice's 'full mediation' criterion. Moreover, our approach departs from the Prentice criteria by applying both to settings where the studied surrogate varies in both treatment arms and to settings where it only varies in the active treatment arm, which is important given the many applications where the latter scenario attains (Gilbert and Hudgens, 2008, Gilbert et al., 2015), whereas in contrast the Prentice approach only applies to the former scenario, e.g., Chan et al. (2002) and Gilbert, Qin, and Self (2008). This is important because the latter scenario is quite common, for example in trials where the candidate surrogate is a biomarker response endpoint that is structurally negative/zero for all placebo/control group recipients (Gilbert and Hudgens, 2008).

The optimal surrogate approach is related to the meta-analysis framework by addressing the common objective of inference on the clinical treatment effect in a future study without collecting the clinical outcome in that study (Gail et al., 2000). However, it tackles this objective based on a single (or few) efficacy trial plus transportability assumptions that are different from the 'extrapolation' assumptions needed via meta-analysis– meta-analysis bases inference on the association of trial-level surrogate and clinical treatment effects estimated from a series of trials and the assumption that the series of trials forms a correct basis for extrapolating the clinical treatment effect to the new setting not included in the series. Finally, the optimal surrogate approach breaks new ground by treating the surrogate endpoint problem as a supervised statistical learning problem. While historically methods evaluate a pre-selected univariable or low-dimensional vector candidate surrogate, the optimal surrogate approach allows all collected baseline and intermediate response data to potentially contribute to the optimal surrogate, based on unbiased machine learning, and does not require parametric modeling assumptions.

Web Appendix C: Proof of Theorem 3

The first statement of Theorem 3 is established by Theorem 2, so that we only need to show the last statement. By assumption $a \rightarrow E_{P_0}(Y \mid W = w, A = a, S = s)$ is constant in a for P_0-a.e (w, s). Thus, $a \rightarrow E_{P_0}(Y_a \mid W = w, S_a = s)$ is constant in a for P_0-a.e. (w, s), but since $E_P(Y^* \mid W^* = w, A^* = a, S^* = s) = E_{P_0}(Y \mid W = w, A = a, S = s)$ for all P-a.e. (w, s) (since the support of (W^*, S^*) is contained in the support of (W, S)), we also have that $a \rightarrow E_P(Y^* \mid W^* = w, A^* = a, S^* = s)$ is constant in a, and, by randomization of A^*, the latter is equivalent to $a \rightarrow E_P(Y^*_a \mid W^* = w, S^*_a = s)$ is constant in a. □

Web Appendix D: Super-learning of the P_0-optimal surrogate

Estimation of the P_0-optimal surrogate is a standard prediction problem. That is, we estimate $E_0(Y \mid W, A, S)$ with a minimizer of the risk of a loss: $\psi_0 = \arg\min_\psi P_0 L(\psi)$, with $Pf \equiv \int f(o) dP(o)$. For example, one could use squared error loss $L(\psi)(O) = (Y - \psi(W, A, S))^2$. To construct an optimal estimator among any given class of candidate estimators, we use loss-based super-learning. The oracle inequality for the cross-validation selector guarantees that the estimator is asymptotically at least as good as any candidate in the set of candidate estimators (van der Laan, Polley, and Hubbard, 2007; van der Laan and Rose, 2011).

Let $\hat{\Psi}_j : \mathcal{M}_{NP} \rightarrow \Psi(\mathcal{M})$ be a candidate estimator that maps an empirical distribution of $(O_1, \ldots, O_n)$ (i.e., an element of the nonparametric model $\mathcal{M}_{NP}$ of probability distributions) into the parameter space $\Psi(\mathcal{M}) = \{\Psi(P) : P \in \mathcal{M}\}$, $j = 1, \ldots, J$. This library of candidate estimators could include a variety of parametric model based estimators as well as a variety of machine learning algorithms, possibly coupled with different dimension reduction strategies, and possibly indexed by a variety of tuning parameters.

Let $B_n \in \{0,1\}^n$ be a random split of the sample into a training sample $\{i : B_n(i) = 0\}$ and validation sample $\{i : B_n(i) = 1\}$. For example, if we use V-fold cross-validation defined by a partitioning of the sample in V equal size groups, then B_n has V possible realizations, each occurring with probability $1/V$, and each split corresponds with setting the components of B_n in one of the V-folds equal to 1 and setting the other components equal to 0. Let P^0_{n,B_n} and P^1_{n,B_n} be the empirical distributions of the training and validation sample corresponding with split-vector B_n, respectively. The cross-validated risk of the j-th candidate estimator is then defined as $E_{B_n} P^1_{n,B_n} L(\hat{\Psi}_j(P^0_{n,B_n}))$, where $L(\cdot)$ should be chosen as squared error loss to be consistent with our proposed criterion (1) from the main article for the optimal surrogate.

One could now define the cross-validation selector

$$J_n = \arg\min_j E_{B_n} P^1_{n,B_n} L(\hat{\Psi}_j(P^0_{n,B_n}))$$

as the selector of the winner, and the corresponding discrete super-learner is then defined as $\hat{\Psi}(P_n) = \hat{\Psi}_{J_n}(P_n)$. One could also propose a parametric family $\{f_\alpha : \alpha\}$ of functions from $\mathrm{I\!R}^J$ to the real line that represents a family of combinations of all the J estimators:

$$\hat{\Psi}_\alpha(P_n) = f_\alpha(\hat{\Psi}_j(P_n) : j = 1, \cdots, J),$$

and where α represents a multivariate vector. For example, one might define $\hat{\Psi}_\alpha = \sum_{j=1}^J \alpha_j \hat{\Psi}_j$ as a weighted linear combination of the candidate estimators, where the weights α_j are restricted to be non-negative and sum to 1. One now defines the cross-validation selector for this continuous family of candidate estimators $\{\hat{\Psi}_\alpha : \alpha\}$ accordingly:

$$\alpha_n = \arg\min_\alpha E_{B_n} P^1_{n,B_n} L(\hat{\Psi}_\alpha(P^0_{n,B_n})).$$

The super-learner is then defined as $\hat{\Psi}(P_n) = \hat{\Psi}_{\alpha_n}(P_n)$. By the oracle inequality for the cross-validation selector, the super-learner is asymptotically equivalent with the oracle selected estimator, as long as the realistic assumption holds that none of the candidate estimators is a correctly specified parametric model (van der Laan, Polley, and Hubbard, 2007).

In addition, we can evaluate the super-learner by its cross-validated risk, using a cross-validation scheme S_n (e.g., using V-fold cross-validation again as in the super-learner):

$$\text{CV-RISK} = E_{S_n} P^1_{n,S_n} L(\hat{\Psi}(P^0_{n,S_n})), \tag{1}$$

which involves rerunning the super-learner on learning samples $\{i : S_n(i) = 0\}$ and evaluating it on test samples $\{i : S_n(i) = 1\}$, and averaging the performance across the different splits.

This represents an estimator of the true conditional risk

$$E_{S_n} R(\hat{\Psi}(P^0_{n,S_n}) \mid P_0) \equiv E_{S_n} P_0 L(\hat{\Psi}(P^0_{n,S_n})),$$

and one can also construct a Wald-type 95% confidence interval for the latter parameter $E_{S_n} R(\hat{\Psi}(P^0_{n,S_n}) \mid P_0)$ given by CV-RISK $\pm 1.96\sigma_n/\sqrt{n}$, where $\sigma^2_n = E_{S_n} P^1_{n,S_n} \left\{ L(\hat{\Psi}(P^0_{n,S_n})) - E_{S_n} P^1_{n,S_n} L(\hat{\Psi}(P^0_{n,S_n})) \right\}^2$. The theory behind the asymptotic correctness of this data adaptive confidence interval is given in van der Laan, Hubbard, and Pajouh (2013). A super-learner can be built and fitted with the R package *superlearner* available at CRAN.

One can also define a cross-validated R^2:

$$\text{CV-R}^2 = 1 - \text{CV-RISK}/E_{S_n} P^1_{n,S_n} L(\hat{\Psi}^0(P^0_{n,S_n})), \tag{2}$$

where $\hat{\Psi}^0(P_n) = \int y dP_n(y)$ is the empirical mean of the Y_i-values. This provides a universal measure of the strength of the estimated surrogate $\hat{\Psi}$, allowing us to compare different candidate surrogate estimators across studies and within a study. For example, one might construct a super-learner $\hat{\Psi}_\delta$ based on δ-specific subsets (W_δ, S_δ) of the complete (W, S), where δ is a measure of the complexity of the resulting surrogate as a function of (W, S).

One could now plot CV-R^2 of $\hat{\Psi}_\delta$ against δ for a sequence of δ-values, and the user can decide on a choice of δ taking into account both complexity and strength of the surrogate. This analysis is practically important given that all of the variables (W_δ, S_δ) used in the estimated optimal surrogate need to be collected in a future trial to use the estimated optimal surrogate in that trial; in practice some variable sets may be selected based on their high likelihood of being collected.

Web Appendix E: Additional analyses of the CYD14 and CYD15 dengue vaccine efficacy trial data sets

Supplemental Figures 1–4 display Month 13 PRNT$_{50}$ and Microneutralization Version 2 (MNv2) neutralization titers to the four dengue serotypes in the CYD-TDV vaccine (S) by protocol-specified age and sex covariate categories (W) and the treatment category A (where $A = 1$ for vaccine and $A = 0$ for placebo). For both CYD14 and CYD15 it is apparent that older children tend to have higher neutralization titers to all 4 serotypes than do younger children, based on both assays. Additionally, for both studies, there is an observable difference in the distributions of Month 13 neutralization titers between the vaccine and placebo groups, with higher Month 13 titers seen on average for the vaccine group. This is expected given that one of the designed purposes of vaccination is to generate neutralizing antibody responses.

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

[Figure 4 about here.]

Web Appendix F: Diagnostic checking of the conditions of Theorem 2 for the dengue vaccine efficacy trials application

In Theorem 2 we established that the estimated optimal surrogate is still a valid surrogate in a new study under the following four conditions:

(1) (Randomization:) A^* remains randomized, conditional on W^*,

(2) (Equal Conditional Means:) The conditional mean of Y given (W, A, S) is the same as the conditional mean of Y^* given (W^*, A^*, S^*),

(3) (Contained Support:) The support of (W^*, A^*, S^*) is contained in the support of (W, A, S), and

(4) (Positivity:) $P(A^* = a|W^*) > 0$ a.e. for $a \in \{0, 1\}$.

We check the four conditions treating CYD14 as the original study and CYD15 as the new study.

Condition 1 (Randomization) is met by the fact that both CYD14 and CYD15 randomized study participants to treatment (vaccine versus placebo).

Condition 2 (Equal Conditional Means) is explored in Supplemental Figures 5 and 6, which display the difference between the targeted estimated optimal surrogate $\psi_n^{\#_{14}}(W, A, S)$ built using CYD14 data and the targeted estimated optimal surrogate $\psi_{n*}^{\#_{15}}(W^*, A^*, S^*)$ built using CYD15 data. For each fixed level of the observations in CYD15, denoted by $(W^*, A^*, S^*) = (w, a, s)$, $\psi_n^{\#_{14}}(w, a, s) = \widehat{E}[Y|W = w, A = a, S = s]$ was calculated and subtracted from $\psi_{n*}^{\#_{15}}(w, a, s) = \widehat{E}[Y^*|W^* = w, A^* = a, S^* = s]$. Should the conditional mean of Y given (W, A, S) be identical to the conditional mean of Y^* given (W^*, A^*, S^*), these differences, $d(w, a, s) \equiv \widehat{E}[Y^*|W^* = w, A^* = a, S^* = s] - \widehat{E}[Y|W = w, A = a, S = s]$ should be close to zero for all observations in the CYD15 data set. As can be seen in Supplemental Figures 5 and 6, these differences generally cluster around zero and are centered around zero. The values are plotted by categories of age, sex, and treatment group, and are plotted against the PRNT_{50} Month 13 (Supplemental Figure 5) and Microneutralization Version 2 (MNv2) Month 13 neutralization titer values (Supplemental Figure 5). For both PRNT_{50} and MNv2 titer values, the older age category of 12–14 years for vaccinated individuals has a smaller spread of the differences $d(w, a, s)$ around 0, which was verified by the standard deviations for each category (results not shown). No other clear differences between covariate categories or between neutralization titer values are apparent.

[Figure 5 about here.]

[Figure 6 about here.]

Condition 3 (Contained Support) requires the support of (W^*, A^*, S^*) to be contained in the support of (W, A, S). The contained-support assumption holds for the age and sex covariates, because CYD14 included 2–14 year-old children and the analysis of CYD15 was restricted to 9–14 year-old children, and both studies included large numbers of male and female participants including sizable subgroups at each numeric age level. However the contained-support assumption appeared to be somewhat violated for the neutralization titer variables (Month 13 PRNT_{50} and MNv2 readouts to the four dengue serotypes). Although all titer variables had the same minimum values, and the PRNT_{50} and MNv2 serotype-specific neutralization titers were also relatively similar between the two studies, maximum titer values were slightly different in CYD15 than in CYD14. The maximum PRNT_{50} neutralization titer values for serotype 1 and for serotype 3 were 14% higher and 18% higher for CYD15 than for CYD14, respectively, but all other maximum PRNT_{50} titer values were smaller for CYD15 when compared to CYD14. For MNv2 titers, the serotype 1 maximum titer value was 9% greater, the serotype 3 maximum titer value was 17% greater, and the serotype 4 maximum titer value was 1% greater for CYD15 when compared to the maximum titers for CYD14. In sum, there are minor violations of the contained-support assumption that are expected to have a minor-to-moderate influence on the results.

Condition 4 (Positivity) requires that $P(A^* = a|W^*) > 0$ a.e. for $a \in \{0,1\}$. The baseline covariates consist of discrete age categories crossed with gender crossed with the four continuous variables the estimated serotype frequencies of placebo recipients in the participant's country crossed with the baseline titer variables. CYD15 was a large study with over 20,000 participants, providing ample data to check positivity by comparing the distribution of W^* between the treatment groups (results not shown). The age × gender × serotype frequency distributions highly overlapped across the treatment groups, supporting positivity. Moreover the baseline titer distributions in vaccine and placebo recipients had similar ranges (Supplemental Figures 1–4), consistent with the positivity assumption.

Web Appendix G: Two simulation studies of the proposed methodology

We conduct two simulation studies to illustrate that the targeted estimated optimal surrogate will generally provide unbiased estimation of $\theta_0 = E_0(Y_1 - Y_0)$ in the original trial, for any distribution of (W, A, S, Y), whereas in contrast a proportion of treatment effect explained based approach that has been popular in practice does not. We then evaluate how well the surrogate built from the original study can be used to estimate θ_0 in a new study that only measures (W, A, S), when the Equal Conditional Means assumption fails.

Data generating distribution

Building upon an example in VanderWeele (2013), we simulate data illustrating the surrogate paradox. The data set is comprised of an outcome Y, a randomized treatment $A \in \{0,1\}$, and 10 candidate surrogates S^k, each with three levels $S^k \in \{0,1,2\}$ for $k = 1, \ldots, 10$. For each k the joint potential outcomes S^k_a for $a \in \{0,1\}$ have the following distribution: $P(S^k_1 = 0, S^k_0 = 0) = P(S^k_1 = 1, S^k_0 = 1) = P(S^k_1 = 2, S^k_0 = 2) = 0.1$, $P(S^k_1 = 1, S^k_0 = 0) = 0.5$, and $P(S^k_1 = 1, S^k_0 = 2) = 0.2$. The outcome $Y = \sum_{k=1}^{3} \left[0.1 * k * I(S^k = 1) + I(S^k = 2)\right] + \epsilon_Y$, where $\epsilon_Y \sim N(0, 0.1^2)$. In this setting $\theta_0 = E_0(Y_1 - Y_0) = -0.18$, whereas $E_0(S^k_1 - S^k_0) = 0.3$ for each k, such that the surrogate paradox occurs for each k.

Methods for estimating θ_0 based on a surrogate

The estimated optimal surrogate $\psi_n^{\#}(A, S)$ and the resulting estimate $\theta_{\psi_n^{\#}}^{TMLE}$ of θ_0 are calculated as for the example. We compare performance of $\theta_{\psi_n^{\#}}^{TMLE}$ to an alternative approach that estimates the Proportion of the Treatment Effect Captured (PCS) (Kobayashi and Kuroki, 2014) by each of the ten candidate surrogate endpoints to select the best single surrogate variable as the one that maximizes the estimated PCS, which we refer to as S^{PCSopt}. Specifically, for each of 100 bootstrapped data sets, the index k maximizing the estimated PCS in a linear regression model of Y on $I(S^k = 1)$ and $I(S^k = 2)$ was selected, and S^{PCSopt} was taken to be the S^k most frequently selected. Then θ_0 was estimated by θ_n^{PCSopt} defined as the difference in average predicted Y values for group $a = 1$ minus $a = 0$ in the fitted model $\widehat{E}_0(Y|S_{opt}^{\text{PCS}} = s, A = a) = \widehat{\beta}_0 + \widehat{\beta}_1 * I(s = 1) + \widehat{\beta}_2 * I(s = 2)$. Since a perfect surrogate captures all of the effect of the treatment A (indicated by PCS=1 in the proportion-of-treatment-effect explained paradigm), A was not included in the model. The true PCS values are 0.87, 0.2, and 0.002 for the first three S^k's that are predictive of Y.

Simulation 1: Comparison for estimating θ_0 in the original trial

For each of 200 hundred simulated data sets each with 2000 subjects, θ_0 was estimated based on the SL-TMLE surrogate and the PCS-selected surrogate as described above. Supplemental Figure 8(a) shows the concordance of the surrogate-based estimates of θ_0 and the gold-standard estimates based on the known clinical outcomes, $\tilde{\theta}_n^{TMLE} = \widehat{E}_0(Y_1) - \widehat{E}_0(Y_0)$, where the $\widehat{E}_0(Y_a)$'s are simply sample averages because no baseline covariates W are considered. The targeted estimated optimal surrogate-based estimates $\theta_{\psi_n^{\#}}^{TMLE}$ much more closely align with the direct Y-based estimates than those based on θ_n^{PCSopt}, with average $\theta_{\psi_n^{\#}}^{TMLE}$ of -0.18 and average θ_n^{PCSopt} of 0.02, compared to the true value $\theta_0 = -0.18$. The surrogate paradox defined by positive θ_n^{PCSopt} occurred for 191 (96%) of 200 repetitions, whereas it never occurred based on $\theta_{\psi_n^{\#}}^{TMLE}$. The PCS-based approach fails because it is not able to capture the 3-variable relationship from the data generating distribution, with CV-R^2 of -0.01 between the S^{PCSopt}-estimated $\widehat{Y}$ values and the Y values, compared to CV-R^2 of 0.98 from the estimated optimal surrogate.

Simulation 2: Comparison for estimating θ_P^* in a second trial

Our second simulation generates 200 pairs of data sets (D1, D2) with D1 generated as for Simulation 1 (the original trial) and D2 under a new data generating distribution where Y^* also depends on the fourth candidate surrogate: $Y^* = \sum_{i=1}^4 \left[0.1 * k * I(S^{*k} = 1) + I(S^{*k} = 2)\right] + \epsilon_{Y^*}$, where $\epsilon_{Y^*} \sim N(0, 0.1^2)$. The surrogates $\psi_n^{\#}(A, S)$ and $S^{\text{PCSopt}}(A, S)$ are calculated from D1 as in Simulation 1. Then, based on the (A^*, S^*) values in the paired data set D2, surrogate-based estimates of $\theta_{\psi_n^{\#}}^{TMLE}(P) = \theta_{\psi_n^{\#}}^{TMLE,1}(P) - \theta_{\psi_n^{\#}}^{TMLE,0}(P)$ are calculated as in Section 6 and $\theta_n^{\text{PCSopt}}(P) = (1/n_1^*) \sum_{i=1}^{n*} A_i^* \widehat{E}[Y|S_i^{*\text{PCSopt}}, A_i^* = 1] - (1/n_0^*) \sum_{i=1}^{n*} (1 - A_i^*) \widehat{E}[Y|S_i^{*\text{PCSopt}}, A_i^* = 0]$. The D2 data set is chosen such that the Equal Conditional Means assumptions is violated, as depicted in Supplementary Figure 7, which shows that $E_P[Y_a^*|S_a^{*4} = s] - E_{P_0}[Y_a|S_a^4 = s]$ varies widely in s for each $a \in \{0, 1\}$.

[Figure 7 about here.]

Supplemental Figure 8(b) displays the $\theta_{\psi_n^{\#}}^{TMLE}(P)$ and $\theta^{\text{PCSopt}}(P)$ estimates versus the gold-standard estimates $\tilde{\theta}_{n*}^{TMLE}$ based on the actual clinical outcomes Y^*. Both approaches demonstrate some bias for estimating θ_P^* (dotted line); however $\theta_{\psi_n^{\#}}^{TMLE}(P)$ does much better at estimating the effect in the correct direction (negative), while $\theta_n^{\text{PCSopt}}(P)$ estimates the effect near zero (the true treatment effect θ_P^* is -0.10, compared to an average estimate $\theta_{\psi_n^{\#}}^{TMLE}(P)$ of -0.18 and an average $\theta_n^{\text{PCSopt}}(P)$ of 0.02). Of the 200 simulation runs, 95% of the PCS-based estimates exhibit the surrogate paradox, compared to 0% for the SL-TMLE method. Therefore, in addition to demonstrating that the Equal Conditional Means assumption is necessary for valid inference on θ_P^* in a new setting, this simulation illustrates that when Equal Conditional Means is majorly violated, the SL-TMLE approach can still preserve some accuracy in bridging the clinical treatment effect to a new setting.

