The TwoPhaseInd package: Estimation of gene-treatment interactions in randomized clinical trials exploiting gene-treatment independence

Xiaoyu Wang, James Y. Dai

August 26, 2024

1 Introduction

In randomized clinical trials, there are often ancillary studies with outcome-dependent sampling to identify baseline genetic markers that modify treatment effect. The TwoPhaseInd package implements several methods we developed to estimate gene-treatment interactions in randomized clinical trials, exploiting gene-treatment independence dictated by randomization [3, 2, 4, 5]. Substantial reduction of variance can be achieved by exploiting gene-treatment independence for estimating gene-treatment interaction and subgroup treatment effects. The sampling schemes considered in TwoPhaseInd include case-only design, case-control sampling, and case-cohort sampling. For case-control sampling, TwoPhaseInd provides two functions that compute two estimators- the semiparametric maximum likelihood estimator (SPMLE) and the maximum estimated likelihood estimator (MELE), both can exploit the gene-treatment independence [3]. For case-cohort sampling, it provides a function (acoarm) to estimate parameters in a cox regression model by a multi-step estimation procedure developed for augmented case-only designs [5]. In this document we show examples of applying the functions in the TwoPhaseInd package for various designs and estimators.

2 Case-only design

Case-only design can be used to estimate the gene-treatment interaction and subgroup treatment effects in trials with rare failure events. A function "caseonly" is provided in the package to estimate the treatment effect when biomarker=0 and the interaction between treatment and biomarker.

The inputs of caseonly function - caseonly(data, treatment, BaselineMarker, extra, fraction), include "data", a data frame contains the case-only data; "treatment", "BaselineMarker", and "extra" are the column names of "data" that represent the randomized treatment assignment, the biomarker of interest, and extra variables to be adjusted for respectively; “fraction” defines the randomization fraction of the active treatment assignment.

We show an example of applying the function below. First we load the example dataset:

```
> data(acodata)
> dim(acodata)

[1] 907  14

> str(acodata)

'data.frame':         907 obs. of  14 variables:
 $ vacc1_evinf      : int  1442 1489 913 920 1448 1465 377 1274 1472 1463
 $ f_evinf          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ subcoh           : logi  TRUE FALSE FALSE FALSE TRUE FALSE ...
 $ ptid             : int  9601 9603 9605 9606 9607 9608 9609 9610 9613 96
 $ f_treat          : int  1 1 1 1 0 0 1 1 1 0 ...
 $ fcgr2a.3         : num  0 NA NA NA NA NA NA NA NA NA ...
 $ f_agele30        : int  0 0 0 0 1 0 0 0 1 1 ...
 $ f_hsv_2          : num  0 1 0 0 0 0 0 0 0 0 ...
 $ f_ad5gt18        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ f_crcm           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ any_drug         : num  1 1 1 0 0 0 0 1 0 0 ...
 $ num_male_part_cat: num  0 0 0 1 0 0 0 0 0 0 ...
 $ uias             : num  0 1 1 1 1 0 0 0 0 0 ...
 $ uras             : num  0 0 0 0 1 0 0 0 0 0 ...
```

The data frame “acodata” was derived from the STEP trial [1, 7] to study the interactions between the host immune gene Fc-$gamma$ receptor and vaccine on HIV infection. We will use part of the data for case-only estimation here and later use this data for the augmented case-only estimation. It contains 907 participants and 14 variables. The key variables include “vacc1_evinf”, the time to HIV infection; “f_evinf”, the indicator variable for HIV infection; “subcoh”, the indicator of whether the participant was selected into the sub-cohort for genotyping; “ptid”, the participant identifier; “f_treatment”, the vaccine assignment variable; “fcgr2a.3”, the genotype of Fcγ receptor FcγRIIIa, the biomarker of interest here; the rest of variables are other covariates that can be adjusted for in the model.

