eAPPENDIX

Using Auxiliary Proxy Characteristics in Analysis

The sensitivity and specificity of $Y*$ may depend on proxy characteristics that are not of scientific interest, but are auxiliary variables that may help to correctly specify the model. Let C denote a low-dimensional proxy characteristic. If $\boldsymbol{Z}$ is low dimensional, then first, estimate $P(Y* = y* \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$. Next, use Eq. (4) to determine sensitivity $P(Y* = 1 \mid Y = 1, X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$, now denoted $Sens(x, \boldsymbol{z}, c)$, and specificity $P(Y* = 0 \mid Y = 0, X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$, now denoted $Spec(x, \boldsymbol{z}, c)$. Plug estimates of $P(Y* = y* \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$, $Sens(x, \boldsymbol{z}, c)$, and $Spec(x, \boldsymbol{z}, c)$ into Eq. (5) to estimate $P(Y = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$. Lastly, calculate $P(Y = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, R = 0) = \sum_c P(Y* = y* \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0) \times P(C = c \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, R = 0)$, and plug the result into Eq. (1). If $\boldsymbol{Z}$ is high dimensional, one would estimate $P(Y* = y* \mid X = x, S = s, C = c, R = 0)$ using propensity-score stratification and follow the same general procedure just described, replacing $\boldsymbol{Z}$ and $\boldsymbol{z}$ with S and s, respectively.

When using multiple imputation, C can be low- or high-dimensional. In this case, the steps to multiple imputation are:

1. Sample participants with $R = 0$ with replacement (bootstrap sample).
2. Estimate parameters of $P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$ (e.g., by logistic regression) using the bootstrapped sample.
3. Calculate fitted values of $P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$ using the original participants with $R = 0$.

4. Plug fitted values of $P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$ into Eq. (5) to estimate $P(Y = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$.

5. For each participant with $R = 0$, draw a value from $P(Y = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, C = c, R = 0)$.

Estimation using the completed datasets is unchanged.

Selecting q_{sens} and q_{spec} for Sensitivity Analysis

We describe three approaches for sensitivity analysis and their advantages and disadvantages: 1) bounded analysis, 2) plausible median sensitivity and specificity, and 3) automated grid analysis. A bounded analysis is performed by first assuming the lower and upper bounds of sensitivity and specificity, respectively, then by assuming the upper and lower bounds of sensitivity and specificity, respectively. By Eq. (3) in the paper, the lower bound of sensitivity is $P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, R = 0)$, and the upper bound of specificity is 1 ($q_{sens} \approx 0, q_{spec} = \infty$); the upper bound of sensitivity is 1 and the lower bound of specificity is $1 - P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, R = 0)$ ($q_{sens} = \infty, q_{spec} \approx 0$). Zero and ∞ can be approximated by arbitrarily small and large values, respectively, such as 0.01 and 99. Bounded analysis has the advantages of requiring calculations for only two sets of assumptions and producing a range of values guaranteed to cover the truth, but it has the disadvantages of likely including implausible values and often producing a range of results so broad that findings are inconclusive. An analysis based on plausible median sensitivity and specificity, described in the next paragraph, can overcome these disadvantages.

Researchers performing an analysis based on plausible median sensitivity and specificity use their background knowledge to first select what they consider to be the most plausible values of sensitivity and specificity. Denote these values as

$Sens_{target}$ and $Spec_{target}$, which can be based, for example, on published validation studies. Next calculate the median value of $P(Y* = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z}, R = 0)$. Denote this value as π^*. Now, using Eq. (4) in the paper, we have

$$Sens_{target} = \frac{\pi^* e^{q_{sens}}}{(1 - \pi^*) + \pi^* e^{q_{sens}}}$$

and

$$Spec_{target} = \frac{(1 - \pi^*) e^{q_{spec}}}{(1 - \pi^*) e^{q_{spec}} + \pi^*}.$$