[Figure 8 about here.]

Web Appendix H: Dummy CYD14 and CYD15 data sets and R code for producing the results of Section 6 for the dummy data sets

Because it is not possible to share the real CYD14 and CYD15 data sets, we provide the code for analyzing dummy versions of the CYD14 and CYD15 data sets, that have the same structure and variables as the real data sets. This code produces the same outputs that were produced for the real data sets (in particular, Figures 1 and 2 and Tables 2 and 3 of the main article are re-produced using the dummy data, and additionally Figures S1–S6 are re-produced using the dummy data). In this appendix, we provide the code used to analyze these dummy data sets, and the resulting tables and figures. This allows readers to re-produce the analysis and check that the same answers are obtained.

Figures 13–18 provide the results using the CYD14 dummy and CYD15 dummy data sets.

[Figure 9 about here.]

[Figure 10 about here.]

[Figure 11 about here.]

[Figure 12 about here.]

[Figure 13 about here.]

[Figure 14 about here.]

[Figure 15 about here.]

[Figure 16 about here.]

[Figure 17 about here.]

[Figure 18 about here.]

The R code consists of a single file “BiometricsPriceGilbertVanDerLaanDummyDataDengueExampleCode.R”

This file is available at the second author’s website (and its contents are printed below):

`http://faculty.washington.edu/peterg/programs.html?`

BiometricsPriceGilbertVanDerLaanDummyDataDengueExampleCode.R

```
##############################################
# "Estimated Optimal Surrogate Example: CYD14 and CYD15 Dengue Vaccine Efficacy Trials"
# author: "Brenda Price"
# In Price BL, Gilbert PB, van der Laan MJ. R code for implementing the Application for
# dengue vaccine efficacy trials (on dummy data sets) in "Estimation of the Optimal
# Surrogate Based on a Randomized Trial" (2017)
# Dec 8, 2017
# R script used to produce results for the example reported in the paper
# USING DUMMY DATA--data sets were simluated to have similar structure as the actual data
#############################################



#######################
# Set working directory
# Enter the path for the working directory
setwd("...")

#######################
# install any of the packages needed
wants <- c("SuperLearner","tmle","xtable","Hmisc","gam","polspline",
           "class","survival","ggplot2","glmnet","gridExtra","pROC","cvAUC",
           "arm","lme4","stats","plyr")
has   <- wants %in% rownames(installed.packages())
if(any(!has)) install.packages(wants[!has])
library(SuperLearner)
library(tmle)
library(ggplot2)
options(warn=-1)
```

```
#######################
# R Screening Functions not included in the SuperLearner package from CRAN
# Each function returns a list of variables that meet specific criteria described below

#### Screen function for colinear variables within X ####
## screen.corX.x: Do not allow any pairs of quantitative variables (PRNT and/or MNv2) with R^2 > 0.x
screen.corX.6 <-function(Y, X, family, obsWeights, id, method = "spearman", minPvalue = 0.1, maxCorr =0.6,
                         minscreen = 2, ...)
{
  Colinear <- corX <- cor(X)
  Colinear <- (corX)^2>=maxCorr # dealing with r^2
  ListPairs <- matrix(NA,nrow=nrow(Colinear)^2,ncol=2)
  index <- 1
  for (i in 1:(ncol(Colinear)-1))
  {
    for(j in (i+1):ncol(Colinear))
    {
      if(isTRUE(Colinear[i,j])){
        ListPairs[index,] <- matrix(c(rownames(Colinear)[i],colnames(Colinear)[j]),nrow=1)
        index <- index + 1
      }
    } #end j
  } #end i
  ListPairs <- ListPairs[is.na(ListPairs[,1])==FALSE,]  ## list of collinear pairs
  drop <- colnames(Colinear) %in% ListPairs[,2]  ## drop the second variable in each pair
  whichVariable <-drop==FALSE
  return(whichVariable)
}

screen.corX.7 <- function(... , maxCorr =0.7) {
  screen.corX.6(... , maxCorr = maxCorr )
}



#### Screen for Univariate logistic regression relationships.
##  Returns a list of the 2 or 3 most significant variables from univariate logistic regression
# 2 most significant variables
screen.univar.logistic.2<-function(Y, X, family, obsWeights, id, rank = 2, minPvalue=0.1, nvar=2, ...)
{
  listp <- apply(X, 2, function(x, Y, method) {
    ifelse(var(x) <= 0, 1, summary(glm(Y~x,family="binomial", weights=obsWeights))$coefficients[2,4])
  }, Y = Y)
  listp.rank <- rank(listp)
  whichVariable <- (listp.rank <= nvar)
  return(whichVariable)
}

# 3 most significant variables
screen.univar.logistic.3 <- function(... , nvar=3) {
  screen.univar.logistic.2(... , nvar=nvar )
}

#### Screening method for disallowing MNv2 serotype variables
screen.PRNT <- function(Y, X, family, obsWeights, id, ...) {
  # set all to True except variables with MNv2 in the name
  vars <- lapply(strsplit(names(X),".",fixed=T),function(x) ifelse(x[2]=="MNv2",FALSE,TRUE))
  vars <- ifelse(is.na(vars),TRUE,vars)
  vars <- unlist(vars)
  return(vars)
}

#### Screening method for disallowing PRNT serotype variables
screen.MNv2 <- function(Y, X, family, obsWeights, id, ...) {
  # set all to True except variables with PRNT in the name
  vars <- lapply(strsplit(names(X),".",fixed=T),function(x) ifelse(x[2]=="PRNT",FALSE,TRUE))
  vars <- ifelse(is.na(vars),TRUE,vars)
  vars <- unlist(vars)
  return(vars)
}

## To focus on parsiomious models, this screen is modified to use the lambda.1se
## (largest value of lambda such that error is within 1 standard error of the minimum.)
## instead of lambda.min
screen.glmnet <- function (Y, X, family, alpha = 1, minscreen = 2, nfolds = 10,
                           nlambda = 100, ...)
{
 # .SL.require("glmnet")
  if (!is.matrix(X)) {
    X <- model.matrix(~-1 + ., X)
  }
  fitCV <- glmnet::cv.glmnet(x = X, y = Y, lambda = NULL, type.measure = "deviance",
                             nfolds = nfolds, family = family$family, alpha = alpha,
                             nlambda = nlambda)
  whichVariable <- (as.numeric(coef(fitCV$glmnet.fit, s = fitCV$lambda.1se))[-1] !=
                      0)
  if (sum(whichVariable) < minscreen) {
    warning("fewer than minscreen variables passed the glmnet screen, increased lambda to allow minscreen variables")
    sumCoef <- apply(as.matrix(fitCV$glmnet.fit$beta), 2,
                     function(x) sum((x != 0)))
    newCut <- which.max(sumCoef >= minscreen)
    whichVariable <- (as.matrix(fitCV$glmnet.fit$beta)[,
                                                       newCut] != 0)
  }
  return(whichVariable)
}

### Re-write SL.step function to include handle case-control weights
## Standard SL.step included in SuperLearner doesn't incorporate sampling weights
SL.step <- function (Y, X, newX, family, obsWeights, direction = "both", trace = 0,
                     k = 2, ...)
{
  fit.glm <- glm(Y ~ ., data = X, family = family, weights = obsWeights)
  fit.step <- step(fit.glm, direction = direction, trace = trace,
                   k = k)
  pred <- predict(fit.step, newdata = newX, type = "response")
  fit <- list(object = fit.step)
  out <- list(pred = pred, fit = fit)
  class(out$fit) <- c("SL.step")
  return(out)
}


#######################
### read in the data ###
d14 <- read.table(".../simCYD14_112017.csv", header=TRUE, stringsAsFactors=FALSE,sep=",")
d15 <- read.table(".../simCYD15_112017.csv", header=TRUE, stringsAsFactors=FALSE,sep=",")

library(plyr)
d14<-rename(d14, c("M13.Sero1"="M13_PRNT_Sero1",
                   "M13.Sero2"="M13_PRNT_Sero2",
                   "M13.Sero3"="M13_PRNT_Sero3",
                   "M13.Sero4"="M13_PRNT_Sero4",
                   "M13.AUC"="M13_PRNT_SeroAverage",
                   "vcdstatus_m13"="VCD",
                   "M13mnv2.Sero1" = "M13_MNv2_Sero1",
                   "M13mnv2.Sero2" = "M13_MNv2_Sero2",
                   "M13mnv2.Sero3" = "M13_MNv2_Sero3",
                   "M13mnv2.Sero4" = "M13_MNv2_Sero4",
                  "M13mnv2.AUCMB" = "M13_MNv2_SeroAverage",
                   "bS1" = "BaselinePRNT_Sero1",
                   "bS2" = "BaselinePRNT_Sero2",
                   "bS3" = "BaselinePRNT_Sero3",
                   "bS4" = "BaselinePRNT_Sero4"))
d15<-rename(d15, c("M13.Sero1"="M13_PRNT_Sero1",
                   "M13.Sero2"="M13_PRNT_Sero2",
                   "M13.Sero3"="M13_PRNT_Sero3",
                   "M13.Sero4"="M13_PRNT_Sero4",
                   "M13.AUC"="M13_PRNT_SeroAverage",
                   "vcdstatus_m13"="VCD",
                   "M13mnv2.Sero1" = "M13_MNv2_Sero1",
                   "M13mnv2.Sero2" = "M13_MNv2_Sero2",
                   "M13mnv2.Sero3" = "M13_MNv2_Sero3",
                   "M13mnv2.Sero4" = "M13_MNv2_Sero4",
                   "M13mnv2.AUCMB" = "M13_MNv2_SeroAverage",
                   "bS1" = "BaselinePRNT_Sero1",
                   "bS2" = "BaselinePRNT_Sero2",
                   "bS3" = "BaselinePRNT_Sero3",
                   "bS4" = "BaselinePRNT_Sero4"))

# Remove observations with missing outcome data
d14 <- d14[is.na(d14$VCD)==FALSE,]
d15 <- d15[is.na(d15$VCD)==FALSE,]

### Create variable for the minimum of the 4 PRNT serotypes for each subject
d14$M13_PRNT_MinSeroTiter <- apply(cbind(d14$M13_PRNT_Sero1,
                                    d14$M13_PRNT_Sero2,
                                    d14$M13_PRNT_Sero3,
                                    d14$M13_PRNT_Sero4),1,min)
### Create variable for the minimum of the 4 PRNT serotypes for each subject
d14$M13_PRNT_MaxSeroTiter <- apply(cbind(d14$M13_PRNT_Sero1,
                                    d14$M13_PRNT_Sero2,
                                    d14$M13_PRNT_Sero3,
                                    d14$M13_PRNT_Sero4),1,max)
### Create variable for the minimum of the 4 PRNT serotypes for each subject
```

```
d15$M13_PRNT_MinSeroTiter <- apply(cbind(d15$M13_PRNT_Sero1,
                                    d15$M13_PRNT_Sero2,
                                    d15$M13_PRNT_Sero3,
                                    d15$M13_PRNT_Sero4),1,min)
### Create variable for the minimum of the 4 PRNT serotypes for each subject
d15$M13_PRNT_MaxSeroTiter <- apply(cbind(d15$M13_PRNT_Sero1,
                                    d15$M13_PRNT_Sero2,
                                    d15$M13_PRNT_Sero3,
                                    d15$M13_PRNT_Sero4),1,max)

# Average Titer for MNv2
d14$M13_MNv2_SeroAverage <- apply(cbind(d14$M13_MNv2_Sero1,d14$M13_MNv2_Sero2,
d14$M13_MNv2_Sero3,d14$M13_MNv2_Sero4),1,mean)
d15$M13_MNv2_SeroAverage <- apply(cbind(d15$M13_MNv2_Sero1,d15$M13_MNv2_Sero2,
d15$M13_MNv2_Sero3,d15$M13_MNv2_Sero4),1,mean)

### Create variable for the minimum of the 4 serotypes for each subject
d14$M13_MNv2_MinSeroTiter <- apply(cbind(d14$M13_MNv2_Sero1,
                                         d14$M13_MNv2_Sero2,
                                         d14$M13_MNv2_Sero3,
                                         d14$M13_MNv2_Sero4),1,min)
### Create variable for the minimum of the 4 serotypes for each subject
d14$M13_MNv2_MaxSeroTiter <- apply(cbind(d14$M13_MNv2_Sero1,
                                         d14$M13_MNv2_Sero2,
                                         d14$M13_MNv2_Sero3,
                                         d14$M13_MNv2_Sero4),1,max)
### Create variable for the minimum of the 4 serotypes for each subject
d15$M13_MNv2_MinSeroTiter <- apply(cbind(d15$M13_MNv2_Sero1,
                                         d15$M13_MNv2_Sero2,
                                         d15$M13_MNv2_Sero3,
                                         d15$M13_MNv2_Sero4),1,min)
### Create variable for the minimum of the 4 serotypes for each subject
d15$M13_MNv2_MaxSeroTiter <- apply(cbind(d15$M13_MNv2_Sero1,
                                         d15$M13_MNv2_Sero2,
                                         d15$M13_MNv2_Sero3,
                                         d15$M13_MNv2_Sero4),1,max)

## Restrict testing data set to age range of interest (9-14 years)
d15 <- d15[d15$AGEYRS<15,]

## calculate observation weights
```

```
CaseControlWeights <- function(d){
  dcase <- d[d$VCD==1,]
  dcontrol <- d[d$VCD==0,]
  # number of cases
  n_case <- nrow(d[d$VCD==1,])
  # number of controls
  n_control <- nrow(d[d$VCD==0,])
  # number of controls w/marker data at m13
  n_control_m13 <- nrow(dcontrol[(is.na(dcontrol$M13_MNv2_Sero4)==F |
                                    is.na(dcontrol$M13_MNv2_Sero3)==F|
                                    is.na(dcontrol$M13_MNv2_Sero2)==F|
                                    is.na(dcontrol$M13_MNv2_Sero1)==F) &
                                   (is.na(dcontrol$M13_PRNT_Sero4)==F |
                                      is.na(dcontrol$M13_PRNT_Sero3)==F|
                                      is.na(dcontrol$M13_PRNT_Sero2)==F|
                                      is.na(dcontrol$M13_PRNT_Sero1)==F),])
  # number of cases w/marker data at m13
  n_case_m13 <- nrow(dcase[(is.na(dcase$M13_MNv2_Sero4)==F |
                              is.na(dcase$M13_MNv2_Sero3)==F|
                              is.na(dcase$M13_MNv2_Sero2)==F|
                              is.na(dcase$M13_MNv2_Sero1)==F) &
                             (is.na(dcase$M13_PRNT_Sero4)==F |
                                is.na(dcase$M13_PRNT_Sero3)==F|
                                is.na(dcase$M13_PRNT_Sero2)==F|
                                is.na(dcase$M13_PRNT_Sero1)==F),]) #244
  p_case <- n_case_m13/n_case
  p_nocase <- n_control_m13/n_control
  weight_case <- 1/p_case
  weight_nocase <- 1/p_nocase
  final_weight <- ifelse(d$VCD==1, weight_case,weight_nocase)
  return(final_weight)
}

## case control weights for full datasets
d14$weight<-CaseControlWeights(d14)
d15$weight<-CaseControlWeights(d15)
###

############
d14$dset <- "CYD14"
d15$dset <- "CYD15"
```

```
################
## Serotype Frequency by Country
# create variables of the relative proportion of each type of Dengue (1-4) within each country

CountryIncidence <- function(dset){
  ### serotype distributions by country for placebo; these are the ones used for models ####
  t1<-table(dset[dset$VACC ==0,]$COUNTRY,dset[dset$VACC ==0,]$s1vcdstatus_m13)
  t2<-table(dset[dset$VACC ==0,]$COUNTRY,dset[dset$VACC ==0,]$s2vcdstatus_m13)
  t3<-table(dset[dset$VACC ==0,]$COUNTRY,dset[dset$VACC ==0,]$s3vcdstatus_m13)
  t4<-table(dset[dset$VACC ==0,]$COUNTRY,dset[dset$VACC ==0,]$s4vcdstatus_m13)
  # calculate the proportion by
  country.rates <- cbind(t1[,2]/t1[,1],t2[,2]/t2[,1],t3[,2]/t3[,1],t4[,2]/t4[,1])
  colnames(country.rates)<- c("Sero1","Sero2","Sero3","Sero4")
  country.incidence <- cbind(t1[,2],t2[,2],t3[,2],t4[,2])
  country.incidence.sum<-apply(country.incidence,1,sum)
  rates.per.country <- country.incidence/country.incidence.sum
  ## percentage of all infections in country that were each serotype
  colnames(rates.per.country)<- c("Sero1","Sero2","Sero3","Sero4")

  ### assign rates.per.country variables to records ####
  c<-levels(as.factor(dset$COUNTRY))
  dset$Sero1.rate <- dset$Sero2.rate <- dset$Sero3.rate <- dset$Sero4.rate <- NA
  dset$Sero1.rate <- ifelse(dset$COUNTRY == c[1], rates.per.country[1,1],dset$Sero1.rate)
  dset$Sero1.rate <- ifelse(dset$COUNTRY == c[2], rates.per.country[2,1],dset$Sero1.rate)
  dset$Sero1.rate <- ifelse(dset$COUNTRY == c[3], rates.per.country[3,1],dset$Sero1.rate)
  dset$Sero1.rate <- ifelse(dset$COUNTRY == c[4], rates.per.country[4,1],dset$Sero1.rate)
  dset$Sero1.rate <- ifelse(dset$COUNTRY == c[5], rates.per.country[5,1],dset$Sero1.rate)

  dset$Sero2.rate <- ifelse(dset$COUNTRY == c[1], rates.per.country[1,2],dset$Sero2.rate)
  dset$Sero2.rate <- ifelse(dset$COUNTRY == c[2], rates.per.country[2,2],dset$Sero2.rate)
  dset$Sero2.rate <- ifelse(dset$COUNTRY == c[3], rates.per.country[3,2],dset$Sero2.rate)
  dset$Sero2.rate <- ifelse(dset$COUNTRY == c[4], rates.per.country[4,2],dset$Sero2.rate)
  dset$Sero2.rate <- ifelse(dset$COUNTRY == c[5], rates.per.country[5,2],dset$Sero2.rate)

  dset$Sero3.rate <- ifelse(dset$COUNTRY == c[1], rates.per.country[1,3],dset$Sero3.rate)
  dset$Sero3.rate <- ifelse(dset$COUNTRY == c[2], rates.per.country[2,3],dset$Sero3.rate)
  dset$Sero3.rate <- ifelse(dset$COUNTRY == c[3], rates.per.country[3,3],dset$Sero3.rate)
  dset$Sero3.rate <- ifelse(dset$COUNTRY == c[4], rates.per.country[4,3],dset$Sero3.rate)
  dset$Sero3.rate <- ifelse(dset$COUNTRY == c[5], rates.per.country[5,3],dset$Sero3.rate)
  dset$Sero4.rate <- ifelse(dset$COUNTRY == c[1], rates.per.country[1,4],dset$Sero4.rate)
  dset$Sero4.rate <- ifelse(dset$COUNTRY == c[2], rates.per.country[2,4],dset$Sero4.rate)
  dset$Sero4.rate <- ifelse(dset$COUNTRY == c[3], rates.per.country[3,4],dset$Sero4.rate)
  dset$Sero4.rate <- ifelse(dset$COUNTRY == c[4], rates.per.country[4,4],dset$Sero4.rate)
  dset$Sero4.rate <- ifelse(dset$COUNTRY == c[5], rates.per.country[5,4],dset$Sero4.rate)
  return(dset)
}

d14.ci <- CountryIncidence(dset=d14)
d15.ci <- CountryIncidence(dset=d15)

# combine datasets for additional data processing before separating into old and new studies
d <-rbind(d14.ci,d15.ci)

## "Only participants with Month 13 titer data on both neutralization assay types are included in the analysis"
d <- d[is.na(d$M13_PRNT_Sero1)==FALSE &
         is.na(d$M13_PRNT_Sero2)==FALSE &
         is.na(d$M13_PRNT_Sero3)==FALSE &
         is.na(d$M13_PRNT_Sero4)==FALSE &
         is.na(d$VCD)==FALSE &
         is.na(d$M13_MNv2_Sero1)==FALSE &
         is.na(d$M13_MNv2_Sero2)==FALSE &
         is.na(d$M13_MNv2_Sero3)==FALSE &
         is.na(d$M13_MNv2_Sero4)==FALSE
       ,]

# create age indicator variables
d$AGE.2.8 <- ifelse(d$AGEYRS<=8,1,0)
d$AGE.9.11 <- ifelse(d$AGEYRS>8 & d$AGEYRS<=11,1,0)
d$AGE.12.14 <- ifelse(d$AGEYRS>11,1,0)

## retain variables with the continuous information for plots
d$M13_MNv2_Sero1c <- d$M13_MNv2_Sero1
d$M13_MNv2_Sero2c <- d$M13_MNv2_Sero2
d$M13_MNv2_Sero3c <- d$M13_MNv2_Sero3
d$M13_MNv2_Sero4c <- d$M13_MNv2_Sero4
d$M13_PRNT_Sero1c <- d$M13_PRNT_Sero1
d$M13_PRNT_Sero2c <- d$M13_PRNT_Sero2
d$M13_PRNT_Sero3c <- d$M13_PRNT_Sero3