We then extract the case-only data, and apply the function to it:

```
> cfit=caseonly(data=acodata[acodata[,2]==1,], ##dataset
+               treatment="f_treat",  ##treatment variable
+               BaselineMarker="fcgr2a.3")  ##biomarker
> cfit
```

```
                                                   beta     stder      pVal
treatment effect when baselineMarker=0  0.6326707 0.4937756 0.2000912
treatment+baselineMarker interaction   -0.2549794 0.3820834 0.5045551
```

The above outputs contain “beta” (the estimated parameter), “stder” (standard error of the estimate), and “pVal” (p-value of the estimate=0) for the treatment effect when biomarker=0 and the interaction between treatment and biomarker.

3 Case-control design

We took a Women’s Health Initiative (WHI) biomarker study to illustrate our methods for case-control sampling. Twenty nine biomarkers were picked by WHI investigators as markers that are possibly associated with either stroke, venous thrombotic disease, or myocardial infarction. A comprehensive analysis of these samples was published by [6]. The results of this particular biomarker example using our methods were also shown in [3]. The methodologies for estimating SPMLE and MELE can be found in [3].

3.1 SPMLE

The spmle function computes semiparametric likelihood estimate for a logistic model under case-control sampling, using or not using gene-treatment independence. The latter is mostly pedagogical to show the efficiency gain of using the independence.

The inputs of spmle function - spmle(data, response, treatment, BaselineMarker, extra, phase, ind, ...), include “data”, a data frame to store all the input data; “response”, “treatment”, “BaselineMarker”, and “extra” are the column names of “data” that represent response variable, the randomized treatment assignment, the biomarker of interest, and extra variables to be adjusted for respectively; “phase” is the column name of phase indicator; “ind” is a logical flag (TURE or FALSE) to indicate if incorporating the independence between the randomized treatment and biomarker.

We illustrate a few examples of applying spmle below. First we load the example dataset:

```
> data(whiBioMarker)
> dim(whiBioMarker)
```

```
[1] 16608    10
```

```
> str(whiBioMarker)
```

```
'data.frame':        16608 obs. of  10 variables:
 $ stroke : num  0 0 0 0 0 0 1 0 0 1 ...
 $ hrtdisp: num  1 1 0 1 1 1 1 1 0 1 ...
 $ papbl  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ age    : num  64 62 62 60 54 57 77 68 73 64 ...
 $ dias   : num  74 70 70 79 70 88 62 60 60 67 ...
 $ hyp    : Factor w/ 3 levels "Missing","No",..: 2 2 2 2 3 3 2 2 2 2 ...
 $ syst   : num  116 135 133 133 119 ...
 $ diabtrt: Factor w/ 3 levels "Missing","No",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ lmsepi : Factor w/ 5 levels "2 - <4 episodes per week",..: 5 4 1 4 1 5 5
 $ phase  : num  1 1 1 1 1 1 1 1 1 1 ...
```

The example dataset “whiBioMarker” was used in WHI hormone trial to study the interaction between biomarker and hormone therapy (estrogen plus progestin) on stroke. It contains 10 variables and 16608 participants. The key variables include “stroke”, the response variable for whether the participant have stroke; “hrtdisp”, the hormone treatment variable; “papbl”, the plasmin-antiplasmin complex, the biomarker example here; “age”, the age of a participant; “dias”, diastolic blood pressure; “hyp”, whether the participant have hypertension; “syst”, systolic blood pressure; “diabtrt”, whether the participant have diabetes; “lmsepi”, physical activity per week of a participant; “phase”, the indicator if the biomarker been measured on an applicant (1: not measured, 2: measured. Usually it is expensive to measure biomarkers, and they are measured only on some applicants).