Solving the equations gives $q_{sens} = \text{logit}(Sens_{target}) - \text{logit}(\pi^*)$ and $q_{spec} = \text{logit}(Spec_{target}) + \text{logit}(\pi^*)$. These values ensure that the median values of sensitivity and specificity meet the targeted values. That is, misclassification can still be differential, where the midpoints of the sensitivity and specificity distributions are at the targeted values.To perform a sensitivity analysis, researchers select a plausible *range* of $Sens_{target}$ and $Spec_{target}$, which they can use to calculate ranges of q_{sens} and q_{spec}. This analysis has the advantage of being informed by scientific background information, but has the disadvantage of being subjective, where readers may have differing opinions from the researchers about which values of sensitivity and specificity are considered plausible.

An automated grid analysis overcomes the limitations of both bounded analysis and an analysis based on plausible median values. In an automated grid analysis, the researcher would specify the number of values of q_{sens} and q_{spec} to consider, denoted n_{sens} and n_{spec}, along with conservative lower and upper bounds (e.g., 0.01 and 99), and produce two vectors of equally spaced values from the lower to the upper bounds. These two vectors form an $n_{sens} \times n_{spec}$ grid consisting of all the combinations of q_{sens} and q_{spec} in their respective vectors. Next, the researcher would analyze the data for all of the grid points and report the results using, for example, contour plots like those in the Figure in the paper. This analysis has the advantage of exploring the full range of q_{sens} and q_{spec}, including bounds and plausible values. A drawback is that this method can be time consuming and computationally intensive. For example, setting $n_{sens} = n_{spec} = 10$ leads to 100 analyses. Furthermore, not all readers are familiar with contour plots, so the results may be difficult to communicate.