d$M13_PRNT_Sero4c <- d$M13_PRNT_Sero4
```

```
# code individual serotypes as binary pos/neg
CodeBinarySero <- function(x){
  b <- ifelse(x <=log10(5)+0.001,0,1) # "+0.001"" is to adjust for possible rounding in the csv file
  b <- ifelse(is.na(x),NA,b)
  b
}

d$M13_MNv2_Sero1 <-CodeBinarySero(d$M13_MNv2_Sero1)
d$M13_MNv2_Sero2 <-CodeBinarySero(d$M13_MNv2_Sero2)
d$M13_MNv2_Sero3 <-CodeBinarySero(d$M13_MNv2_Sero3)
d$M13_MNv2_Sero4 <-CodeBinarySero(d$M13_MNv2_Sero4)
d$M13_PRNT_Sero1 <-CodeBinarySero(d$M13_PRNT_Sero1)
d$M13_PRNT_Sero2 <-CodeBinarySero(d$M13_PRNT_Sero2)
d$M13_PRNT_Sero3 <-CodeBinarySero(d$M13_PRNT_Sero3)
d$M13_PRNT_Sero4 <-CodeBinarySero(d$M13_PRNT_Sero4)

# separate into old and new studies (CYD14 and CYD15)
d14 <- d[d$dset== "CYD14",]
d15 <- d[d$dset== "CYD15",]

#################
## SuperLearner Dummy Data Matrices
Y14 <- as.numeric(d14$VCD)
X14 <- data.frame(d14$VACC, #1
                  d14$AGE.9.11,  #2
                  d14$AGE.12.14, #3
                  d14$MALE, #4
                  d14$M13_PRNT_Sero1, d14$M13_PRNT_Sero2,
                  d14$M13_PRNT_Sero3, d14$M13_PRNT_Sero4, # 5-8
                  d14$M13_PRNT_SeroAverage, # 9
                  d14$M13_PRNT_MinSeroTiter,d14$M13_PRNT_MaxSeroTiter,# 10,11
                  d14$M13_MNv2_Sero1, d14$M13_MNv2_Sero2,
                  d14$M13_MNv2_Sero3, d14$M13_MNv2_Sero4, # 12-15
                  d14$M13_MNv2_SeroAverage, # 16
                  d14$M13_MNv2_MinSeroTiter,d14$M13_MNv2_MaxSeroTiter,# 17,18
                  d14$weight, # 19: weight
                  d14$Sero1.rate,d14$Sero2.rate,d14$Sero3.rate,d14$Sero4.rate
)
names(X14) <- cbind("VACC","AGE.9.11","AGE.12.14","MALE",
                    "M13.PRNT.S1","M13.PRNT.S2","M13.PRNT.S3","M13.PRNT.S4",
                    "M13.PRNT.Ave","M13.PRNT.Min","M13.PRNT.Max",
                     "M13.MNv2.S1","M13.MNv2.S2","M13.MNv2.S3","M13.MNv2.S4",
                     "M13.MNv2.Ave","M13.MNv2.Min","M13.MNv2.Max", # 19: weight
                     "IPCWeight",
                     "Sero1.frequency","Sero2.frequency","Sero3.frequency","Sero4.frequency"
)


Y14_vaccine <- Y14[X14$VACC ==1]
Y14_placebo <- Y14[X14$VACC ==0]
X14_vaccine <- X14[X14$VACC ==1,]
X14_placebo <- X14[X14$VACC ==0,]

## Demo + PRNT + MNv2 + SeroRate
X14_vaccine_subset <- X14_vaccine[,c(2:18,21:23)]
X14_placebo_subset <- X14_placebo[,c(2:18,21:23)]

## weight vector IPCW weights calculated above
weight14.v <- X14_vaccine[,19]
weight14.p <- X14_placebo[,19]

#################
## SuperLearner Dummy Data Matrices
Y15 <- as.numeric(d15$VCD)
X15 <- data.frame(d15$VACC, #1
                  d15$AGE.9.11,  #2
                  d15$AGE.12.14, #3
                  d15$MALE, #4
                  d15$M13_PRNT_Sero1, d15$M13_PRNT_Sero2, d15$M13_PRNT_Sero3,
                  d15$M13_PRNT_Sero4, # 5-8
                  d15$M13_PRNT_SeroAverage, # 9
                  d15$M13_PRNT_MinSeroTiter,d15$M13_PRNT_MaxSeroTiter,# 10,11
                  d15$M13_MNv2_Sero1, d15$M13_MNv2_Sero2, d15$M13_MNv2_Sero3,
                  d15$M13_MNv2_Sero4, # 12-15
                  d15$M13_MNv2_SeroAverage, # 16
                  d15$M13_MNv2_MinSeroTiter,d15$M13_MNv2_MaxSeroTiter,# 17,18
                  d15$weight, # 19: weight
                  d15$Sero1.rate,d15$Sero2.rate,d15$Sero3.rate,d15$Sero4.rate

)
names(X15) <- cbind("VACC","AGE.9.11","AGE.12.14","MALE",
                     "M13.PRNT.S1","M13.PRNT.S2","M13.PRNT.S3","M13.PRNT.S4",
                    "M13.PRNT.Ave","M13.PRNT.Min","M13.PRNT.Max",
                    "M13.MNv2.S1","M13.MNv2.S2","M13.MNv2.S3","M13.MNv2.S4",
                    "M13.MNv2.Ave","M13.MNv2.Min","M13.MNv2.Max", # 19: weight
                    "IPCWeight",
                    "Sero1.frequency","Sero2.frequency","Sero3.frequency","Sero4.frequency"
)

Y15_vaccine <- Y15[X15$VACC ==1]
Y15_placebo <- Y15[X15$VACC ==0]
X15_vaccine <- X15[X15$VACC ==1,]
X15_placebo <- X15[X15$VACC ==0,]

## Demo + PRNT + MNv2 + SeroRate
X15_vaccine_subset <- X15_vaccine[,c(2:18,21:23)]
X15_placebo_subset <- X15_placebo[,c(2:18,21:23)]

## weight vector IPCW weights calculated above
weight15.v <- X15_vaccine[,19]
weight15.p <-  X15_placebo[,19]

#######################
## SuperLearner on Dummy CYD14 data
# Stratify V-fold cross-validation so that all validation samples have roughly the same number of events
.cvFolds <- function(Y, V){ #Create CV folds (stratify by outcome) -- code from cvAUC documentation
  Y0 <- split(sample(which(Y==0)), rep(1:V, length=length(which(Y==0))))
  Y1 <- split(sample(which(Y==1)), rep(1:V, length=length(which(Y==1))))
  folds <- vector("list", length=V)
  for (v in seq(V)) {folds[[v]] <- c(Y0[[v]], Y1[[v]])}
  return(folds)
}

### set up screens

# note screen.glmnet is built into SuperLearner package
screens <- c("screen.glmnet", "screen.corX.6", "screen.corX.7",
             "screen.univar.logistic.2", "screen.univar.logistic.3",
             "screen.MNv2","screen.PRNT"
)

# Define non-data-adaptive methods
regular.methods <- c("SL.mean","SL.step","SL.bayesglm", "SL.glm")
```

```
# Construct SL library of non-data-adaptive methods
SL.library.regular <- lapply(regular.methods, function(method) c(method, screens))

# Define data-adaptive methods
adaptive.methods <- c("SL.polymars")
SL.library.adaptive <- lapply(adaptive.methods, function(method) c(method, screens))

SL.library <- c(SL.library.regular, SL.library.adaptive)

#######################
## Cross-Validated SuperLearner on Dummy CYD14
#######################

CV.fitSL14.v <- list()
set.seed(0987)
CV.fitSL14.v <- CV.SuperLearner(Y=Y14_vaccine, X=X14_vaccine_subset,
                                V = 7,
                                family = binomial(), SL.library,
                                method = "method.NNLS",obsWeights = weight14.v
                                ,saveAll = TRUE,control=SuperLearner.control(saveFitLibrary = TRUE)
                                ,verbose=FALSE)

set.seed(0987)
## Fit Estimated Optimal Surrogate on CYD14 Placebo Treatment Group
CV.fitSL14.p <- list()
CV.fitSL14.p <- CV.SuperLearner(Y=Y14_placebo, X=X14_placebo_subset,
                                V = 7,
                                family = binomial(), SL.library,
                                method = "method.NNLS",obsWeights = weight14.p
                                ,saveAll = TRUE,control=SuperLearner.control(saveFitLibrary = TRUE)
                                ,verbose=FALSE)

#######################
# Single run of superlearner to get model and optimal surrogate values for SuperLearner
#######################
# using the same SL.library as specified for CV.SuperLearner

# vaccine group
fitSL14.v <- list()
set.seed(1106)
```

```
fitSL14.v <- SuperLearner(Y=Y14_vaccine, X=X14_vaccine_subset,
                          family = binomial(), SL.library=SL.library,
                          method = "method.NNLS",
                           #method = "method.AUC",
                          obsWeights = weight14.v
                          ,control=SuperLearner.control(saveFitLibrary = TRUE)
                          ,verbose=FALSE)

# placebo group
fitSL14.p <- list()
set.seed(1106)
fitSL14.p <- SuperLearner(Y=Y14_placebo, X=X14_placebo_subset,
                          family = binomial(),SL.library=SL.library,
                          method = "method.NNLS",
                          #method = "method.AUC",
                          obsWeights = weight14.p
                          ,control=SuperLearner.control(saveFitLibrary = TRUE)
                          ,verbose=FALSE)



########################
## Figure 1 MSE Black and White using Dummy CYD14
########################
ForestPlotMatrix1 <- summary(CV.fitSL14.v)$Table
ForestPlotMatrix1$CVMSE<-ForestPlotMatrix1$Ave
ForestPlotMatrix1$Lower <- ForestPlotMatrix1$Ave - 1.96*ForestPlotMatrix1$se
ForestPlotMatrix1$Upper <- ForestPlotMatrix1$Ave + 1.96*ForestPlotMatrix1$se

ForestPlotDataset <-as.data.frame((ForestPlotMatrix1))

library(stringr)
splits <- matrix(unlist(strsplit(colnames(CV.fitSL14.v$library.predict), "_", fixed=TRUE)), ncol=2, byrow=TRUE)
ForestPlotDataset$Algorithm1 <- c("Super Learner","Discrete SL",str_replace(splits[,1], "SL.", ""))
ForestPlotDataset$Screen <- c("Super Learner","Discrete SL",str_replace(splits[,2], "screen.", ""))

ForestPlotDataset$Algorithm <- c("Super Learner","Discrete SL",colnames(CV.fitSL14.v$library.predict))
ForestPlotDataset$Algorithm1<- factor(ForestPlotDataset$Algorithm1,
                                      levels = c("mean",
                                                 "polymars","Super Learner","Discrete SL","glmnet"
                                                 ,"bayesglm","step","glm")
)
```

```
ForestPlotDataset <- ForestPlotDataset[order(ForestPlotDataset$CVMSE),]
ForestPlotDataset <- rbind(ForestPlotDataset[1:10,],
                           ForestPlotDataset[ForestPlotDataset$Algorithm==("Discrete SL") |
                                               ForestPlotDataset$Algorithm==("Super Learner"),])
ForestPlotDataset <- ForestPlotDataset[order(ForestPlotDataset$CVMSE),]

ForestPlotDataset$Algorithm2 <- paste(ForestPlotDataset$Algorithm1,", screen:",ForestPlotDataset$Screen,sep="")
ForestPlotDataset$Algorithm2 <- ifelse(ForestPlotDataset$Algorithm1==
                                         "Discrete SL","Discrete SL",ForestPlotDataset$Algorithm2)
ForestPlotDataset$Algorithm2 <- ifelse(ForestPlotDataset$Algorithm1==
                                         "Super Learner","Super Learner",ForestPlotDataset$Algorithm2)

ForestPlotDataset$Algorithm2 <- reorder(ForestPlotDataset$Algorithm2,
                                        -ForestPlotDataset$Ave)



ForestPLotMSE.v <-  ggplot(ForestPlotDataset) +
  geom_errorbarh(aes(x=CVMSE,xmin=Lower,xmax=Upper,y=Algorithm2)) +
  geom_point(aes(x=CVMSE,y=Algorithm2),size=0.7,alpha=0.85) +
  theme_bw() +
  xlab(’CV-MSE’) +
  ylab(’Algorithm’) +
  ggtitle("DUMMY CYD14 Vaccine CV-MSE")

######## Placebo
ForestPlotMatrix1 <- summary(CV.fitSL14.p)$Table
ForestPlotMatrix1$CVMSE<-ForestPlotMatrix1$Ave
ForestPlotMatrix1$Lower <- ForestPlotMatrix1$Ave - 1.96*ForestPlotMatrix1$se
ForestPlotMatrix1$Upper <- ForestPlotMatrix1$Ave + 1.96*ForestPlotMatrix1$se

ForestPlotDataset <-as.data.frame((ForestPlotMatrix1))
library(stringr)
splits <- matrix(unlist(strsplit(colnames(CV.fitSL14.p$library.predict), "_", fixed=TRUE)), ncol=2, byrow=TRUE)
ForestPlotDataset$Algorithm1 <- c("Super Learner","Discrete SL",str_replace(splits[,1], "SL.", ""))
ForestPlotDataset$Screen <- c("NA","NA",str_replace(splits[,2], "screen.", ""))

ForestPlotDataset$Algorithm <- c("Super Learner","Discrete SL",colnames(CV.fitSL14.p$library.predict))
ForestPlotDataset$Algorithm1<- factor(ForestPlotDataset$Algorithm1,
                                      levels = c("mean",
                                                 "polymars","Super Learner","Discrete SL","glmnet"
```

```
                                          ,"bayesglm","step","glm"))


ForestPlotDataset <- ForestPlotDataset[order(ForestPlotDataset$CVMSE),]
ForestPlotDataset <- rbind(ForestPlotDataset[1:8,],
                           ForestPlotDataset[ForestPlotDataset$Algorithm==("Discrete SL") |
                             ForestPlotDataset$Algorithm==("Super Learner"),])
ForestPlotDataset <- ForestPlotDataset[order(ForestPlotDataset$CVMSE),]



ForestPlotDataset$Algorithm2 <- paste(ForestPlotDataset$Algorithm1,", screen:",ForestPlotDataset$Screen,sep="")
ForestPlotDataset$Algorithm2 <- ifelse(ForestPlotDataset$Algorithm1==
                                         "Discrete SL","Discrete SL",ForestPlotDataset$Algorithm2)
ForestPlotDataset$Algorithm2 <- ifelse(ForestPlotDataset$Algorithm1==
                                         "Super Learner","Super Learner",ForestPlotDataset$Algorithm2)
ForestPlotDataset$Algorithm2 <- reorder(ForestPlotDataset$Algorithm2,
                                        -ForestPlotDataset$Ave)

ForestPLotMSE.p <-  ggplot(ForestPlotDataset) +
  geom_errorbarh(aes(x=CVMSE,xmin=Lower,xmax=Upper,y=Algorithm2)) +
  geom_point(aes(x=CVMSE,y=Algorithm2),size=0.7,alpha=0.85) +
  theme_bw() +
  xlab('CV-MSE') +
  ylab('Algorithm') +
  ggtitle("DUMMY CYD14 Placebo CV-MSE")



# Output Figure 1 pdf
library(gridExtra)
pdf("BiometricsPriceGilbertVanDerLaan_CVMSE_Figure1_blackwhite_DummyDataCYD14.pdf",width=8, height=4)
grid.arrange(ForestPLotMSE.v, ForestPLotMSE.p, ncol=2)
dev.off()



#######################
######### TMLE for CYD14
#######################
## Function to target (perform TMLE) on the estimated optimal surrogate
## Inputs
# w = covariates
# a = treatment group
# y = optimal surrogate
# g.SL.library = SL library desired for estimating initial g values
# wgts = case-control weights, if any
# max.wgt = if needed

weighted.tmle.EOS <- function(w,a,y,g.SL.library,wgts,max.wgt=Inf){
  require(SuperLearner)
  n <- nrow(w)
  WA <- data.frame(w,A=a)
  wgts = n*wgts/sum(wgts)
  Qbar.ests <- y
  # obtain an estimate of g_n
  temp <- SuperLearner(a,w,SL.library=g.SL.library,newX=w,obsWeights=wgts,
                       family=binomial,cvControl=list(V=10))
  g.ests <- temp$SL.predict[,1]
  rm(temp)
  g.ests[a==0] <- 1-g.ests[a==0]
  a.ind <- 2*a-1
  offset <- qlogis(pmin(pmax(Qbar.ests[1:n],0.0005),0.9995))
  g.and.wgts = pmin(wgts/g.ests,max.wgt)
  eps = coef(glm(y ˜ -1 + offset(offset) + a.ind,weights=g.and.wgts,family=binomial))
  if(a==1){QA <-plogis(qlogis(Qbar.ests[1:n]) + eps)}else{QA<-plogis(qlogis(Qbar.ests[1:n]) + eps)}
  return(list(QA = QA))}


###########################
## Function to perform TMLE with observation weighting to handle
## the case-cohort sampling design
## Inputs
# w = covariates baseline covariates (intermediate response endpoints s
#                          are not used for this TMLE)
# a = treatment group (vaccine or placebo)
# y = outcome (dengue endpoint occurrence or optimal surrogate)
# Q.SL.library = SL library desired for estimating initial Q values
# g.SL.library = SL library desired for estimating initial g values
# wgts = case-cohort weights, if any
# max.wgt = if needed
# Even though for the data sets (W,A,Y) are measured on everyone,
# the TMLE is implemented with inverse probability weighting
# because only the data set with S measured is used for the analysis.

weighted.tmle.ate <- function(w,a,y,Q.SL.library,g.SL.library,wgts,max.wgt=Inf){
  require(SuperLearner)
  n <- nrow(w)
  WA <- WA0 <- WA1<-data.frame(w,A=a)
  WA0$A <- 0 # counterfactual for control
  WA1$A <- 1 # counterfactual for case
  wgts = n*wgts/sum(wgts)
  temp <- SuperLearner(y,data.frame(w,A=a),SL.library=Q.SL.library,newX=rbind(WA,WA0,WA1),
                       obsWeights=wgts,family=binomial,cvControl=list(V=10))
  Qbar.ests <- temp$SL.predict[,1]
  rm(temp)
  temp <- SuperLearner(a,w,SL.library=g.SL.library,newX=w,obsWeights=wgts,
                       family=binomial,cvControl=list(V=10))
  g.ests <- temp$SL.predict[,1]
  rm(temp)
  g.ests[a==0] <- 1-g.ests[a==0]
  a.ind <- 2*a-1 # instead of a = 0,1, a = -1, +1
  offset <- qlogis(pmin(pmax(Qbar.ests[1:n],0.0005),0.9995))
  g.and.wgts = pmin(wgts/g.ests,max.wgt)
  eps = coef(glm(y ~ -1 + offset(offset) + a.ind,weights=g.and.wgts,family=binomial))
  Q0 <- plogis(qlogis(Qbar.ests[(n+1):(2*n)]) - eps) # Qbar.ests[(n+1):(2*n)]=WA0 predicted values
  Q1 <- plogis(qlogis(Qbar.ests[(2*n+1):(3*n)]) + eps) # Qbar.ests[(2*n+1):(3*n)]=WA1 predicted values
  est <- mean((Q1-Q0)*wgts)
  EY1 <- mean(Q1*wgts)
  EY0 <- mean(Q0*wgts)
  ic <- a.ind*g.and.wgts*(y-(a*Q1 + (1-a)*Q0)) + wgts*((Q1-Q0) - est)
  ic1 = a*g.and.wgts*(y- Q1) + wgts*(Q1-EY1) # equals only wgts*(Q1-EY1) for placebo subjects
  ic0 = (1-a)*g.and.wgts*(y- Q0) + wgts*(Q0-EY0) # equals only  wgts*(Q0-EY0) for vaccine subjects
  ci <- c(est-qnorm(0.975)*sd(ic)/sqrt(n),est+qnorm(0.975)*sd(ic)/sqrt(n))
  ci1 <- c(EY1-qnorm(0.975)*sd(ic1)/sqrt(length(ic1)),EY1+qnorm(0.975)*sd(ic1)/sqrt(length(ic1)))
  ci0 <- c(EY0-qnorm(0.975)*sd(ic0)/sqrt(length(ic0)),EY0+qnorm(0.975)*sd(ic0)/sqrt(length(ic0)))
  return(list(est=est,ci=ci,ic=ic,EY1=EY1,EY0=EY0,Q0 = Q0, Q1=Q1,
              ic1=ic1, ic0=ic0, ci1=ci1, ci0=ci0))}

####################
## For appropriate TMLE, the correct dataset size is needed;
## since observations without M13 data would have a weight of 0,
## we can use dummy data appended to the m13 data for when the function is called.
## The only variables that reflect the true data are the a (vaccine/placebo) and y (all non cases)
## CYD14 VACC 6859 and Placebo 3391
## so 6859-774=6085 dummy vaccinated and 3391-426=2965 dummy placebo
```

```
remaining14_w <- matrix(0,nrow=9050,ncol=3)
colnames(remaining14_w)<-c("AGE.9.11","AGE.12.14","MALE")
remaining14_y <- rep(0,9050)
remaining14_weight <- rep(0,9050)
remaining14_VACC <- c(rep(1,6085),rep(0,2965))

## TMLE on CYD14 with clinical outcomes
tmle.outB<- weighted.tmle.ate(w=rbind(X14_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                      X14_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                      remaining14_w)
                              ,a=c(X14_vaccine$VACC,X14_placebo$VACC,remaining14_VACC)
                              ,y=c(Y14_vaccine,Y14_placebo,remaining14_y) # clinical outcomes
                              ,Q.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                              ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                              ,wgts=c(weight14.v,weight14.p,remaining14_weight)
                              ,max.wgt=Inf
)

## TMLE on CYD14 for Relative Risk with clinical outcomes
psi14_rr <-tmle.outB$EY1/tmle.outB$EY0
logpsi14_rr <- log(psi14_rr)
var.logrr <- ((1/tmle.outB$EY1)^2)*var(tmle.outB$ic1)/length(c(weight14.v,weight14.p,remaining14_weight)) +
  ((1/tmle.outB$EY0)^2)*var(tmle.outB$ic0)/length(c(weight14.v,weight14.p,remaining14_weight))
log14_rr.ci.upper <- logpsi14_rr + 1.96*sqrt(var.logrr)