Here is an example code for estimating SPMLE without exploiting independent and with several covariates included in the model:

```
> spmleNonIndExtra <- spmle(data=whiBioMarker,  ## dataset
+                response="stroke",  ## response variable
+                treatment="hrtdisp",        ## treatment variable
+                BaselineMarker="papbl",        ## biomarker
+                extra=c(
+                        "age"
+                         , "dias"
+                         , "hyp"
+                         , "syst"
+                         , "diabtrt"
+                         , "lmsepi"
+                             ),        ## extra variable(s)
+                phase="phase",        ## phase indicator
+                ind=FALSE        ## independent or non-indepentent
+ )
> spmleNonIndExtra

                              beta  stder         pVal
(Intercept)                -3.9599 0.6756 4.602982e-09
hrtdisp (Treatment)         0.3698 0.1599 2.071078e-02
papbl (BaselineMarker)      2.3487 1.0565 2.620678e-02
hrtdisp:papbl              -4.1924 1.3313 1.637308e-03
age                         1.3736 1.1935 2.497868e-01
```

```
dias                       -0.8499 0.9990 3.949167e-01
hypNo                      -0.7751 0.6229 2.133320e-01
hypYes                     -0.7607 0.6288 2.263832e-01
syst                        3.3370 1.2286 6.603730e-03
diabtrtYes                  0.8811 0.3707 1.746453e-02
lmsepi4+ episodes per week  0.0022 0.3927 9.954563e-01
lmsepiMissing              -0.1904 0.6121 7.557121e-01
lmsepiNo activity           0.3231 0.4145 4.356103e-01
lmsepiSome activity         0.0659 0.3522 8.516191e-01
```

The above outputs contain “beta”, “stder”, and “pVal” for the estimated parameters of the model.

Similarly we show an example of estimating SPMLE with exploiting independent and with several covariates included in the model:

```
> spmleIndExtra <- spmle(data=whiBioMarker,        ## dataset
+             response="stroke",        ## response variable
+             treatment="hrtdisp",        ## treatment variable
+             BaselineMarker="papbl",        ## biomarker
+             extra=c(
+                "age"
+                    , "dias"
+               , "hyp"
+               , "syst"
+               , "diabtrt"
+               , "lmsepi"
+                  ),        ## extra variable(s)
+             phase="phase", ## phase indicator
+             ind=TRUE ## independent or non-indepentent
+ )
> spmleIndExtra

                              beta  stder         pVal
(Intercept)                -3.9647 0.6734 3.923845e-09
hrtdisp (Treatment)         0.3102 0.1467 3.440407e-02
papbl (BaselineMarker)      1.9058 0.9375 4.206694e-02
hrtdisp:papbl              -3.8688 1.1590 8.435224e-04
age                         1.7675 1.2051 1.424797e-01
dias                       -0.6402 0.9864 5.163626e-01
hypNo                      -0.8253 0.6189 1.823383e-01
hypYes                     -0.8161 0.6244 1.911675e-01
syst                        3.0481 1.2110 1.183348e-02
diabtrtYes                  0.9493 0.3715 1.060836e-02
```

```
lmsepi4+ episodes per week  0.1714 0.3879 6.586897e-01
lmsepiMissing              -0.1447 0.6089 8.121264e-01
lmsepiNo activity           0.3950 0.4085 3.336300e-01
lmsepiSome activity         0.1540 0.3488 6.588986e-01
```

3.2 MELE

The mele function computes semiparametric estimated likelihood estimate for a logistic model under case-control sampling, using or not using gene-treatment independence. It is slightly less efficient compared to the SPMLE, with less computation burden.

The inputs of mele function - mele(data, response, treatment, BaselineMarker, extra, phase, ind), are the same as those of spmle. Users need to provide a data frame with column names of response, treatment, biomarker of interest, extra variables, phase indicator. The independence flag indicates if incorporating the independence between the randomized treatment and biomarker.