SAS Program for Low-Dimensional Covariates

This program estimates p_x with low-dimensional covariates. The data consist of Y, $Y*$ (Ystar in the program), R, X, and a single categorical covariate Z. Values of q_{sens} and q_{spec} (qsens and qspec in the program) are assumed known and varied by the analyst for sensitivity analysis. The PROC NLMIXED module estimates $P(Y = 1 \mid X = x, \boldsymbol{Z} = \boldsymbol{z})$ and is followed by additional code to estimate p_x and $p_1 - p_0$.

```
PROC SORT data=data1; by Z; run;
PROC NLMIXED data=data1;
   parms  pR_0z=0.5 /*P(R=1|X=0, Z=z)*/
          pR_1z=0.5   /*P(R=1|X=1, Z=z)*/
          pYstar_0z0=0.5 /*P(Ystar=1|X=0, Z=z, R=0)*/
          pYstar_1z0=0.5 /*P(Ystar=1|X=1, Z=z, R=0)*/
          pY_0z1=0.5 /*P(Y=1|X=0, Z=z, R=1)*/
          pY_1z1=0.5 /*P(Y=1|X=1, Z=z, R=1)*/;
   by  Z;
   qsens=0.5; qspec=1.75;  /*assumed values*/

   *P(Ystar=1|Y=1, X=0, Z=z, R=0);
```

```
    sens_0z0=pYstar_0z0*exp(qsens)/
             (pYstar_0z0*exp(qsens) + (1-pYstar_0z0));
*P(Ystar=0|Y=0, X=0, Z=z, R=0);
    spec_0z0=(1-pYstar_0z0)*exp(qspec)/
             (pYstar_0z0 + (1-pYstar_0z0)*exp(qspec));
*P(Y=1|X=0, Z=z, R=0);
    pY_0z0=(pYstar_0z0 + spec_0z0 - 1)/(sens_0z0 + spec_0z0 - 1);
*P(Ystar=1|Y=1, X=1, Z=z, R=0);
    sens_1z0=pYstar_1z0*exp(qsens)/
             (pYstar_1z0*exp(qsens) + (1-pYstar_1z0));
*P(Ystar=0|Y=0, X=1, Z=z, R=0);
    spec_1z0=(1-pYstar_1z0)*exp(qspec)/
             (pYstar_1z0 + (1-pYstar_1z0)*exp(qspec));
*P(Y=1|X=1, Z=z, R=0);
    pY_1z0=(pYstar_1z0 + spec_1z0 - 1)/(sens_1z0 + spec_1z0 - 1);
pY_0z=pY_0z0*(1-pR_0z) + pY_0z1*pR_0z  /*P(Y=1|X=0, Z=z)*/;
pY_1z=pY_1z0*(1-pR_1z) + pY_1z1*pR_1z  /*P(Y=1|X=1, Z=z)*/;
if X=0 and R=0 then
   loglik=log(1-pR_0z) + Ystar*log(pYstar_0z0) +
          (1-Ystar)*log(1-pYstar_0z0);
else if X=1 and R=0 then
   loglik=log(1-pR_1z) + Ystar*log(pYstar_1z0) +
          (1-Ystar)*log(1-pYstar_1z0);
else if X=0 and R=1 then
    loglik=log(pR_0z) + Y*log(pY_0z1) + (1-Y)*log(1-pY_0z1);
   else
       loglik=log(pR_1z) + Y*log(pY_1z1) + (1-Y)*log(1-pY_1z1);
   model Y~general(loglik);
   estimate "P(Y=1|X=0, Z=z)" pY_0z;
   estimate "P(Y=1|X=1, Z=z)" pY_1z;
   ods output AdditionalEstimates=data2; run;

*Additional code for obtaining standardized proportions and difference;
ODS LISTING CLOSE;
PROC FREQ data=data1; Table Z; ods output OneWayFreqs=data3; run;
DATA data4; merge data2 data3; by  Z;
   var_zprop=(StandardError*Percent/100)**2;
   mean_zprop=Estimate*Percent/100;
   run;
PROC SORT data=data4; by Label; run;
PROC MEANS data=data4; by Label; var mean_zprop var_zprop;
   output out=data5; run;
DATA data6; set data5;
   SE = sqrt(var_zprop*_FREQ_); phat=mean_zprop*_FREQ_;
   if _STAT_ ~="MEAN" then delete; run;
DATA data7; set data6;
   if Label="P(Y=1|X=1, Z=z)" then Label="Xeq1"; else Label="Xeq0";
   keep  Label phat SE; run;
PROC TRANSPOSE data=data7 out=data8 prefix=phat;
   id Label; var phat; run;
```

```
PROC TRANSPOSE data=data7 out=data9 prefix=SE_; id Label; var SE; run;
DATA data10; merge data8(drop=_name_) data9(drop=_name_);
   phatdiff=phatXeq1 - phatXeq0;
   SEdiff=sqrt(SE_Xeq1**2 + SE_Xeq0**2); run;
ODS LISTING;
PROC PRINT data=data10;
   var phatXeq1  phatXeq0 SE_Xeq1 SE_Xeq0 phatdiff SEdiff; run;
```

SAS Program for Propensity Score Stratification

This program provides necessary first steps to perform propensity-score stratification to estimate the pattern-mixture model. The data consist of Y, $Y*$ (Ystar in the program), R, X, ID, and three covariates (of any kind) Z_1, Z_2, and Z_3 (Z1, Z2, and Z3 in the program). Once again, values of q_{sens} and q_{spec} (qsens and qspec in the program) are assumed known and varied by the analyst to perform sensitivity analysis. The code below calculates the propensity score stratum, S. Setting $Z = S$ in a DATA step, naming the data 'data1', and implementing the complete program for low-dimensional covariates above will produce estimates of p_x and $p_1 - p_0$.

```
PROC SORT data=data1;by ID;run;
PROC LOGISTIC  data=data1; model X(event='1') = Z1 Z2 Z3;
   output out=data2 p=PS1 /*PS for X=1*/; run;
PROC RANK data=data2 groups=5 /*quintiles*/ out=data3(keep=ID S);
   var PS1; ranks S /*S=stratum*/; run;
DATA data4; merge data2 data3; by ID; run;
```