log14_rr.ci.lower <- logpsi14_rr - 1.96*sqrt(var.logrr)

psi14_ve <- 1-exp(logpsi14_rr)
psi14_ve.ci.lower <- 1-exp(log14_rr.ci.upper)
psi14_ve.ci.upper <- 1-exp(log14_rr.ci.lower)

## TMLE on CYD14 with targeted SuperLearner estimated optimal surrogate
pred.v<- fitSL14.v$SL.predict
pred.p<- fitSL14.p$SL.predict

SL.predict.TMLE.14 <- weighted.tmle.EOS(w=rbind(X14_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                           X14_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                           remaining14_w)
                                   ,a=c(X14_vaccine$VACC,X14_placebo$VACC,remaining14_VACC)
                                   ,y=c(pred.v,pred.p,remaining14_y) # optimal surrogate
                                   ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                   ,wgts=c(weight14.v,weight14.p,remaining14_weight)
                                   ,max.wgt=Inf)



tmle.outB.psi <- weighted.tmle.ate(w=rbind(X14_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                           X14_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                           remaining14_w)
                                   ,a=c(X14_vaccine$VACC,X14_placebo$VACC,remaining14_VACC)
                                   ,y=SL.predict.TMLE.14$QA # optimal surrogate
                                   ,Q.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                   ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                   ,wgts=c(weight14.v,weight14.p,remaining14_weight)
                                   ,max.wgt=Inf
)

## TMLE for Relative Risk with SuperLearner estimated optimal surrogate
psi14_rr.psi <-tmle.outB.psi$EY1/tmle.outB.psi$EY0
log.psi14_rr.psi <- log(psi14_rr.psi)
var.logrr.psi <- ((1/tmle.outB.psi$EY1)^2)*var(tmle.outB.psi$ic1)/length(c(weight14.v,weight14.p,remaining14_weight)) +
  ((1/tmle.outB.psi$EY0)^2)*var(tmle.outB.psi$ic0)/length(c(weight14.v,weight14.p,remaining14_weight))
log.psi14_rr.psi.ci.upper <- log.psi14_rr.psi + 1.96*sqrt(var.logrr.psi)
log.psi14_rr.psi.ci.lower <- log.psi14_rr.psi - 1.96*sqrt(var.logrr.psi)

psi14_ve.psi <- 1-psi14_rr.psi
psi14_ve.psi.ci.lower <- 1-exp(log.psi14_rr.psi.ci.upper)
psi14_ve.psi.ci.upper <- 1-exp(log.psi14_rr.psi.ci.lower)



########################
## Table of Dummy CYD14 Results
########################
library(xtable)
suppressPackageStartupMessages(library(xtable))
options(xtable.comment = FALSE)

CYD14_TMLEVE <- matrix(NA, ncol=4, nrow=4)
CYD14_TMLEVE[1,]<-c("",
                    paste(expression("Surrogate Parameters Based on the TMLE of the Optimal Surrogate "*theta[psi[n]^"#"]^"
                    "",
                    paste(expression("Clinical Parameters Based on the TMLE "*tilde(theta)^"TMLE")) )
CYD14_TMLEVE[2:4,1] <- c(paste(expression(theta[psi[n]^"#"]^"TMLE,1" ) ),
                         paste(expression(theta[psi[n]^"#"]^"TMLE,0" ) ),
                         paste(expression(VE[psi[n]^"#"]^"TMLE" ) ) )
CYD14_TMLEVE[2:4,2] <- c(paste(round(tmle.outB.psi$EY1,3),
                               " (95% CI ",round(tmle.outB.psi$ci1[1],3),"-",round(tmle.outB.psi$ci1[2],3),")",sep=""),
                         paste(round(tmle.outB.psi$EY0,3),
                               " (95% CI ",round(tmle.outB.psi$ci0[1],3),"-",round(tmle.outB.psi$ci0[2],3),")",sep=""),
                         paste(100*round(psi14_ve.psi,2),
                               "% (95% CI ",100*round(psi14_ve.psi.ci.lower,2),"-",100*round(psi14_ve.psi.ci.upper,2),")",s
)
CYD14_TMLEVE[2:4,3] <- c(paste(expression(E[n](Y[1])^"TMLE")),
                         paste(expression(E[n](Y[0])^"TMLE")),
                         paste(expression(VE[n]^"TMLE")))
CYD14_TMLEVE[2:4,4] <- c(paste(round(tmle.outB$EY1,3),
                               " (95% CI ",round(tmle.outB$ci1[1],3),"-",round(tmle.outB$ci1[2],3),")",sep=""),
                         paste(round(tmle.outB$EY0,3),
                               " (95% CI ",round(tmle.outB$ci0[1],3),"-",round(tmle.outB$ci0[2],3),")",sep=""),
                         paste(100*round(psi14_ve,2),
                               "% (95% CI ",100*round(psi14_ve.ci.lower,2),"-",100*round(psi14_ve.ci.upper,2),")",sep="")
)

CYD14_TMLEVE[1,]<-c("",
                    paste(
                    expression("Surrogate Parameters Based on the TMLE of the Optimal Surrogate "*theta[psi[n]^"#"]^"TMLE")
                    "",
                    paste(expression("Clinical Parameters Based on the TMLE "*tilde(theta[n])^"TMLE"))
)

print(xtable(CYD14_TMLEVE,
             caption="Comparison of inferences on the surrogate parameters and VE, versus inferences on the
             clinical dengue parameters in CYD14.  The TMLEs for the Optimal Surrogate and for the clinical outcome
             are nearly identical in values.",
             align="l|l|c|l|c", label="tab:TMLEVE" ),
      include.rownames=FALSE,
      include.colnames=FALSE, type="latex")

library(gridExtra)
pdf("BiometricsPriceGilbertVanDerLaan_EstimatesForCYD14_Section6.1_DummyDataCYD14.pdf",height=3,width=12)
tt <- ttheme_default(colhead=list(fg_params = list(parse=TRUE)),parse=TRUE)
table <- tableGrob(CYD14_TMLEVE, theme=tt)
library(grid)
library(gtable)
title <- textGrob("TMLE Estimates Reported in Section 6.1 Using Dummy CYD14 Data*",gp=gpar(fontsize=16))
```

```
footnote <- textGrob("*Results based on simulated data and do not necessarily reflect actual results",
x=0, hjust=0, gp=gpar( fontface="italic"))
padding <- unit(0.5,"line")
table <- gtable_add_rows(table,
                         heights = grobHeight(title) + padding,
                         pos = 0)
table <- gtable_add_rows(table,
                         heights = grobHeight(footnote)+ padding)
table <- gtable_add_grob(table, list(title, footnote),
                         t=c(1, nrow(table)), l=c(1,2),
                         r=ncol(table))
grid.newpage()
grid.draw(table)
dev.off()



#######################
######### TMLE for Dummy CYD15 using the Dummy CYD15 clinical outcome for VCD
######### Calculation of Clinical Parameters Based on the TMLE: tilde(theta)["n*"]^"TMLE"*(P)
#######################

## For appropriate TMLE, the correct dataset size is needed;
## since observations without M13 data would have a weight of 0,
## we can use dummy data appended to the m13 data for when the function is called.
## The only variables that reflect the true data are the a (vaccine/placebo) and y (all non cases)
## CYD15 VACC 11849 and Placebo 5803
## so 11849-1101=10748 dummy vaccinated and 5803-637=5166 dummy placebo

remaining15_w <- matrix(0,nrow=15914,ncol=3)
colnames(remaining15_w)<-c("AGE.9.11","AGE.12.14","MALE")
remaining15_y <- rep(0,15914)
remaining15_weight <- rep(0,15914)
remaining15_VACC <- c(rep(1,10748),rep(0,5166))



tmle.outB15 <-  weighted.tmle.ate(w=rbind(X15_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                          X15_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                          remaining15_w)
                                  ,a=c(X15_vaccine$VACC,X15_placebo$VACC,remaining15_VACC)
                                  ,y=c(Y15_vaccine,Y15_placebo,remaining15_y) # clinical outcome
                                  ,Q.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                  ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                  ,wgts=c(weight15.v,weight15.p,remaining15_weight)
                                  ,max.wgt=Inf
)

## TMLE for Relative Risk with clinical outcomes
psi15_rr <-tmle.outB15$EY1/tmle.outB15$EY0
log.psi15_rr <- log(psi15_rr)
var.logrr <- ((1/tmle.outB15$EY1)^2)*var(tmle.outB15$ic1)/length(tmle.outB15$ic1) +
  ((1/tmle.outB15$EY0 )^2)*var(tmle.outB15$ic0)/length(tmle.outB15$ic0)
log.psi15_rr.ci.upper <- log.psi15_rr + 1.96*sqrt(var.logrr )
log.psi15_rr.ci.lower <- log.psi15_rr - 1.96*sqrt(var.logrr )

psi15_ve <- 1-psi15_rr
psi15_ve.ci.lower <- 1-exp(log.psi15_rr.ci.upper)
psi15_ve.ci.upper <- 1-exp(log.psi15_rr.ci.lower)

#######################
######### TMLE for E[Y] and VE for Dummy CYD15
#########   using the Dummy CYD14-estimated optimal surrogate applied to Dummy CYD15
######### Calculation of Surrogate Parameters Based on the TMLE of the Optimal Surrogate: theta[psi[n]^"#"]^"TMLE"
#######################

##########################
### calculate for Dummy CYD15 the Dummy CYD14-estimated optimal surrogate
NewdataV <- X15_vaccine[,c(2:18,21:23)]
NewdataP <- X15_placebo[,c(2:18,21:23)]

CYD15V <- predict.SuperLearner(object=fitSL14.v, newdata=NewdataV)#, onlySL = TRUE)
pred15.v<- CYD15V$pred
## placebo group with placebo model
CYD15P <- predict.SuperLearner(object=fitSL14.p, newdata=NewdataP)#, onlySL = TRUE)
pred15.p<- CYD15P$pred

## TMLE on Dummy CYD15 with Dummy CYD14-estimated optimal surrogate
SL.predict.TMLE.15 <- weighted.tmle.EOS(w=rbind(X15_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                                X15_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                                remaining15_w)
                                        ,a=c(X15_vaccine$VACC,X15_placebo$VACC,remaining15_VACC)
                                        ,y=c(pred15.v,pred15.p,remaining15_y) # optimal surrogate
                                        ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                        ,wgts=c(weight15.v,weight15.p,remaining15_weight)
                                        ,max.wgt=Inf)

## TMLE on Dummy CYD15 with Dummy CYD14-estimated optimal surrogate
tmle.outB15.psi <-  weighted.tmle.ate(w=rbind(X15_vaccine[,c("AGE.9.11","AGE.12.14","MALE")],
                                              X15_placebo[,c("AGE.9.11","AGE.12.14","MALE")],
                                              remaining15_w)
                                      ,a=c(X15_vaccine$VACC,X15_placebo$VACC,remaining15_VACC)
                                      ,y=c(SL.predict.TMLE.15$QA)
                                      ,Q.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                      ,g.SL.library=c("SL.mean","SL.glm","SL.glm.interaction")
                                      ,wgts=c(weight15.v,weight15.p,remaining15_weight)
                                      ,max.wgt=Inf
)

##  Dummy CYD15 TMLE for Relative Risk estimated with optimal surrogate
psi15_rr.psi <-tmle.outB15.psi$EY1/tmle.outB15.psi$EY0
log.psi15_rr.psi <- log(psi15_rr.psi)
var.logrr.psi <- ((1/tmle.outB15.psi$EY1)^2)*var(tmle.outB15.psi$ic1)/length(tmle.outB15.psi$ic1) +
  ((1/tmle.outB15.psi$EY0)^2)*var(tmle.outB15.psi$ic0)/length(tmle.outB15.psi$ic0)

log.psi15_rr.psi.ci.upper <- log.psi15_rr.psi + 1.96*sqrt(var.logrr.psi )
log.psi15_rr.psi.ci.lower <- log.psi15_rr.psi - 1.96*sqrt(var.logrr.psi )

psi15_ve.psi <- 1-psi15_rr.psi
psi15_ve.psi.ci.lower <- 1-exp(log.psi15_rr.psi.ci.upper)
psi15_ve.psi.ci.upper <- 1-exp(log.psi15_rr.psi.ci.lower)



########################
## Table 3 Results on Dummy Data
########################
library(xtable)
suppressPackageStartupMessages(library(xtable))
options(xtable.comment = FALSE)

CYD15_TMLEVE <- matrix(NA, ncol=4, nrow=4)
CYD15_TMLEVE[1,]<-c("",
                    paste(
                    expression("Surrogate Parameters Based on the TMLE of the Optimal Surrogate "*theta[psi[n]^"#"]^"TMLE")
                    "",
                    paste(expression("Clinical Parameters Based on the TMLE "*tilde(theta)["n*"]^"TMLE"*(P)))
)
CYD15_TMLEVE[2:4,1] <- c(paste(expression(theta[psi[n]^"#"]^1*(P)) ),
                         paste(expression(theta[psi[n]^"#"]^0*(P) ) ),
                         paste(expression(VE[psi[n]^"#"](P)) ) )
CYD15_TMLEVE[2:4,2] <- c(paste(round(tmle.outB15.psi$EY1,3),
                               " (95% CI ",round(tmle.outB15.psi$ci1[1],3),"-",round(tmle.outB15.psi$ci1[2],3),")",sep=""),
                         paste(round(tmle.outB15.psi$EY0,3),
                               " (95% CI ",round(tmle.outB15.psi$ci0[1],3),"-",round(tmle.outB15.psi$ci0[2],3),")",sep=""),
                         paste(100*round(psi15_ve.psi,2),"%",
                               " (95% CI ",100*round(psi15_ve.psi.ci.lower,2),"-",
                               100*round(psi15_ve.psi.ci.upper,2),")",sep="")
)
CYD15_TMLEVE[2:4,3] <- c(paste(expression(E[P](Y[1]^"*"))),
                         paste(expression(E[P](Y[0]^"*"))),
                         paste(expression(VE[P]^"*")))
CYD15_TMLEVE[2:4,4] <- c(paste(round(tmle.outB15$EY1,3),
                               " (95% CI ",round(tmle.outB15$ci1[1],3),"-",round(tmle.outB15$ci1[2],3),")",sep=""),
                         paste(round(tmle.outB15$EY0,3),
                               " (95% CI ",round(tmle.outB15$ci0[1],3),"-",round(tmle.outB15$ci0[2],3),")",sep=""),
                         paste(100*round(psi15_ve,2),"%",
                               " (95% CI ",100*round(psi15_ve.ci.lower,2),"-",100*round(psi15_ve.ci.upper,2),")",sep="")
)



print(xtable(CYD15_TMLEVE,
             caption="Comparison of inferences on the surrogate parameters and VE,
             versus inferences on the clinical dengue parameters in CYD15.",
             align="l|l|c|l|c", label="tab:TMLEVECYD15" ),
      include.rownames=FALSE,
      include.colnames=FALSE, type="latex")

library(gridExtra)
pdf("BiometricsPriceGilbertVanDerLaan_Table3_DummyDataCYD14CYD15.pdf",height=3,width=12)
tt <- ttheme_default(colhead=list(fg_params = list(parse=TRUE)),parse=TRUE)
table <- tableGrob(CYD15_TMLEVE, theme=tt)
library(grid)
library(gtable)
title <- textGrob("Recreation of Table 3 Using Dummy Data* for CYD14 and CYD15",gp=gpar(fontsize=16))
footnote <- textGrob("*Results based on simulated data and do not necessarily reflect actual results",
x=0, hjust=0, gp=gpar( fontface="italic"))
```

```
padding <- unit(0.5,"line")
table <- gtable_add_rows(table,
                         heights = grobHeight(title) + padding,
                         pos = 0)
table <- gtable_add_rows(table,
                         heights = grobHeight(footnote)+ padding)
table <- gtable_add_grob(table, list(title, footnote),
                         t=c(1, nrow(table)), l=c(1,2),
                         r=ncol(table))
grid.newpage()
grid.draw(table)
dev.off()

########################
## Figure 2: Reverse CDF Plots
########################
ReverseCDF14 <- function(Vvalues,Pvalues,VYp,PYp,sample,pct=0.05){
  rcdf <- function (x) {
    cdf <- ecdf(x)
    y <- cdf(x)
    xrcdf <- 1-y
  }
  dsetV <- data.frame(cbind(VYp,Vvalues))
  dsetP <- data.frame(cbind(PYp,Pvalues))
  names(dsetV) <- c("Yp","SL.predict")
  names(dsetP) <- c("Yp","SL.predict")
  dsetCaseV <- dsetV[VYp==1,]
  dsetCaseV$rcdf <- rcdf(dsetCaseV$SL.predict)
  dsetControlV <- dsetV[VYp==0,]
  dsetControlV$rcdf <- rcdf(dsetControlV$SL.predict)
  dsetCaseP <- dsetP[PYp==1,]
  dsetCaseP$rcdf <- rcdf(dsetCaseP$SL.predict)
  dsetControlP <- dsetP[PYp==0,]
  dsetControlP$rcdf <- rcdf(dsetControlP$SL.predict)
  xlabs <- expression(paste("Estimated Optimal Surrogate ", psi["n"]^"#", "(W,A,S) = s", sep="") )
  ylabs <- expression(paste("Probability ", psi["n"]^"#","(W,A,S)", sep="") >= s )

  title <- paste(sample)
  pctlabel=pct*100
  # calculate the 95th percentile for cases and controls
  Vq95C <- quantile(dsetCaseV$SL.predict, probs = 1- pct, na.rm = TRUE)
  Vq95c <- quantile(dsetControlV$SL.predict, probs = 1- pct, na.rm = TRUE)
  Pq95C <- quantile(dsetCaseP$SL.predict, probs = 1- pct, na.rm = TRUE)
  Pq95c <- quantile(dsetControlP$SL.predict, probs = 1- pct, na.rm = TRUE)
  # percentiles for Control 95th for catching # cases
  Vq95cC <- ecdf(dsetCaseV$SL.predict)(Vq95c)
  Pq95cC <- ecdf(dsetCaseP$SL.predict)(Pq95c)

  ggplot(dsetCaseV, aes(x = SL.predict,y=rcdf))+geom_step(colour="black") +
    labs(x=xlabs, y=ylabs) +
    ylim(0, 1) + xlim(0, 0.2) + theme_bw() + ggtitle(title) +
    geom_step(data=dsetControlV, aes(x = SL.predict,y=rcdf),colour="grey") +
    geom_step(data=dsetCaseP, aes(x = SL.predict,y=rcdf),colour="black",linetype=2) +
    geom_step(data=dsetControlP, aes(x = SL.predict,y=rcdf),colour="grey",colour="black",linetype=2) +
    geom_hline(aes(yintercept=pct),linetype = 1,colour="black")+
    #legend labels
    annotate("text", x = 0.10, y = 0.95, label = paste("Vaccine Case (",pctlabel,"th pctl ",
    round(Vq95C,2),")",sep=""),cex=2.5,hjust = 0) +
    annotate("text", x = 0.10, y = 0.90, label = paste("Placebo Case (",pctlabel,"th pctl ",
    round(Pq95C,2),")",sep=""),cex=2.5,hjust = 0) +
    annotate("text", x = 0.10, y = 0.85, label = paste("Vaccine Control (",pctlabel,"th pctl ",
    round(Vq95c,2),")",sep=""),cex=2.5,hjust = 0) +
    annotate("text", x = 0.10, y = 0.80, label = paste("Placebo Control (",pctlabel,"th pctl ",
    round(Pq95c,2),")",sep=""),cex=2.5,hjust = 0) +
    # legend line segments
    annotate("segment", x = 0.06, xend = 0.08, y = 0.95, yend = 0.95,
             colour = "black",linetype=1) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.90, yend = 0.90,
             colour = "black",linetype=2) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.85, yend = 0.85,
             colour = "grey",linetype=1) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.80, yend = 0.80,
             colour = "grey",linetype=2) +
    guides(col = guide_legend()) +
    theme(axis.text=element_text(size=8),
          axis.title=element_text(size=8,face="bold"),
          title=element_text(size=8) #,face="bold"))
    )
}

ReverseCDF15 <- function(Vvalues,Pvalues,VYp,PYp,sample,pct=0.05){
  rcdf <- function (x) {
```

```
  cdf <- ecdf(x)
  y <- cdf(x)
  xrcdf <- 1-y
}
dsetV <- data.frame(cbind(VYp,Vvalues))
dsetP <- data.frame(cbind(PYp,Pvalues))
names(dsetV) <- c("Yp","SL.predict")
names(dsetP) <- c("Yp","SL.predict")
dsetCaseV <- dsetV[VYp==1,]
dsetCaseV$rcdf <- rcdf(dsetCaseV$SL.predict)
dsetControlV <- dsetV[VYp==0,]
dsetControlV$rcdf <- rcdf(dsetControlV$SL.predict)