Here is an example of estimating MELE with exploiting independent and with several covariates included in the model:

```
> melIndExtra <- mele(data=whiBioMarker,        ## dataset
+           response="stroke",        ## response variable
+           treatment="hrtdisp",                ## treatment variable
+           BaselineMarker="papbl",                ## biomarker
+           extra=c(
+              "age"
+               , "dias"
+               , "hyp" ##
+               , "syst"
+               , "diabtrt"
+               , "lmsepi"
+               ),        ## extra variable(s)
+           phase="phase",        ## phase indicator
+           ind=TRUE        ## independent or non-indepentent
+ )
> melIndExtra

                              beta  stder         pVal
(Intercept)                -3.8846 0.7172 6.089906e-08
hrtdisp (Treatment)         0.3083 0.1463 3.511160e-02
papbl (BaselineMarker)      1.8662 0.9282 4.436775e-02
hrtdisp:papbl              -3.7931 1.1548 1.021672e-03
age                         1.7872 1.2034 1.375141e-01
dias                       -0.8270 1.0211 4.180127e-01
hypNo                      -0.8560 0.6636 1.971193e-01
```

```
hypYes                     -0.9329 0.6739 1.662278e-01
syst                        3.3869 1.2285 5.834062e-03
diabtrtYes                  0.9363 0.3711 1.164302e-02
lmsepi4+ episodes per week  0.1278 0.3903 7.434100e-01
lmsepiMissing              -0.2114 0.6500 7.450406e-01
lmsepiNo activity           0.4480 0.4086 2.729547e-01
lmsepiSome activity         0.1385 0.3515 6.935112e-01
```

We also show an example of estimating MELE without exploiting independent and with several covariates included in the model:

```
> melNoIndExtra <- mele(data=whiBioMarker,        ## dataset
+             response="stroke",        ## response variable
+             treatment="hrtdisp",        ## treatment variable
+             BaselineMarker="papbl",        ## biomarker
+             extra=c(
+                 "age"
+                 , "dias"
+                 , "hyp"
+                 , "syst"
+                 , "diabtrt"
+                 , "lmsepi"
+                 ),        ## extra variable(s)
+             phase="phase",        ## phase indicator
+             ind=FALSE        ## independent or non-indepentent
+ )
> melNoIndExtra

                              beta  stder         pVal
(Intercept)                -3.9227 0.7239 5.999024e-08
hrtdisp (Treatment)         0.3190 0.1587 4.441772e-02
papbl (BaselineMarker)      2.0377 1.0557 5.358469e-02
hrtdisp:papbl              -3.7559 1.3308 4.767720e-03
age                         1.8170 1.2290 1.392979e-01
dias                       -1.0119 1.0309 3.263064e-01
hypNo                      -0.7987 0.6694 2.328199e-01
hypYes                     -0.9390 0.6790 1.666968e-01
syst                        3.5970 1.2565 4.199987e-03
diabtrtYes                  0.7687 0.3844 4.551940e-02
lmsepi4+ episodes per week  0.1654 0.3953 6.756095e-01
lmsepiMissing              -0.2160 0.6578 7.426848e-01
lmsepiNo activity           0.4793 0.4148 2.478730e-01
lmsepiSome activity         0.1717 0.3586 6.319940e-01
```

4 case-cohort design

For two-arm, placebo-controlled trials with rare failure time endpoints, we can augment the case-only (ACO) design with random samples of controls from both arms, as in the classical case-cohort sampling scheme, or with a random sample of controls from the active treatment arm only. We show that these designs can identify all parameters in a Cox model and that the efficient case-only estimator can be incorporated in a two-step plug-in procedure[5]. A data example was shown in [5] incorporating case-only estimators in the classical case-cohort design improves the precision of all estimated parameters; sampling controls only in the active treatment arm attains a similar level of efficiency. A function "acoarm" was provided for case-cohort studies.

The inputs of acoarm function - acoarm(data, svtime, event, treatment, BaselineMarker, id, subcohort, esttype, augment , extra), include "data", a data frame for input data; "svtime", "event", "treatment" "BaselineMarker", "id", "subcohort", and "extra" are column names of "data" that store survival time, indicator of failure event, treatment, biomarker of interest, participant identifier, sub-cohort indicator, extra variables to be adjusted for, respectively; "esttype" defines the option for methods used in case-cohort model (1: Self-Prentice estimator, 0: Lin-Ying estimator); "augment" defines how the controls augmented to case-only data (0: from the placebo arm, 1: from the active treatment arm, or 2: from both arms).