SAS Macro for Multiple Imputation

This macro performs multiple imputation to estimate the pattern-mixture model. The data are the same as that for propensity-score stratification. The quantities qsens and qspec are specified when calling the macro and varied by the analyst to perform sensitivity analysis. Inverse probability weighting with the completed datasets is used to estimate p_x and $p_1 - p_0$.

```
*mm=number of imputations, numcovs=number of covariates Z;
%MACRO proxy_mi_macro(mm, qsens,qspec,seed,numcovs);
ODS LISTING CLOSE;
PROC SORT data=data1; by  ID; run;
PROC LOGISTIC  data=data1;
   model X(event='1') = Z1-Z&numcovs;
   output out=data2 p=PS1 /*PS for X=1*/; run;
DATA data3; set data2;
   if R=1 then delete; keep ID R Ystar X Z1-Z&numcovs PS1; run;
DATA data4; set data2; if R=0 then delete; keep ID  R Y X PS1; run;
DATA data5; set data4; DO i=1 to &mm; replicate=i;
   output; end; drop i; run;
DATA data6; set data3; DO i=1 to &mm; replicate=i;
   output; end; drop i; run;
PROC SORT data=data6; by replicate; run;
PROC SURVEYSELECT data=data3 out=data7 seed=&seed method=urs samprate=1
   outhits rep=&mm; run; /*data7 = mm bootstrap samples of data3*/
PROC SORT data=data7; by replicate; run;
PROC LOGISTIC data=data7 outmodel=miparms; by  replicate;
```

```
   model Ystar(event=’1’)=X Z1-Z&numcovs;
   score data=data6 out=data8;
   run;/*data8=mm estimates of P(Ystar=1|X=x, Z=z, R=0)*/
DATA data9; set data8; theseed = &seed+replicate;
 *P_1=P(Ystar=1|X=x, Z=z, R=0);
   sens=P_1*exp(&qsens)/(P_1*exp(&qsens) + (1-P_1));
   spec=(1-P_1)*exp(&qspec)/((1-P_1)*exp(&qspec) + P_1);
 *ProbY=P(Y=1|X=x, Z=z, R=0);
   probY = (P_1 + spec -1)/(sens+spec-1);
   Yimpute = ranbin(theseed,1,probY); run;
DATA data10; set  data5 data9;
   if R=1 then Ycomp=Y; else Ycomp=Yimpute;
   ipw = X/PS1 + (1-X)/(1-PS1); _IMPUTATION_ = replicate;
   keep ID _IMPUTATION_ Ycomp X  ipw; run;
PROC SORT data=data10; by _IMPUTATION_ ID; run;
PROC GENMOD data=data10 descending; by  _IMPUTATION_; class ID;
   model Ycomp=X/covb dist=bin link=identity; /*standardization*/
   weight ipw; repeated subject=ID/type=ind;
   ods output GEEEmpPEst=gmparms ParmInfo=gmpinfo CovB=gmcovb; run;
PROC MIANALYZE parms=gmparms covb=gmcovb parminfo=gmpinfo;
   modeleffects intercept X;
 *Xeq0 and Xeq1 are standardized probabilities;
   Xeq0: test intercept; Xeq1: test intercept+X;
   ods output TestParameterEstimates=data11; run;
DATA data12; set data11; phat=Estimate; SE=StdErr;
```

```
   keep Test phat SE; run;
PROC TRANSPOSE data=data12 out=data13 prefix=phat;
   id Test; var phat; run;
PROC TRANSPOSE data=data12 out=data14 prefix=SE_;
   id Test; var SE; run;
DATA data15; merge data13(drop=_name_) data14(drop=_name_);
   phatdiff=phatXeq1 - phatXeq0;
   SEdiff=sqrt(SE_Xeq1**2 + SE_Xeq0**2); run;
ODS LISTING ;
PROC PRINT data=data15;
   var phatXeq1  phatXeq0 SE_Xeq1 SE_Xeq0 phatdiff SEdiff;run;
%MEND proxy_mi_macro;

*mm=20, qsens=1.75, qspec=0.50, seed=9876, numcovs=3;
%proxy_mi_macro(20,0.50,1.75,9876,3);
```