dsetCaseP <- dsetP[PYp==1,]
dsetCaseP$rcdf <- rcdf(dsetCaseP$SL.predict)
dsetControlP <- dsetP[PYp==0,]
dsetControlP$rcdf <- rcdf(dsetControlP$SL.predict)
xlabs <- expression(paste("Estimated Optimal Surrogate ", psi["n"]^"#", "(W*,A*,S*) = s", sep="") )
ylabs <- expression(paste("Probability ", psi["n"]^"#","(W*,A*,S*)", sep="") >= s )

title <- paste(sample)
pctlabel=pct*100
# calculate the 95th percentile for cases and controls
Vq95C <- quantile(dsetCaseV$SL.predict, probs = 1- pct, na.rm = TRUE)
Vq95c <- quantile(dsetControlV$SL.predict, probs = 1- pct, na.rm = TRUE)
Pq95C <- quantile(dsetCaseP$SL.predict, probs = 1- pct, na.rm = TRUE)
Pq95c <- quantile(dsetControlP$SL.predict, probs = 1- pct, na.rm = TRUE)
# percentiles for Control 95th for catching # cases
Vq95cC <- ecdf(dsetCaseV$SL.predict)(Vq95c)
Pq95cC <- ecdf(dsetCaseP$SL.predict)(Pq95c)

ggplot(dsetCaseV, aes(x = SL.predict,y=rcdf))+geom_step(colour="black") +
  labs(x=xlabs, y=ylabs) +
  ylim(0, 1) + xlim(0, 0.2) + theme_bw() + ggtitle(title) +
  geom_step(data=dsetControlV, aes(x = SL.predict,y=rcdf),colour="grey") +
  geom_step(data=dsetCaseP, aes(x = SL.predict,y=rcdf),colour="black",linetype=2) +
  geom_step(data=dsetControlP, aes(x = SL.predict,y=rcdf),colour="grey",colour="black",linetype=2) +
  geom_hline(aes(yintercept=pct),linetype = 1,colour="black")+
  #legend labels
  annotate("text", x = 0.10, y = 0.95, label = paste("Vaccine Case (",pctlabel,"th pctl ",
  round(Vq95C,2),")",sep=""),cex=2.5,hjust = 0) +
  annotate("text", x = 0.10, y = 0.90, label = paste("Placebo Case (",pctlabel,"th pctl ",
    round(Pq95C,2),")",sep=""),cex=2.5,hjust = 0) +
    annotate("text", x = 0.10, y = 0.85, label = paste("Vaccine Control (",pctlabel,"th pctl ",
    round(Vq95c,2),")",sep=""),cex=2.5,hjust = 0) +
    annotate("text", x = 0.10, y = 0.80, label = paste("Placebo Control (",pctlabel,"th pctl ",
    round(Pq95c,2),")",sep=""),cex=2.5,hjust = 0) +
    # legend line segments
    annotate("segment", x = 0.06, xend = 0.08, y = 0.95, yend = 0.95,
             colour = "black",linetype=1) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.90, yend = 0.90,
             colour = "black",linetype=2) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.85, yend = 0.85,
             colour = "grey",linetype=1) +
    annotate("segment", x = 0.06, xend = 0.08, y = 0.80, yend = 0.80,
             colour = "grey",linetype=2) +
    guides(col = guide_legend()) +
    theme(axis.text=element_text(size=8),
          axis.title=element_text(size=8,face="bold"),
          title=element_text(size=8) #,face="bold"))
    )
}

Plot2a <- ReverseCDF14(Vvalues=CV.fitSL14.v$SL.predict,
                       Pvalues=CV.fitSL14.p$SL.predict,
                       VYp=Y14_vaccine,
                       PYp=Y14_placebo,
                       sample="(a) DUMMY CYD14 Reverse CDFs",pct=0.05)

Plot2b <- ReverseCDF15(Vvalues=CYD15V$pred,
                       Pvalues=CYD15P$pred,
                       VYp=Y15_vaccine,
                       PYp=Y15_placebo,
                       sample="(b) DUMMY CYD15 Reverse CDFs",pct=0.05)

pdf("BiometricsPriceGilbertVanDerLaan_ReverseCDFs_Figure2_DummyDataCYD14.pdf",width=7, height=2.75)
library("gridExtra")
grid.arrange(Plot2a, Plot2b, ncol=2)
dev.off()

########################
## Table 2 Results (Discrete SuperLearner Models)
########################
```

```
Table2vaccine <- matrix(NA,ncol=4,nrow=nrow(summary(fitSL14.v$fitLibrary$SL.glm_screen.MNv2$object)$coefficients))
Table2placebo <- matrix(NA,ncol=4,nrow=nrow(summary(fitSL14.p$fitLibrary$SL.glm_screen.MNv2$object)$coefficients))

colnames(Table2vaccine) <- colnames(Table2placebo) <- c("Variable","Coefficient","Odds Ratio","p-value")
# Output the coefficients for the Best Stepwise model (the discrete superlearner)
# Variable names
Table2vaccine[,1] <- c(names(summary(fitSL14.v$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1] ))
# Coefficients
Table2vaccine[,2]<-round(summary(fitSL14.v$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1],2)
# Odds Ratio
Table2vaccine[,3]<-round(exp(summary(fitSL14.v$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1]),4)
# p-value
Table2vaccine[,4]<-signif(as.numeric(summary(fitSL14.v$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,4]),2)

# Variable names
Table2placebo[,1] <- c(names(summary(fitSL14.p$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1] ))
# Coefficients
Table2placebo[,2]<-round(summary(fitSL14.p$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1],2)
# Odds Ratio
Table2placebo[,3]<-round(exp(summary(fitSL14.p$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,1]),4)
# p-value
Table2placebo[,4]<-signif(as.numeric(summary(fitSL14.p$fitLibrary$SL.glm_screen.MNv2$object)$coefficients[,4]),2)

library(xtable)
print(xtable(Table2vaccine,
             caption="Model terms for a logistic regression based on variables
             selected from the MNv2 screen (which disallows any PRNT titer variables).",
             align="l|c|c|c|l", label="tab:BestVacc" ),
      include.rownames=FALSE,
      include.colnames=TRUE, type="latex")

print(xtable(Table2placebo,
             caption="Model terms for a logistic regression based on variables
             selected from the MNv2 screen (which disallows any PRNT titer variables).",
             align="l|c|c|c|l", label="tab:BestPlac" ),
      include.rownames=FALSE,
      include.colnames=TRUE, type="latex")
## end of Table 2 results


########################
```

```
### Supplemental Figures
########################

############################
## Supplemental Figure Function for Dummy CYD14; Supplemental Figures 1 & 2
############################
## Function to output the Violin/Bean plots for CYD14
## Inputs
# dataset = dataset with the variables and covariates
# var1, var2, var3, var4 = Variable to be plotted; in this case, PRNT or MNv2 Month 13 serotype titers
# title_X = X axis label
# title_study = Plot title

Distribution_plot14 <- function(dataset,var1,var2,var3,var4,title_X,title_study,title_Y){
  library(ggplot2)
  ylim <- c(-1, 5)
  B1 <- var1
  B2 <- var2
  B3 <- var3
  B4 <- var4
  c <- length(B1)
  d2 <- data.frame(
    Serotype=factor(
      c(rep(’Type 1’,c),rep(’Type 2’,c),rep(’Type 3’,c),rep(’Type 4’,c)),
      levels=c(’Type 1’,’Type 2’,’Type 3’,’Type 4’)
    ),
    Value=c(B1,B2,B3,B4)
  )
  d2$Vaccine <- rep(dataset$VACC,4)
  d2$Vaccine <- ifelse(d2$Vaccine==1,"Vaccine","Placebo")
  d2$MALE <- c(rep(dataset$MALE,4))
  d2$AGE<- c(rep(dataset$AGEYRS,4))
  d2$SEX <- ifelse(d2$MALE== 1, "Male","Female")
  d2$AGE <- ifelse(d2$AGE<9,"2-8",d2$AGE )
  d2$AGE <- ifelse(d2$AGE==9 | d2$AGE==10 | d2$AGE==11,"9-11",d2$AGE )
  d2$AGE <- ifelse(d2$AGE==12 | d2$AGE==13| d2$AGE==14,"12-14",d2$AGE )
  d2$AGE <- paste("Age: ",d2$AGE,sep="")
  d2$AGE_F <- as.factor(d2$AGE)
  d2$AGE_F = factor(d2$AGE_F,levels(d2$AGE_F)[c(2,3,1)])
  d2$Facets <- paste(d2$Vaccine,", ",d2$SEX,", ",d2$AGE_F,sep="")
  d2$Facets <- as.factor(d2$Facets)
  d2$Facets = factor(d2$Facets,levels(d2$Facets)[c(2,3,1,5,6,4,8,9,7,11,12,10)])


  plotp <-ggplot(data=d2)+
    geom_violin(aes(x=Serotype,y=Value),fill='grey',trim=F)+
    geom_segment(aes(
      x=match(Serotype,levels(Serotype))-0.05,
      xend=match(Serotype,levels(Serotype))+0.05,
      y=Value,yend=Value),
      col='black'
    )+ theme_bw() +
    ggplot2::ylab(title_Y) +
    ggplot2::xlab(title_X) +
    ggtitle(title_study) +
    theme(axis.text=element_text(size=10),
          axis.title=element_text(size=12,face="bold"),
          plot.title = element_text(size = rel(1)),
          strip.text=element_text(size=10, lineheight=0.2),
          strip.text.x=element_text(size=10, lineheight=0.2),
          strip.text.y=element_text(size=10, lineheight=0.2))+
    facet_wrap(~Facets, ncol = 3)
  return(plotp)
}



Distribution_plot15 <- function(dataset,var1,var2,var3,var4,title_X,title_study,title_Y){
  library(ggplot2)
  ylim <- c(-1, 5)
  B1 <- var1
  B2<- var2
  B3 <- var3
  B4 <- var4
  c <- length(B1)
  d2 <- data.frame(
    Serotype=factor(
      c(rep('Type 1',c),rep('Type 2',c),rep('Type 3',c),rep('Type 4',c)),
      levels=c('Type 1','Type 2','Type 3','Type 4')
    ),
    Value=c(B1,B2,B3,B4)
  )
  d2$Vaccine <- rep(dataset$VACC,4)
  d2$Vaccine <- ifelse(d2$Vaccine==1,"Vaccine","Placebo")
  d2$MALE <- c(rep(dataset$MALE,4))
  d2$AGE<- c(rep(dataset$AGE,4))
  d2$SEX <- ifelse(d2$MALE== 1, "Male","Female")
  d2$AGE <- ifelse(d2$AGE=="<=5","2-5",d2$AGE )
  d2$AGE <- ifelse(d2$AGE=="<=11","9-11",d2$AGE )
  d2$AGE <- ifelse(d2$AGE==">11","12-14",d2$AGE )
  d2$AGE <- paste("Age: ",d2$AGE,sep="")
  d2$AGE_F <- as.factor(d2$AGE)
  d2$AGE_F = factor(d2$AGE_F,levels(d2$AGE_F)[c(2,1)])
  d2$Facets <- paste(d2$Vaccine,", ",d2$SEX,", ",d2$AGE_F,sep="")
  d2$Facets <- as.factor(d2$Facets)
  d2$Facets = factor(d2$Facets,levels(d2$Facets)[c(2,1,4,3,6,5,8,7)])

  plotp <-ggplot(data=d2)+
    geom_violin(aes(x=Serotype,y=Value),fill='grey',trim=F)+
    geom_segment(aes(
      x=match(Serotype,levels(Serotype))-0.05,
      xend=match(Serotype,levels(Serotype))+0.05,
      y=Value,yend=Value),
      col='black'
    )+ theme_bw() +
    ggplot2::ylab(title_Y) +
    ggplot2::xlab(title_X) +
    ggtitle(title_study) +
    theme(axis.text=element_text(size=10),
          axis.title=element_text(size=12,face="bold"),
          plot.title = element_text(size = rel(1)),
          strip.text=element_text(size=10, lineheight=0.2),
          strip.text.x=element_text(size=10, lineheight=0.2),
          strip.text.y=element_text(size=10, lineheight=0.2))+
    facet_wrap(~Facets, ncol = 2)
  return(plotp)
}


############################
## Supplemental Figure 1: Dummy CYD14 M13 PRNT
############################

pdf("BiometricsPriceGilbertVanDerLaan_SuppFig1_BeanPlots_PRNT_M13_DummyDataCYD14.pdf",width= 8, height=8)
Distribution_plot14(dataset=d14,
                    var1=d14$M13_PRNT_Sero1c,
                    var2=d14$M13_PRNT_Sero2c,
                    var3=d14$M13_PRNT_Sero3c,
                    var4=d14$M13_PRNT_Sero4c,
                    title_X= "Serotypes",
                    title_study="DUMMY CYD14",
                    title_Y=expression('Month 13 Log'[10]* ' PRNT'[50]*' Neutralization Titer'))
dev.off()


############################
## Supplemental Figure 2: Dummy CYD14 Month 13 MNv2
############################

pdf("BiometricsPriceGilbertVanDerLaan_SuppFig2_BeanPlots_MNv2_M13_DummyDataCYD14.pdf",width= 8, height=8)
Distribution_plot14(dataset=d14,
                    var1=d14$M13_MNv2_Sero1c,
                    var2=d14$M13_MNv2_Sero2c,
                    var3=d14$M13_MNv2_Sero3c,
                    var4=d14$M13_MNv2_Sero4c,
                    title_X= "Serotypes",
                    title_study="DUMMY CYD14",
                    title_Y=expression('Month 13 Log'[10]* ' MNv2'*' Neutralization Titer'))
dev.off()


############################
## Supplemental Figure Function for CYD15; Supplemental Figures 3 & 4
############################

############################
## Supplemental Figure 3: CYD15 Month 13 PRNT
############################

pdf("BiometricsPriceGilbertVanDerLaan_SuppFig3_BeanPlots_PRNT_M13_DummyDataCYD15.pdf",width= 6, height=8)
Distribution_plot15(dataset=d15,
                    var1=d15$M13_PRNT_Sero1c,
                    var2=d15$M13_PRNT_Sero2c,
                    var3=d15$M13_PRNT_Sero3c,
                    var4=d15$M13_PRNT_Sero4c,
                    title_X= "Serotypes",
                    title_study="DUMMY CYD15",
                    title_Y=expression(’Month 13 Log’[10]* ’ PRNT’[50]*’ Neutralization Titer’))
dev.off()

############################
## Supplemental Figure 4: Dummy CYD15 Month 13 MNv2
############################

pdf("BiometricsPriceGilbertVanDerLaan_SuppFig4_BeanPlots_MNv2_M13_DummyDataCYD15.pdf",width= 6, height=8)
Distribution_plot15(dataset=d15,
                    var1=d15$M13_MNv2_Sero1c,
                    var2=d15$M13_MNv2_Sero2c,
                    var3=d15$M13_MNv2_Sero3c,
                    var4=d15$M13_MNv2_Sero4c,
                    title_X= "Serotypes",
                    title_study="DUMMY CYD15",
                    title_Y=expression(’Month 13 Log’[10]* ’ MNv2’*’ Neutralization Titer’))
dev.off()


############################
### Supplemental Figures 5 & 6
############################

## Single run of the SuperLearner on Dummy CYD15 to get predicted values
library(SuperLearner)
set.seed(1106)
## vaccine group
fit_Va15<- SuperLearner(Y=Y15_vaccine, X=cbind(X15_vaccine_subset),
                        family = binomial(), SL.library=SL.library,
                        method = "method.NNLS",obsWeights = weight15.v
                        ,control = list(saveFitLibrary = TRUE)
                        ,verbose=FALSE)
set.seed(1106)
## placebo group
fit_Pa15<- SuperLearner(Y=Y15_placebo, X=cbind(X15_placebo_subset),
                        family = binomial(), SL.library=SL.library,
                        method = "method.NNLS",obsWeights = weight15.p
                        ,control = list(saveFitLibrary = TRUE)
                        ,verbose=FALSE)
```

```
## CYD15 predicted values Dummy CYD14 SuperLearner

CYD15V <- predict.SuperLearner(object=fitSL14.v, newdata=X15_vaccine_subset)

CYD15P <- predict.SuperLearner(object=fitSL14.p, newdata=X15_placebo_subset)


## Difference between Dummy CYD15 outcome predicted values and Dummy CYD14 outcome predicted values (on Dummy CYD14 data)

TransportV <- fit_Va15$SL.predict-CYD15V$pred

TransportP <- fit_Pa15$SL.predict-CYD15P$pred




############################

## Supplemental Figure 5: Transportability M13 Microneutralization

############################


library(ggplot2)


d2V <- d15[d15$VACC==1,]

d2P <- d15[d15$VACC==0,]

d2 <- as.data.frame(rbind(d2V,d2P))

d2$Value <- c(TransportV,TransportP)

d2$Vaccine <- ifelse(d2$VACC==1,"Vaccine","Placebo")

d2$SEX <- ifelse(d2$MALE== 1, "M","F")

d2$AGE <- ifelse(d2$AGE=="<=11","9-11","12-14")

d2$AGE <- paste("Age: ",d2$AGE,sep="")

d2$AGE_F <- as.factor(d2$AGE)

d2$AGE_F = factor(d2$AGE_F,levels(d2$AGE_F)[c(2,1)])

d2$Facets <- paste(d2$Vaccine,", ",d2$SEX,", ",d2$AGE_F,sep="")

d2$Facets <- as.factor(d2$Facets)

d2$Facets = factor(d2$Facets,levels(d2$Facets)[c(2,1,4,3,6,5,8,7)])

d2$Facets2 <- paste(d2$SEX,", ",d2$AGE_F,sep="")

d2$Facets2 <- as.factor(d2$Facets2)

d2$Facets2 = factor(d2$Facets2,levels(d2$Facets2)[c(2,1,4,3)])




d2rep <- rbind(d2,d2,d2,d2,d2,d2,d2,d2)

d2rep$SeroValue <-NA

n <- nrow(d2)

d2rep$SeroValue <-c(d2rep$M13_MNv2_Sero1c[1:n],

                    d2rep$M13_MNv2_Sero2c[1:n],

                    d2rep$M13_MNv2_Sero3c[1:n],

                    d2rep$M13_MNv2_Sero4c[1:n],

                    d2rep$M13_PRNT_Sero1c[1:n],
                    d2rep$M13_PRNT_Sero2c[1:n],
                    d2rep$M13_PRNT_Sero3c[1:n],
                    d2rep$M13_PRNT_Sero4c[1:n])
d2rep$SeroType <- c(rep(’Type 1’,n),rep(’Type 2’,n), # MNv2
                    rep(’Type 3’,n),rep(’Type 4’,n), # MNv2
                    rep(’Type 1’,n),rep(’Type 2’,n), # PRNT
                    rep(’Type 3’,n),rep(’Type 4’,n)) # PRNT

d2rep$Facets3 <- paste(d2rep$Vaccine,", ",d2rep$SEX,", ",d2rep$AGE_F,", ",d2rep$SeroType,sep="")

d2MVn2 <- d2rep[1:(4*n),] #MNv2
d2PRNT <- d2rep[(4*n +1):(8*n),] #PRNT
d2MVn2$Facets3 <- as.factor(d2MVn2$Facets3)
d2MVn2$Facets3 = factor(d2MVn2$Facets3,levels(d2MVn2$Facets3)[
  c(5,6,7,8,13,14,15,16,1,2,3,4,9,10,11,12,
    21,22,23,24,29,30,31,32,17,18,19,20,25,26,27,28)])
d2PRNT$Facets3 <- as.factor(d2PRNT$Facets3)
d2PRNT$Facets3 = factor(d2PRNT$Facets3,levels(d2PRNT$Facets3)[
  c(5,6,7,8,13,14,15,16,1,2,3,4,9,10,11,12,
    21,22,23,24,29,30,31,32,17,18,19,20,25,26,27,28)])



pdf("BiometricsPriceGilbertVanDerLaan_SuppFig5_Transport_MNv2_DummyDataCYD14CYD15.pdf",width= 9, height=12)
ggplot(data=d2MVn2)+
  geom_point(aes(x = Value,y=SeroValue),size=0.1) +
  theme_bw() +
  ggplot2::xlab(expression(
    ’E’ * "[" * ’Y*’ * "|" * ’W*’ * ’=’ * ’w’ * ’,’ * ’A*’ * ’=’ * ’a’ * ’,’ *’S*’ * ’=’ *’s’ * "]" * ’
    - ’ * ’E’ * "[" * ’Y’ * "|" * ’W’ * ’=’ * ’w’ * ’,’ * ’A’ * ’=’ * ’a’ * ’,’ *’S’ * ’=’ *’s’ * "]" )) +
  ggplot2::ylab(expression(Log[10]~Month~13~MNv2~Titer)) +
  ggtitle(’Assessment of Equal Conditional Means for DUMMY Month 13 MNv2 Titers’) +
  theme(axis.text=element_text(size=10,angle = 90),
        axis.title=element_text(size=12,face="bold"),
        plot.title = element_text(size = rel(1),hjust = 0.5),
        strip.text=element_text(size=8, lineheight=0.2),
        strip.text.x=element_text(size=8, lineheight=0.2),
        strip.text.y=element_text(size=8, lineheight=0.2)) +
  facet_wrap(~Facets3, ncol = 4)
dev.off()

############################
```

```
## Supplemental Figure 6: Transportability Month 13 PRNT
############################


library(ggplot2)
pdf("BiometricsPriceGilbertVanDerLaan_SuppFig6_Transport_PRNT_DummyDataCYD14CYD15.pdf",width= 9, height=12)
ggplot(data=d2PRNT)+
  geom_point(aes(x = Value,y=SeroValue),size=0.1) +
  theme_bw() +
  ggplot2::xlab(expression(
    ’E’ * "[" * ’Y*’ * "|" * ’W*’ * ’=’ * ’w’ * ’,’ * ’A*’ * ’=’ * ’a’ * ’,’ *’S*’ * ’=’ *’s’ * "]" * ’
    - ’ * ’E’ * "[" * ’Y’ * "|" * ’W’ * ’=’ * ’w’ * ’,’ * ’A’ * ’=’ * ’a’ * ’,’ *’S’ * ’=’ *’s’ * "]" )) +
  ggplot2::ylab(expression(Log[10]~Month~13~PRNT[50]~Titer)) +
  ggtitle(’Assessment of Equal Conditional Means for DUMMY Month 13 PRNT Titers’) +
  theme(axis.text=element_text(size=10,angle = 90),
        axis.title=element_text(size=12,face="bold"),
        plot.title = element_text(size = rel(1),hjust = 0.5),
        strip.text=element_text(size=8, lineheight=0.2),
        strip.text.x=element_text(size=8, lineheight=0.2),
        strip.text.y=element_text(size=8, lineheight=0.2)) +
  facet_wrap(~Facets3, ncol = 4)
dev.off()
## end of supplemental figures

### end of file BiometricsPriceGilbertVanDerLaanDummyDataDengueExampleCode.R ###
```