We show a few examples to apply the function using the same data we used in the case-only section:

First we load the example dataset:

```
> data(acodata)
> dim(acodata)

[1] 907  14

> str(acodata)

'data.frame':        907 obs. of  14 variables:
 $ vacc1_evinf      : int  1442 1489 913 920 1448 1465 377 1274 1472 1463
 $ f_evinf          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ subcoh           : logi   TRUE FALSE FALSE FALSE TRUE FALSE ...
 $ ptid             : int  9601 9603 9605 9606 9607 9608 9609 9610 9613 961
 $ f_treat          : int  1 1 1 1 0 0 1 1 1 0 ...
 $ fcgr2a.3         : num  0 NA NA NA NA NA NA NA NA NA ...
 $ f_agele30        : int  0 0 0 0 1 0 0 0 1 1 ...
 $ f_hsv_2          : num  0 1 0 0 0 0 0 0 0 0 ...
 $ f_ad5gt18        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ f_crcm           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ any_drug         : num  1 1 1 0 0 0 0 1 0 0 ...
 $ num_male_part_cat: num  0 0 0 1 0 0 0 0 0 0 ...
 $ uias             : num  0 1 1 1 1 0 0 0 0 0 ...
 $ uras             : num  0 0 0 0 1 0 0 0 0 0 ...
```

Here is an example of ACO using controls from the placebo arm:

```
> rfit0 <- acoarm(data=acodata,  ## dataset
+                  svtime="vacc1_evinf", ## survival time
+                  event="f_evinf",  ## event
+                  treatment="f_treat", ## treatment
+                  BaselineMarker="fcgr2a.3",  #biomarker
+                  subcohort="subcoh", #subcohort
+                  esttype=1, ## use Self-Prentice method
+                  augment=0, ## augment from placebo arm
+                  extra=c("f_agele30"
+                          ,"f_hsv_2"
+                          ,"f_ad5gt18"
+                          ,"f_crcm"
+                          ,"any_drug"
+                          ,"num_male_part_cat"
+                          ,"uias"
+                          ,"uras")) ## extra varibles
> rfit0$Estimate

                                beta  stder   pVal
fcgr2a.3 (BaselineMarker)     0.1784 0.3871 0.6449
f_treat (Treatment)           0.6327 0.4856 0.1926
Marker-treatment interatcion -0.2550 0.3763 0.4980
f_agele30                     0.3637 0.6260 0.5612
f_hsv_2                       1.6177 0.6588 0.0141
f_ad5gt18                    -0.2784 0.6874 0.6855
f_crcm                        0.5609 1.0100 0.5787
any_drug                      0.9704 0.6623 0.1429
num_male_part_cat            -1.7869 0.8573 0.0371
uias                          0.7115 0.5203 0.1715
uras                          0.9528 0.6391 0.1360

> rfit0$Covariance

               fcgr2a.3     f_treat Interaction
fcgr2a.3     0.14982105  0.09291346 -0.07971158
f_treat      0.09291346  0.23580443 -0.15117945
Interaction -0.07971158 -0.15117945  0.14158429
```

Here is another example of ACO using controls from the active arm:

```
> rfit1 <- acoarm(data=acodata,  ## dataset
+                  svtime="vacc1_evinf",  ## survival time
+                  event="f_evinf",  ## event
+                  treatment="f_treat", ## treatment
+                  BaselineMarker="fcgr2a.3",  #biomarker
+                  subcohort="subcoh", #subcohort
+                  esttype=1, ## use Self-Prentice method
+                  augment=1,## augment from active arm
+                  weight=NULL,
+                  extra=c("f_agele30"
+                          ,"f_hsv_2"
+                          ,"f_ad5gt18"
+                          ,"f_crcm"
+                          ,"any_drug"
+                          ,"num_male_part_cat"
+                          ,"uias"
+                          ,"uras")) ## extra varibles
> rfit1$Estimate

                                beta  stder   pVal
fcgr2a.3 (BaselineMarker)     0.2360 0.3706 0.5242
f_treat (Treatment)           0.6327 0.4856 0.1926
Marker-treatment interatcion -0.2550 0.3763 0.4980
f_agele30                     0.1902 0.5041 0.7059
f_hsv_2                       0.8494 0.5389 0.1150
f_ad5gt18                     0.3646 0.4553 0.4233
f_crcm                       -0.1616 0.5843 0.7821
any_drug                      1.0837 0.5540 0.0505
num_male_part_cat             0.1792 0.6052 0.7671
uias                          0.0663 0.4531 0.8837
uras                          1.1437 0.4905 0.0197

> rfit1$Covariance

               fcgr2a.3    f_treat Interaction
fcgr2a.3     0.13734360  0.1105739 -0.09905306
f_treat      0.11057388  0.2358044 -0.15117945
Interaction -0.09905306 -0.1511794  0.14158429
```