References

Alonso, A., Molenberghs, G., Geys, H., Buyse, M., and Vangeneugden, T. (2006). A unifying approach for surrogate marker validation based on Prentice’s criteria. *Statistics in Medicine* **25,** 205–221.

Bareinboim, E. and Pearl, J. (2012). Transportability of causal effects: Completeness results. *Proceedings of the Twenty-Sixth National Conference on Artificial Intelligence, Menlo Park, CA* pages 698–704.

Buyse, M. and Molenberghs, G. (1998). Criteria for the validation of surrogate endpoints in randomized experiments. *Biometrics* **54,** 1014–1029.

Buyse, M., Molenberghs, G., Burzykowski, T., Renard, D., and Geys, H. (2000). The validation of surrogate endpoints in meta-analyses of randomized experiments. *Biostatistics* **1,** 49–67.

Chan, I., Shu, L., Matthews, H., Chan, C., Vessey, R., Sadoff, J., and Heyse, J. (2002). Use of statistical models for evaluating antibody response as a correlate of protection against varicella. *Statistics in Medicine* **21,** 3411–3430.

Chen, H., Geng, Z., and Jia, J. (2007). Criteria for surrogate end points. *Journal of the Royal Statistical Society, Series B* **69,** 919–932.

Daniels, M. and Hughes, M. (1997). Meta-analysis for the evaluation of potential surrogate markers. *Statistics in Medicine* **16,** 1965–1982.

Freedman, L., Graubard, B., and Schatzkin, A. (1992). Statistical validation of intermediate endpoints for chronic diseases. *Statistics in Medicine* **11,** 167–178.

Gabriel, E. and Gilbert, P. (2014). Evaluating principle surrogate endpoints with time-to-event data accounting for time-varying treatment efficacy. *Biostatistics* **15,** 251–265.

Gail, M., Pfeiffer, R., Van Houwelingen, H., and Carroll, R. (2000). On meta-analytic assessment of surrogate outcomes. *Biostatistics* **1,** 231–246.

Gilbert, P., Gabriel, E., Huang, Y., and Chan, I. (2015). Surrogate endpoint evaluation: Principal stratification criteria and the prentice definition. *Journal of Causal Inference* **3(2),** 157–175.

Gilbert, P. and Hudgens, M. (2008). Evaluating candidate principal surrogate endpoints. *Biometrics* **64,** 1146–1154.

Gilbert, P., Hudgens, M., and Wolfson, J. (2011). Commentary on “Principal stratification–a goal or a tool?” by Judea Pearl. *The International Journal of Biostatistics* **7,** Article 1.

Gilbert, P., Qin, L., and Self, S. (2008). Evaluating a surrogate endpoint at three levels, with application to vaccine development. *Statistics in Medicine* **27,** 4758–4778. PMCID: PMC2646675.

Huang, Y., Gilbert, P., and Wolfson, J. (2013). Design and estimation for evaluating principal surrogate markers in vaccine trials. *Biometrics* **69,** 301–309.

Joffe, M. and Greene, T. (2009). Related causal frameworks for surrogate outcomes. *Biometrics* **65,** 530–538.

Kobayashi, F. and Kuroki, M. (2014). A new proportion measure of the treatment effect captured by candidate surrogate endpoints. *Statistics in Medicine* **33,** 3338–3353.

Li, Y., Taylor, J., and Elliott, M. (2010). A Bayesian approach to surrogacy assessment using principal stratification in clinical trials. *Biometrics* **66,** 523–531.

Lin, D., Fleming, T., and De Gruttola, V. (1997). Estimating the proportion of treatment effect explained by a surrogate marker. *Statistics in Medicine* **16,** 1515–1527.

Pearl, J. (2001). *Direct and Indirect Effects.* Morgan Kaufmann, San Francisco.

Pearl, J. and Bareinboim, E. (2011). Transportability of causal and statistical relations: A formal approach. *Proceedings of the Twenty-Fifth National Conference on Artificial Intelligence, Menlo Park, CA* pages 247–254.

Prentice, R. (1989). Surrogate endpoints in clinical trials: definition and operational criteria. *Statistics in Medicine* **8,** 431–440.

Robins, J. and Greenland, S. (1992). Identifiability and exchangeability of direct and indirect effects. *Epidemiology* **3,** 143–155.

Taylor, J., Wang, Y., and Thibaut, R. (2005). Counterfactual links to the proportion of treatment effect explained by a surrogate marker. *Biometrics* **61,** 1102–1111.

van der Laan, M. J., Hubbard, A. E., and Pajouh, S. K. (2013). Statistical inference for data adaptive target parameters. *U.C. Berkeley Division of Biostatistics Working Paper Series* page Paper 314.

van der Laan, M. J., Polley, E. C., and Hubbard, A. E. (2007). Super learner. *Statistical Applications in Genetics and Molecular Biology* **6,** number 1.

van der Laan, M. J. and Rose, S. (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data.* Springer, New York.

VanderWeele, T. (2008). Simple relations between principal stratification and direct and indirect effects. *Statistics and Probability Letters* **78,** 2957–2962.

VanderWeele, T. (2013). Surrogate measures and consistent surrogates. *Biometrics* **69,** 561–568.

Wang, Y. and Taylor, J. (2002). A measure of the proportion of treatment effect explained by a surrogate marker. *Biometrics* **58,** 803–812.

Weir, C. and Walley, R. (2006). Statistical evaluation of biomarkers as surrogate endpoints: a literature review. *Statistics in Medicine* **25,** 183–203.

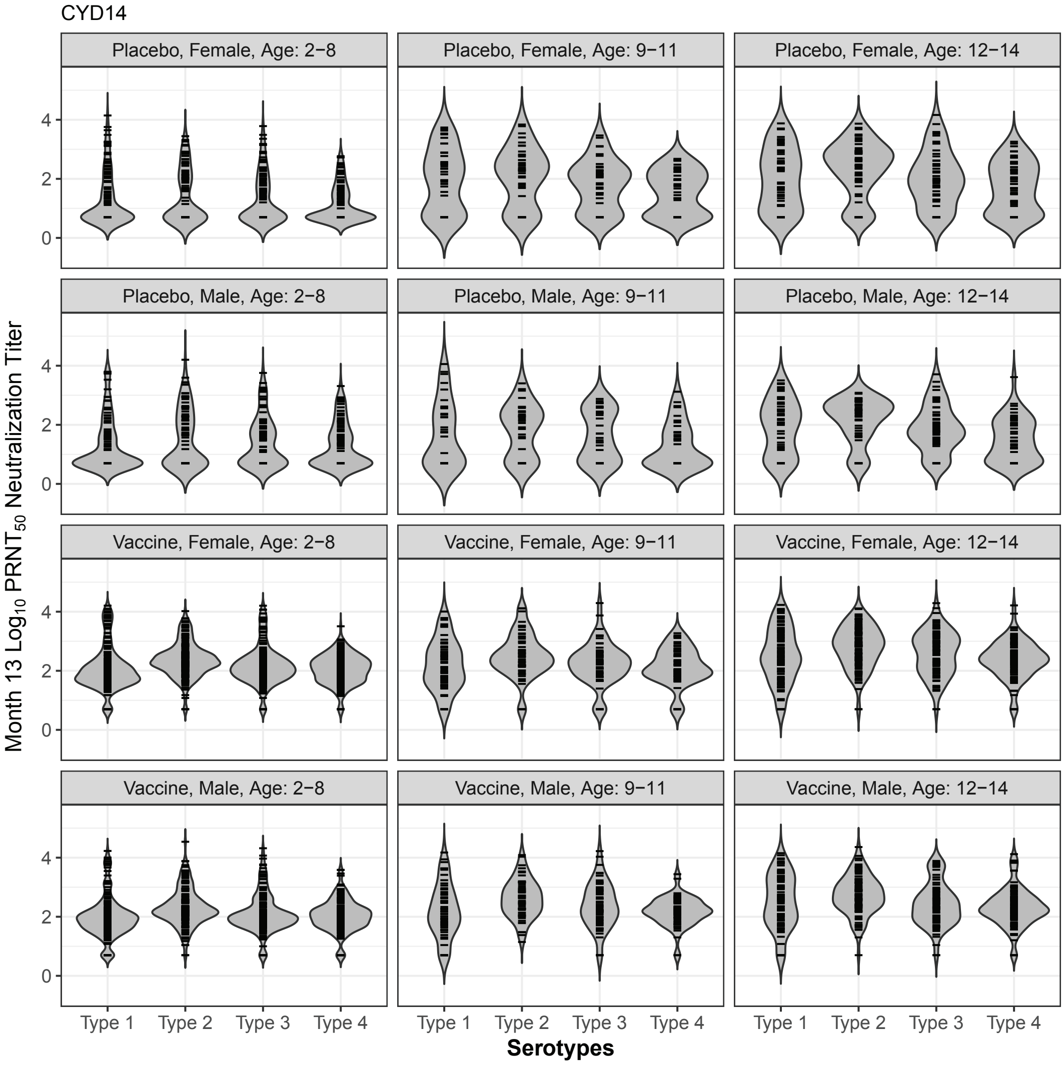


Figure 1. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the $PRNT_{50}$ assay for each of the 4 dengue serotypes by sex and age categories for the CYD14 trial

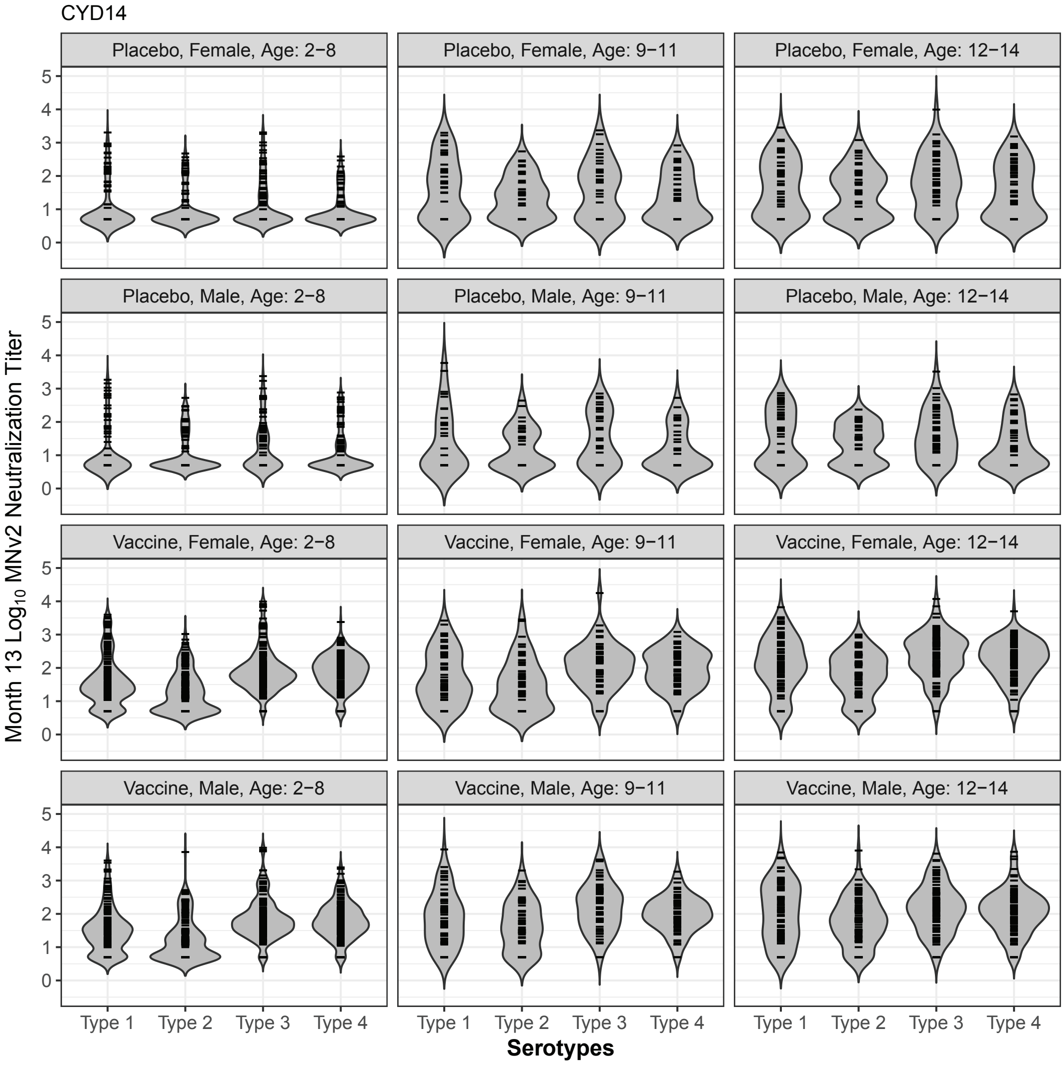


Figure 2. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the Microneutralization Version 2 assay (MNv2) for each of the 4 dengue serotypes by sex and age categories for the CYD14 trial

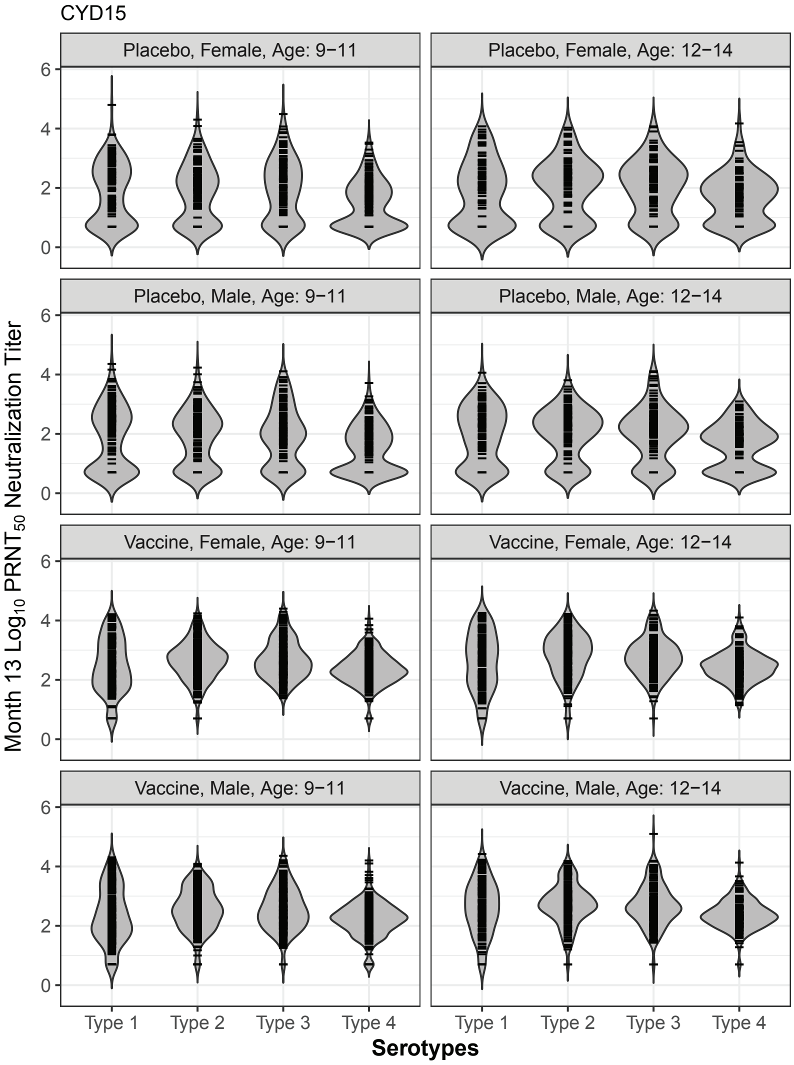


Figure 3. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the $PRNT_{50}$ assay for each of the 4 dengue serotypes by sex and age categories for the CYD15 trial

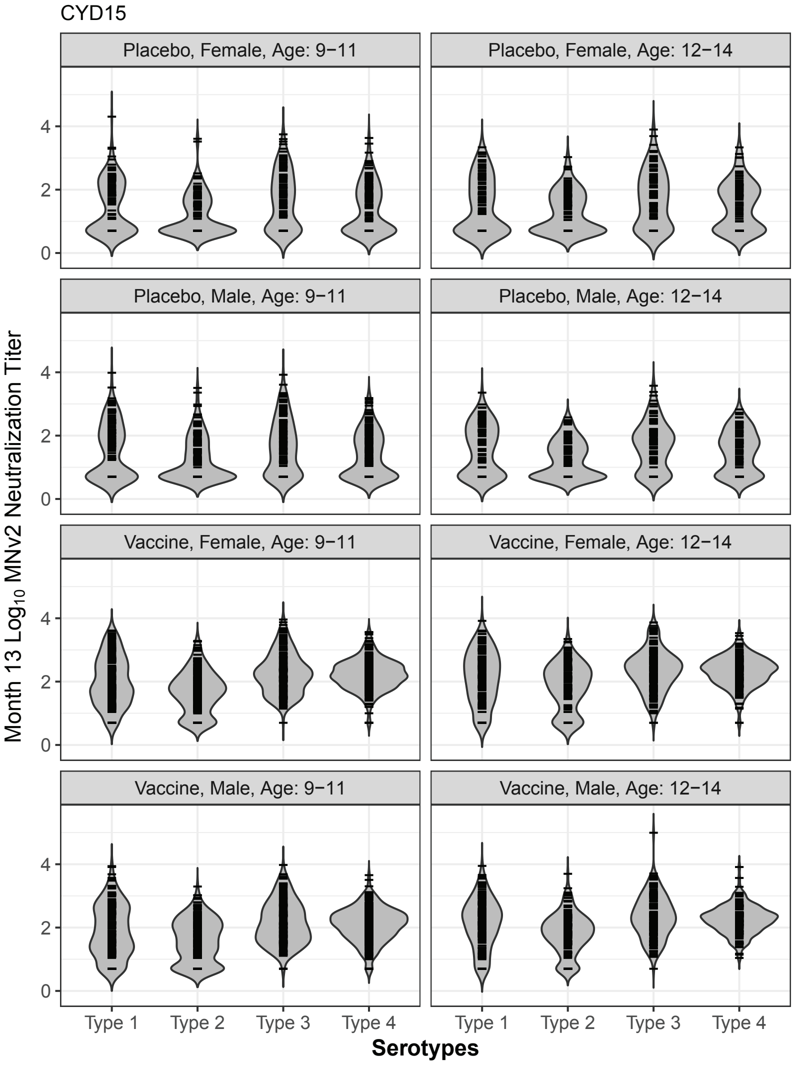


Figure 4. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the Microneutralization Version 2 assay (MNv2) for each of the 4 dengue serotypes by sex and age categories for the CYD15 trial

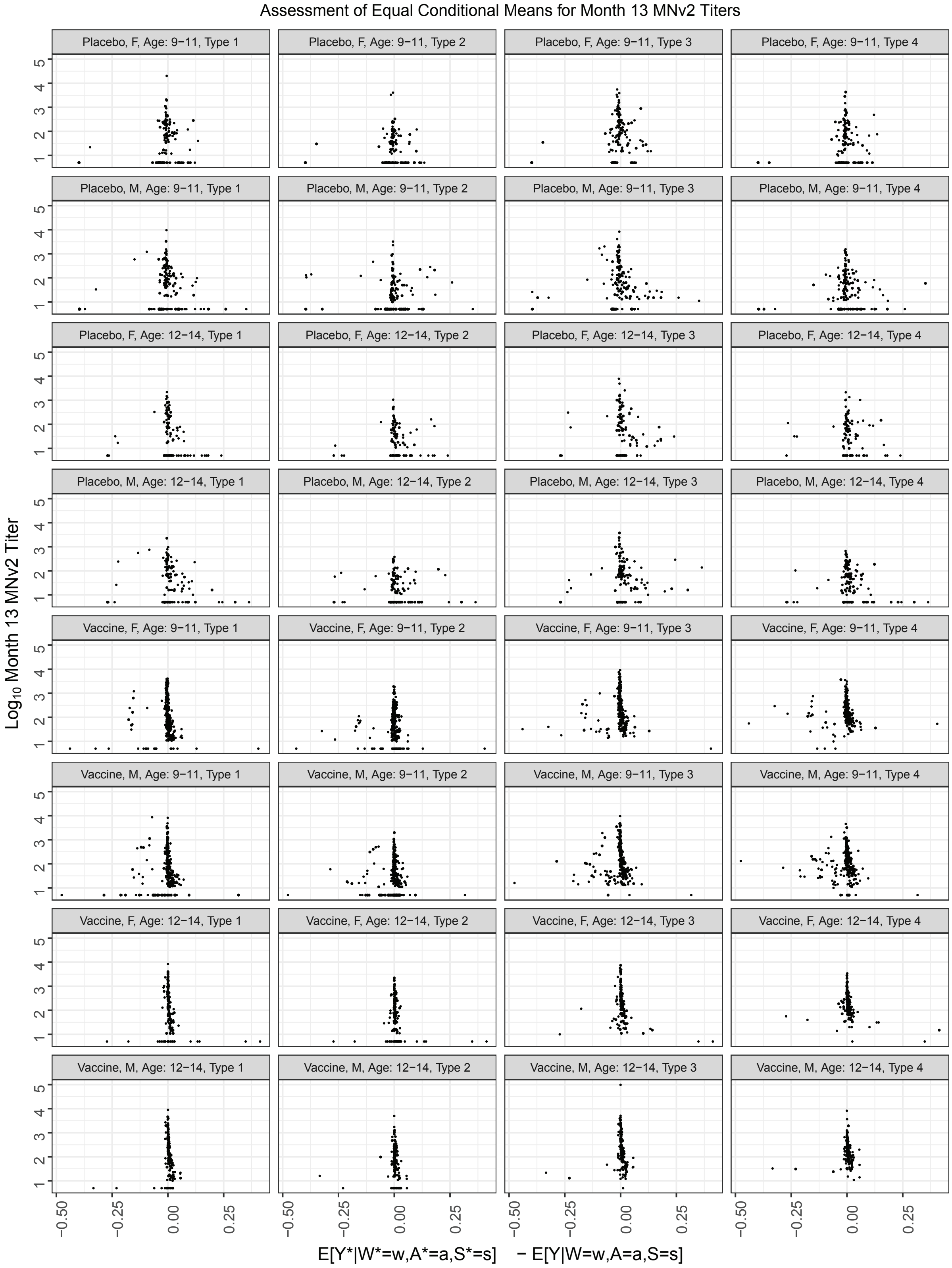


Figure 5. Diagnostics of the Equal Conditional Means assumption (Theorem 2): Plot of the differences (CYD15 - CYD14) in estimated optimal surrogate values for all observed values of CYD15 participants, by covariate categories and month 13 Microneutralization Version 2 titer values.

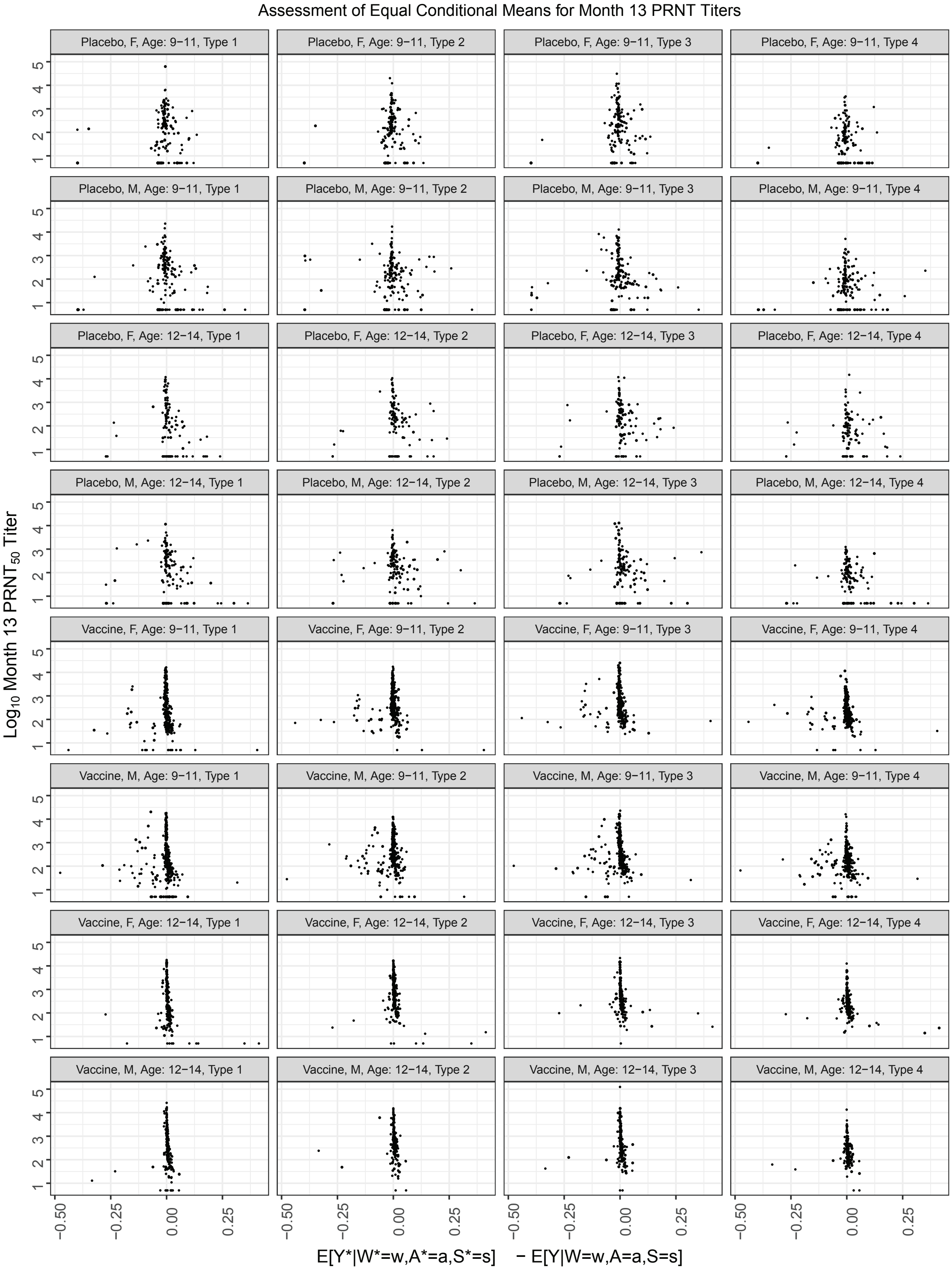


Figure 6. Diagnostics of the Equal Conditional Means assumption (Theorem 2): Plot of the differences (CYD15 - CYD14) in estimated optimal surrogate values for all observed values of CYD15 participants, by covariate categories and month 13 PRNT$_{50}$ neutralization titer values.

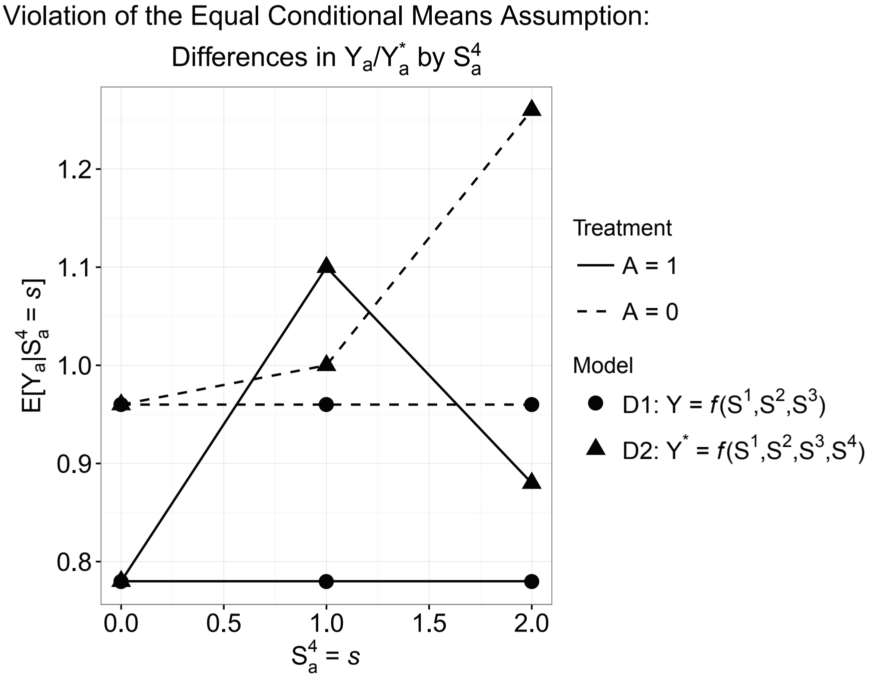


Figure 7. Consider two data sets: D1 in which $Y = f(S^1, S^2, S^3) = \sum_{k=1}^{3} \left[0.1 * k * I(S^k = 1) + I(S^k = 2)\right] + \epsilon_Y$, and D2 in which $Y^* = f(S^1, S^2, S^3, S^4) = \sum_{k=1}^{4} \left[0.1 * k * I(S^{*k} = 1) + I(S^{*k} = 2)\right] + \epsilon_{Y^*}$ where $\epsilon_Y \sim N(0, 0.1^2)$ and $\epsilon_{Y^*} \sim N(0, 0.1^2)$ (as described in Web Appendix G). When comparing the conditional means across values of S^4_a, we see that $E[Y_a|S^4_a = s]$ differs from $E[Y^*_a|S^{*4}_a = s]$ for some values of s (most dramatically for the treatment group $a = 1$ at $s = 1$ and for the control group $a = 0$ at $s = 2$), and thus, the equal conditional means assumption is violated in this example.

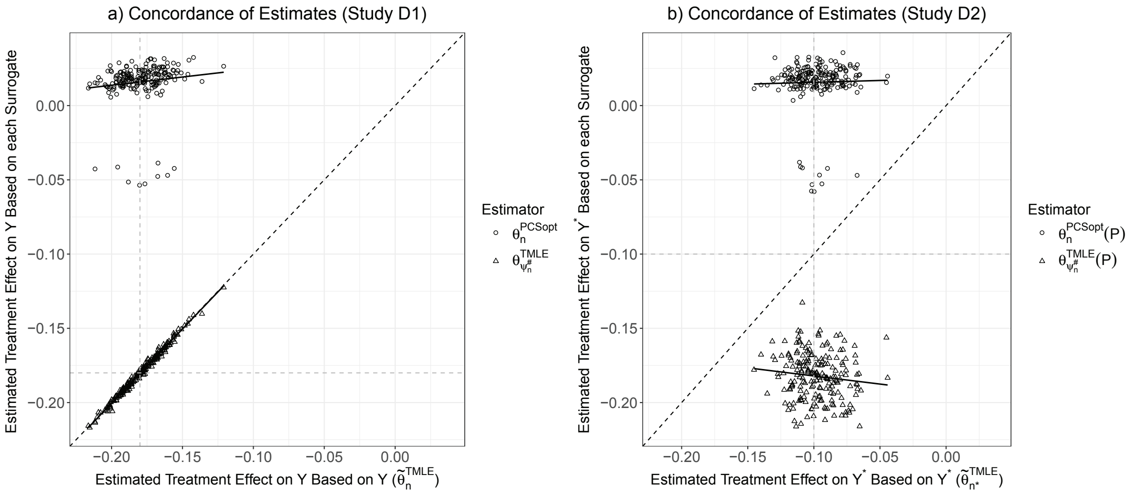


Figure 8. (a) For Simulation 1, estimates of $\theta_0 = E_0(Y_1 - Y_0)$ based on two surrogate endpoint approaches [superlearner-TMLE (SL-TMLE) and proportion of treatment effect captured (PCS)] versus estimates based on sample averages of the clinical endpoints Y. For the PCS method, S^{PCSopt} was selected to be S^1 (the best candidate surrogate, with PCS=0.87) in 191 of 200 (95%) data sets. (b) For Simulation 2, estimates of $\theta_P^* = E_P(Y_1^* - Y_0^*)$ for a second trial D2 based on the two surrogate endpoint approaches with surrogates built from the first trial D1, versus estimates based on sample averages of the clinical endpoints Y^*.

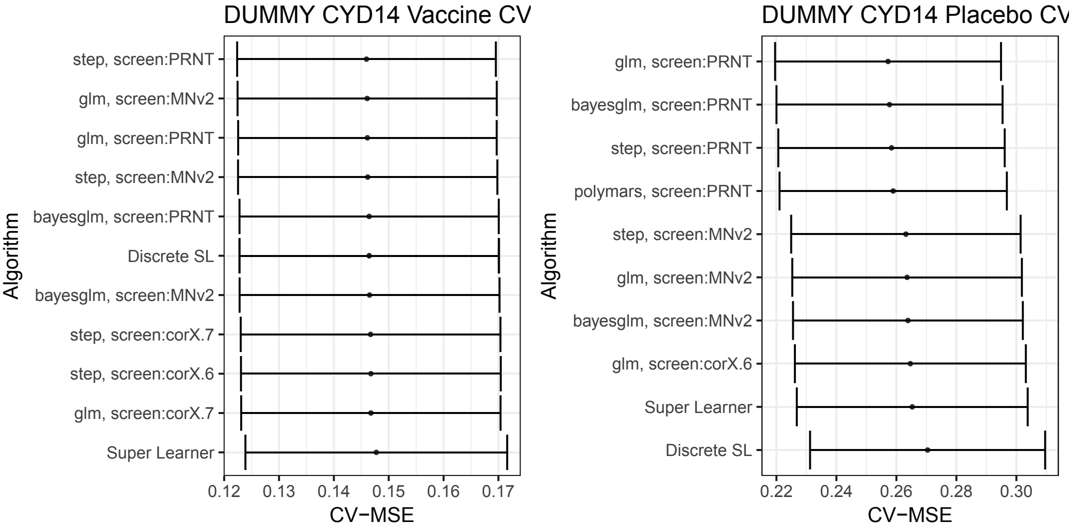


Figure 9. Point and 95% confidence interval estimates of cross-validated mean squared error (CV-MSE) for the vaccine and placebo groups of the CYD14 trial dummy data, for the top 8 performing individual learners, the discrete super-learner, and the super-learner.

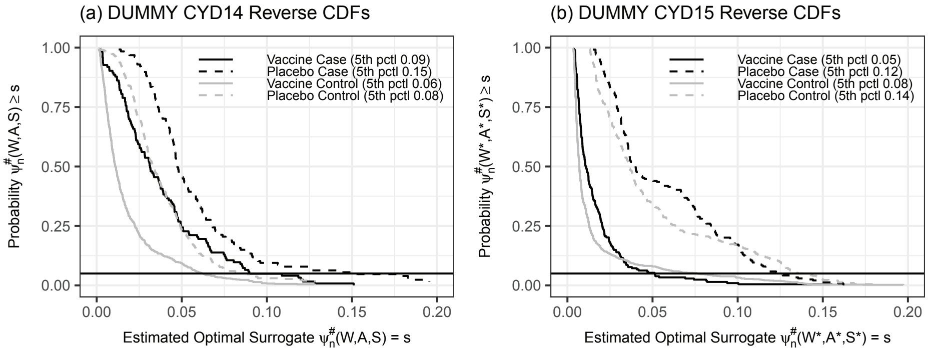
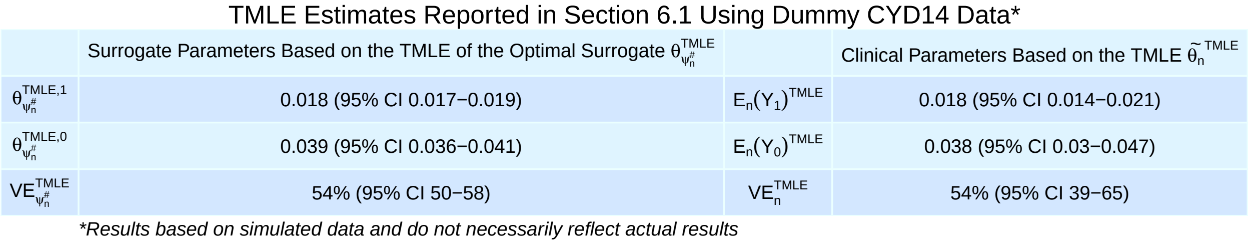


Figure 10. (a) Empirical reverse cumulative distribution functions (cdfs) of the estimated optimal surrogate $\psi_n^{\#}(W_i, A_i = a, S_i)$ for the CYD14 trial dummy data by vaccine/placebo assignment $A = a \in \{0, 1\}$ and dengue outcome case/control status $Y = y \in \{0, 1\}$. (b) Empirical reverse cdfs of $\psi_n^{\#}(W_i^*, A_i^* = a, S_i^*)$ for CYD15 dummy data participants by vaccine/placebo assignment $A^* = a \in \{0, 1\}$ and dengue outcome case/control status $Y^* = y \in \{0, 1\}$, where $\psi_n^{\#}(\cdot)$ was estimated from the CYD14 trial dummy data. The results show that the surrogate better classifies dengue outcomes of participants in the original trial than in the new trial, as expected.

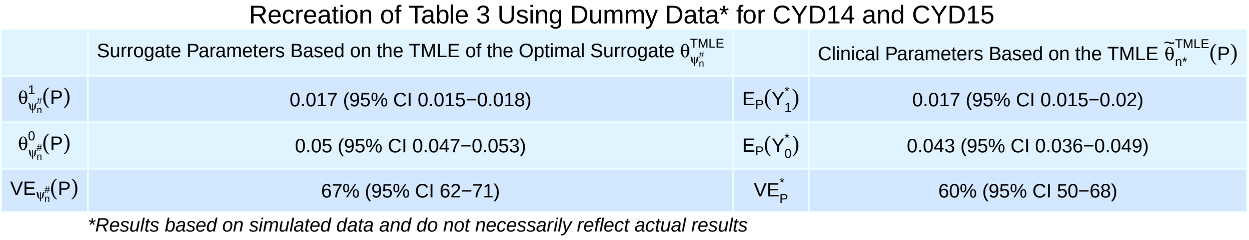


TMLE Estimates Reported in Section 6.1 Using Dummy CYD14 Data*

	Surrogate Parameters Based on the TMLE of the Optimal Surrogate $\theta_{\psi_n^{\#}}^{TMLE}$		Clinical Parameters Based on the TMLE $\tilde{\theta}_n^{TMLE}$
$\theta_{\psi_n^{\#}}^{TMLE,1}$	0.018 (95% CI 0.017−0.019)	$E_n(Y_1)^{TMLE}$	0.018 (95% CI 0.014−0.021)
$\theta_{\psi_n^{\#}}^{TMLE,0}$	0.039 (95% CI 0.036−0.041)	$E_n(Y_0)^{TMLE}$	0.038 (95% CI 0.03−0.047)
$VE_{\psi_n^{\#}}^{TMLE}$	54% (95% CI 50−58)	VE_n^{TMLE}	54% (95% CI 39−65)

**Results based on simulated data and do not necessarily reflect actual results*

Figure 11. TMLE estimates reported in Section 6.1 of the main paper using CYD14 dummy data.



Recreation of Table 3 Using Dummy Data* for CYD14 and CYD15

	Surrogate Parameters Based on the TMLE of the Optimal Surrogate $\theta_{\psi_n^{\#}}^{\mathrm{TMLE}}$		Clinical Parameters Based on the TMLE $\tilde{\theta}_{n^*}^{\mathrm{TMLE}}(P)$
$\theta_{\psi_n^{\#}}^{1}(P)$	0.017 (95% CI 0.015−0.018)	$E_P(Y_1^*)$	0.017 (95% CI 0.015−0.02)
$\theta_{\psi_n^{\#}}^{0}(P)$	0.05 (95% CI 0.047−0.053)	$E_P(Y_0^*)$	0.043 (95% CI 0.036−0.049)
$VE_{\psi_n^{\#}}(P)$	67% (95% CI 62−71)	VE_P^*	60% (95% CI 50−68)

**Results based on simulated data and do not necessarily reflect actual results*

Figure 12. Comparison of inferences on the surrogate parameters $\theta^a_{\psi_n^{\#}}(P) \equiv E_P(E_P(\psi_n^{\#}(W^*, a, S^*) \mid W^*, A^* = a)$ for each $a \in \{0, 1\}$ and $\theta_{\psi_n^{\#}}(P) = VE_{\psi_n^{\#}}(P) = 1 - \theta^1_{\psi_n^{\#}}(P)/\theta^0_{\psi_n^{\#}}(P)$ based on $(W^*, A^*, \psi_n^{\#}(W^*, A^*, S^*))$ versus inferences on the clinical dengue endpoint parameters $E_P(Y_a^*)$ and $\theta_P^* = VE_P^* = 1 - E_P(Y_1^*)/E_P(Y_0^*)$ from $(W^*, A^*, Y^*))$ for CYD15 dummy data.

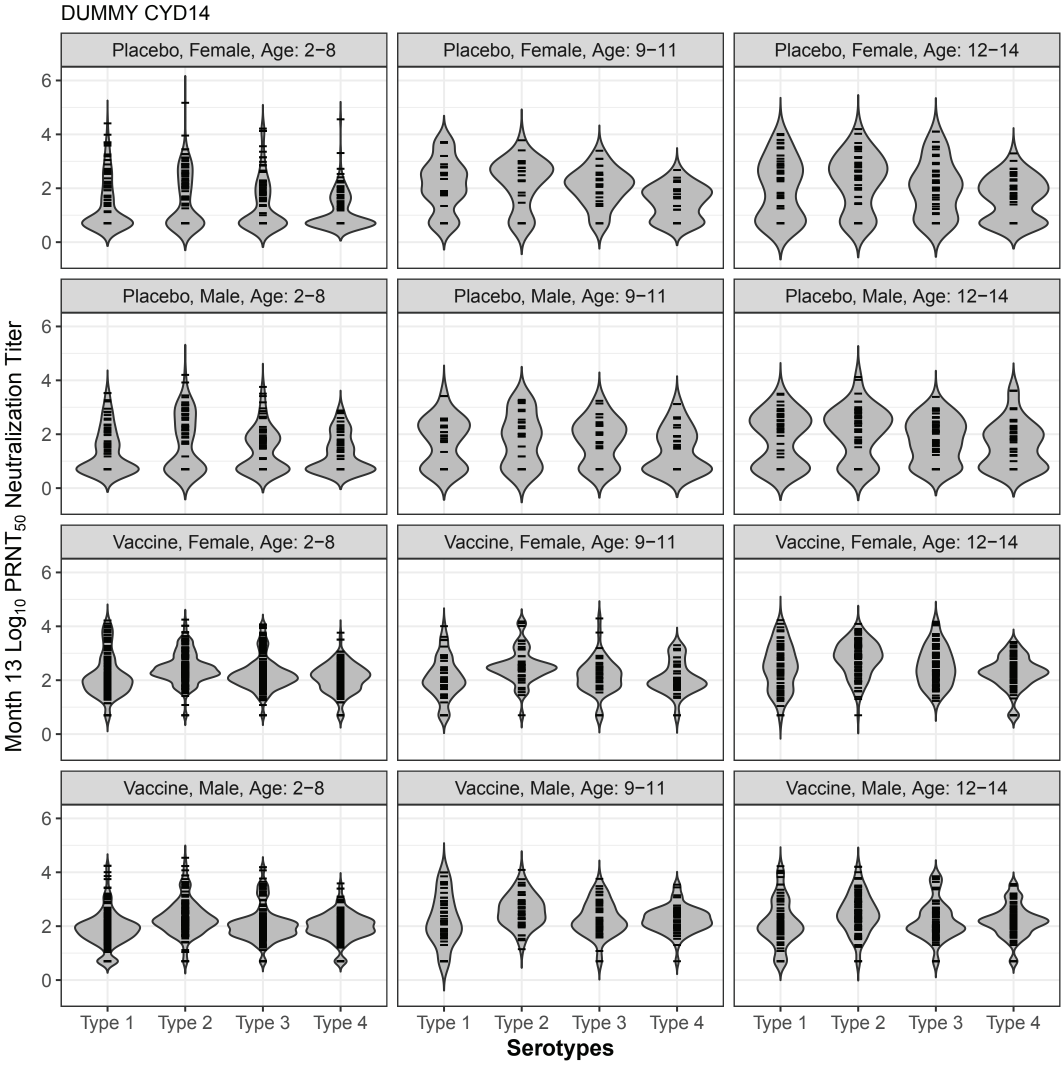


Figure 13. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the $PRNT_{50}$ assay for each of the 4 dengue serotypes by sex and age categories for the CYD14 trial dummy data set

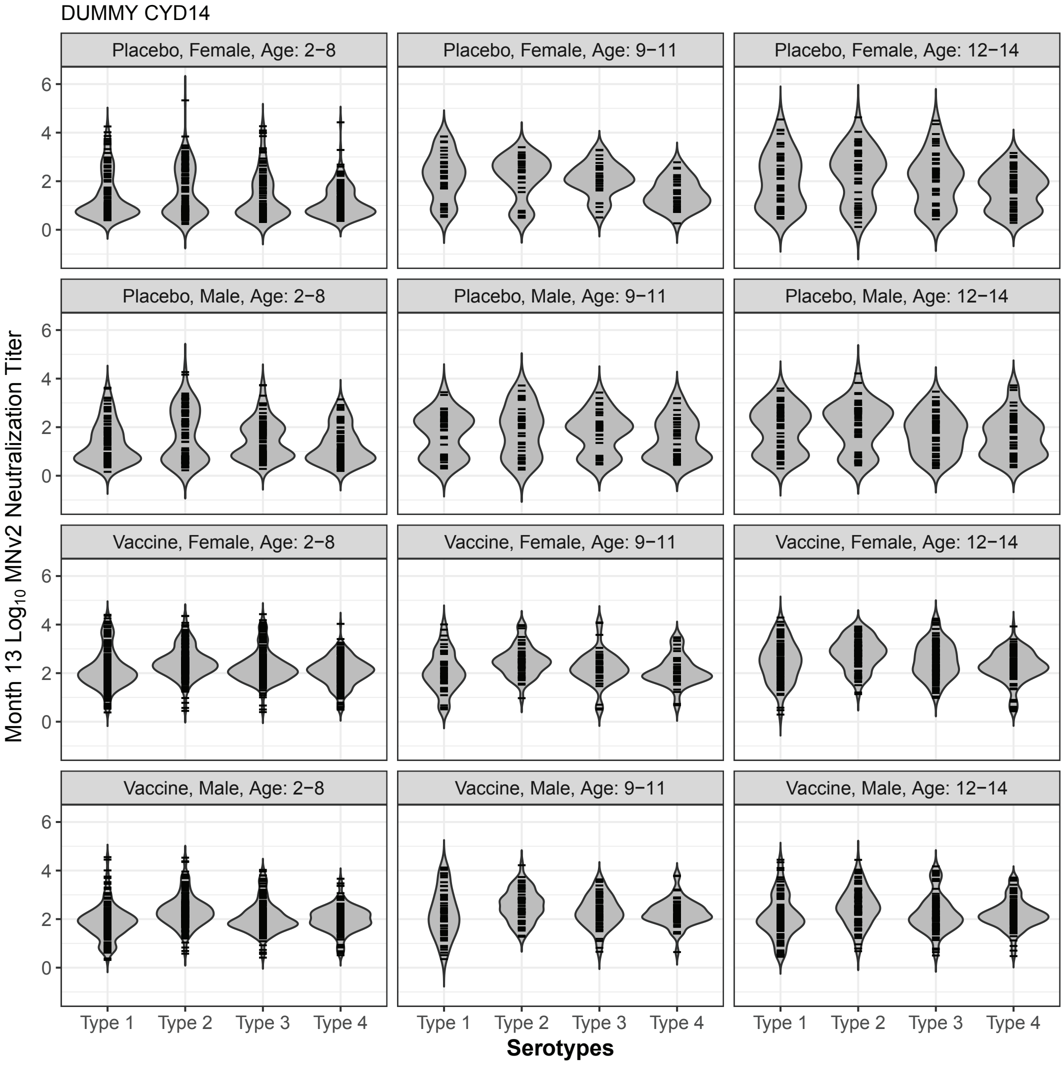


Figure 14. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the Microneutralization Version 2 assay (MNv2) for each of the 4 dengue serotypes by sex and age categories for the CYD14 trial dummy data set

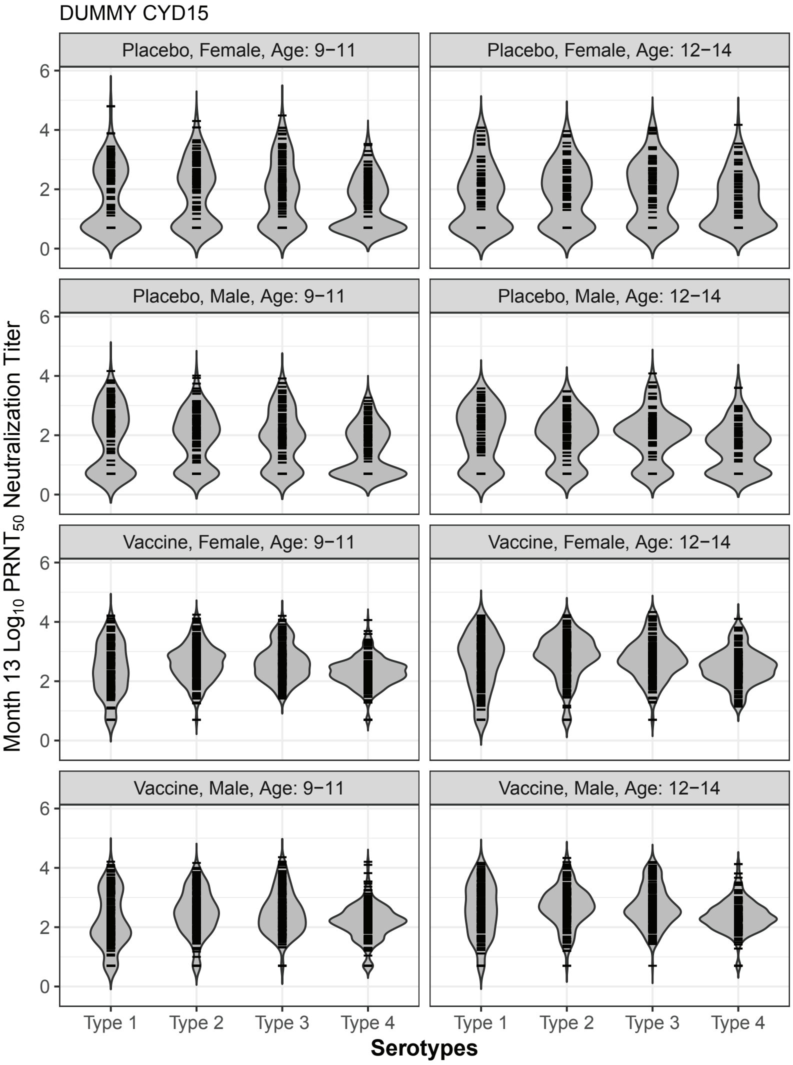


Figure 15. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the $PRNT_{50}$ assay for each of the 4 dengue serotypes by sex and age categories for the CYD15 trial dummy data set

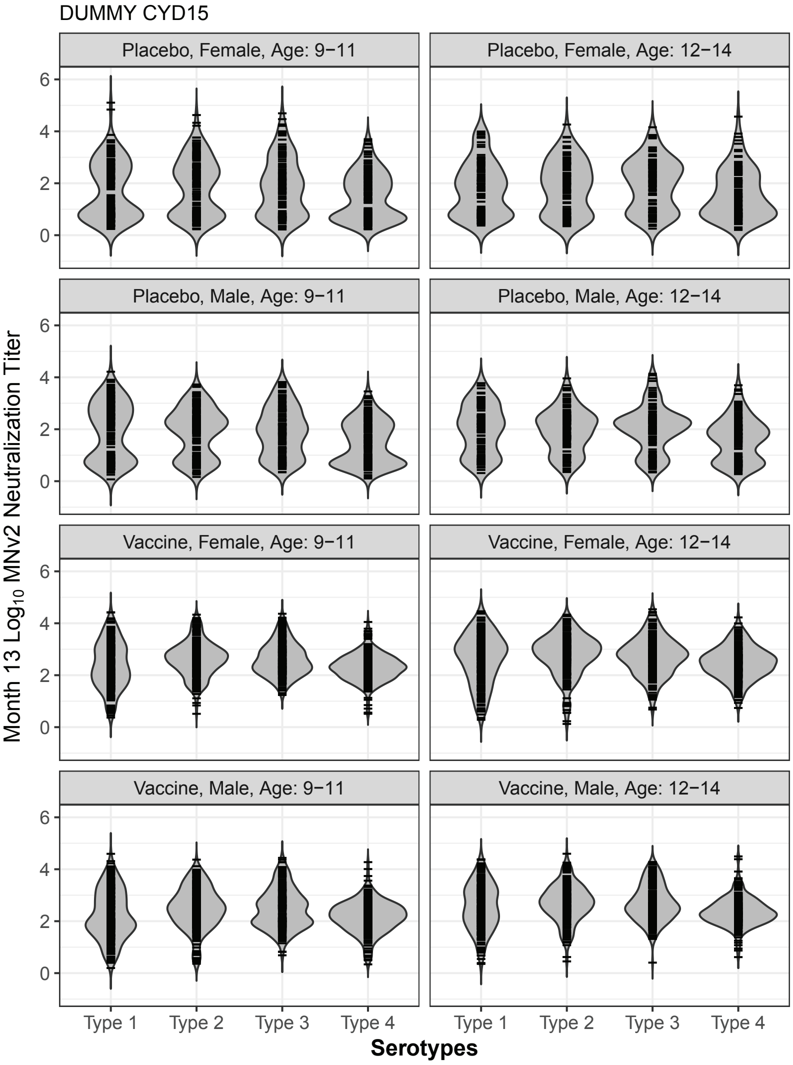


Figure 16. Distributions of log_{10} Month 13 neutralizing antibody titers measured by the Microneutralization Version 2 assay (MNv2) for each of the 4 dengue serotypes by sex and age categories for the CYD15 trial dummy data set

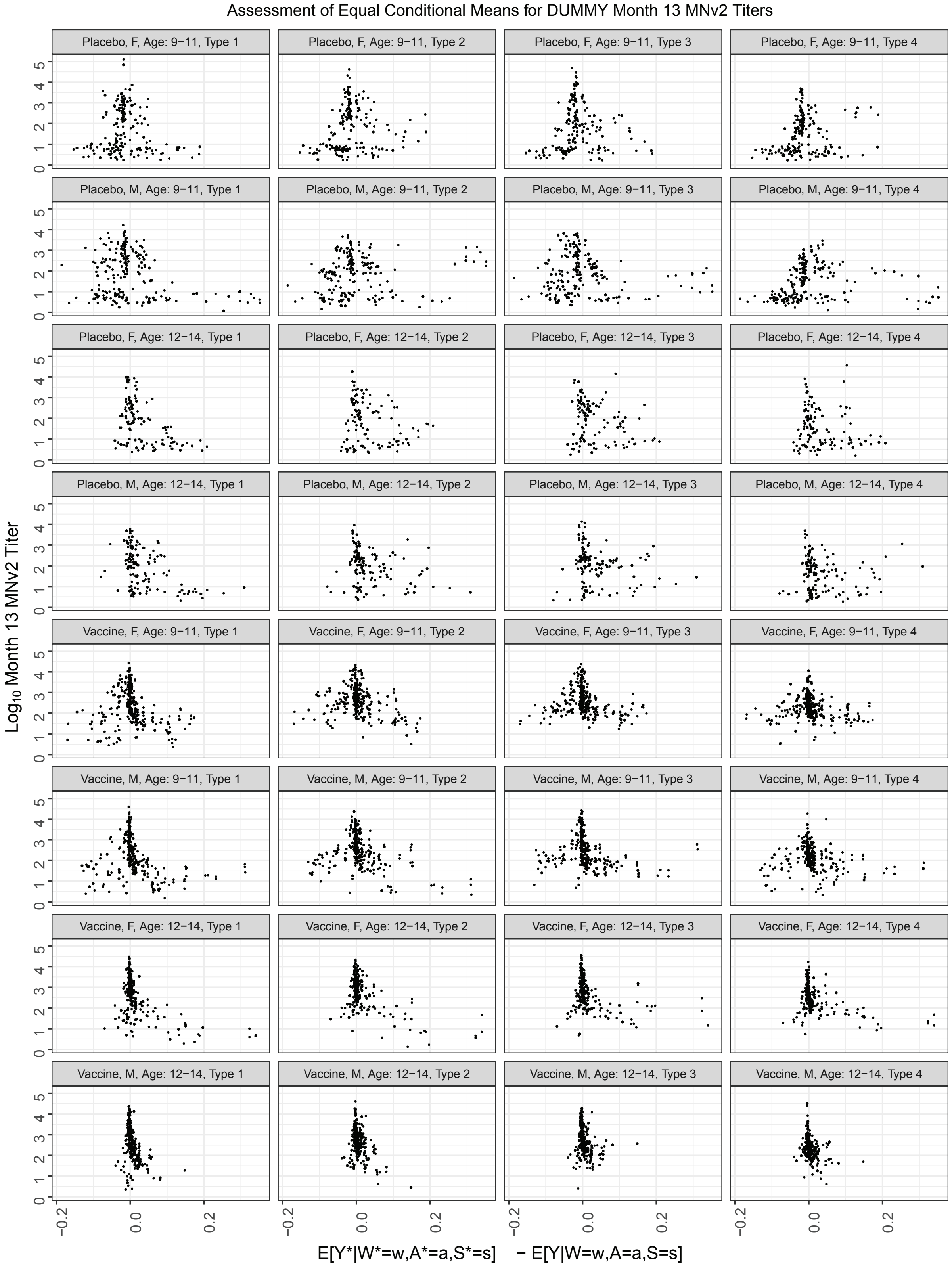


Figure 17. Diagnostics of the Equal Conditional Means assumption (Theorem 2): Plot of the differences (CYD15 - CYD14) dummy data in estimated optimal surrogate values for all observed values of CYD15 dummy data participants, by covariate categories and month 13 Microneutralization Version 2 titer values.

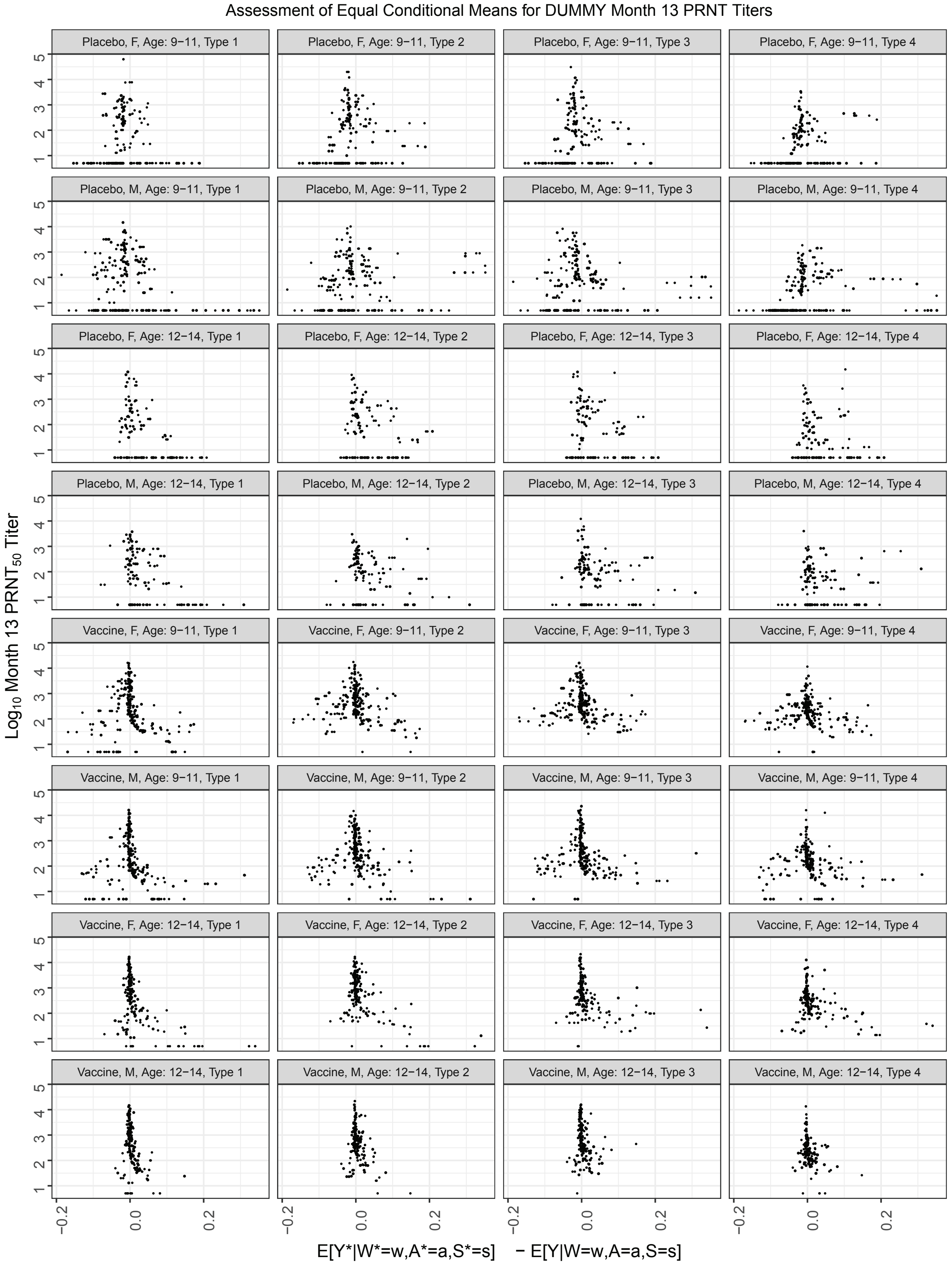


Figure 18. Diagnostics of the Equal Conditional Means assumption (Theorem 2): Plot of the differences (CYD15 - CYD14) dummy data in estimated optimal surrogate values for all observed values of CYD15 dummy data participants, by covariate categories and month 13 PRNT$_{50}$ neutralization titer values.