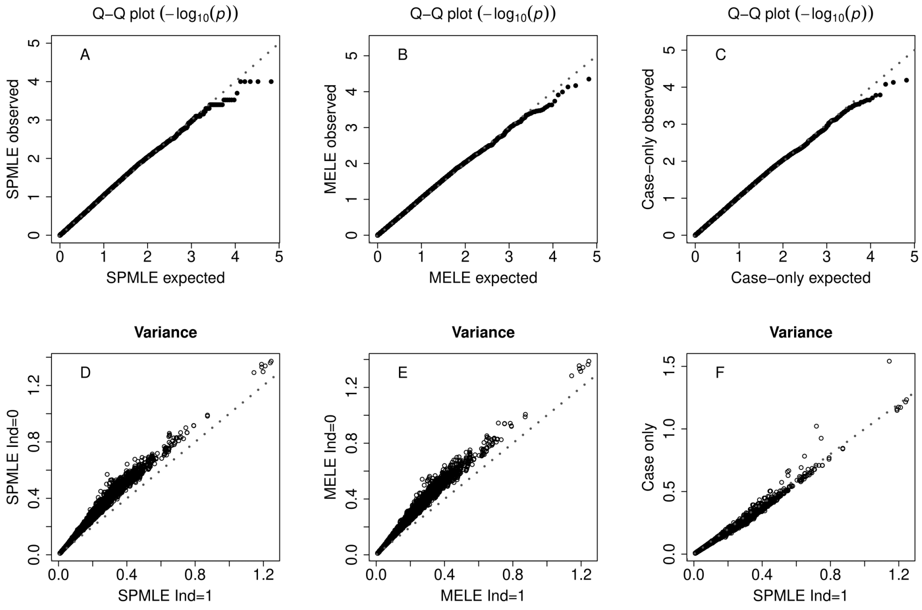
Here is an additional example of ACO using controls from both arms:

```
> rfit2 <- acoarm(data=acodata,  ## dataset
+                  svtime="vacc1_evinf",  ## survival time
+                   event="f_evinf",  ## event
+                   treatment="f_treat", ## treatment
+                   BaselineMarker="fcgr2a.3",  #biomarker
+                   subcohort="subcoh", #subcohort
+                   esttype=1, ## use Self-Prentice method
+                   augment=2,## augment from both arms
+                   weight=NULL,
+                   extra=c("f_agele30"
+                           ,"f_hsv_2"
+                           ,"f_ad5gt18"
+                           ,"f_crcm"
+                           ,"any_drug"
+                           ,"num_male_part_cat"
+                           ,"uias"
+                           ,"uras")) ## extra varibles
> rfit2

$Estimate
                                beta  stder   pVal
fcgr2a.3 (Baseline Marker)    0.1904 0.3119 0.5415
f_treat (Treatment)           0.6327 0.4856 0.1926
Marker-treatment interatcion -0.2550 0.3763 0.4980
f_agele30                     0.0740 0.3436 0.8294
f_hsv_2                       1.2066 0.3981 0.0024
f_ad5gt18                     0.1039 0.3728 0.7805
f_crcm                        0.1086 0.4375 0.8039
any_drug                      1.1332 0.3709 0.0022
num_male_part_cat            -0.4866 0.4127 0.2384
uias                          0.2364 0.3324 0.4769
uras                          1.1534 0.3458 0.0009

$Covariance
               fcgr2a.3     f_treat Interaction
fcgr2a.3     0.09727814  0.09846147  -0.0849746
f_treat      0.09846147  0.23580443  -0.1511794
Interaction -0.08497460 -0.15117945   0.1415843
```

5 apply to whole-genome data

The functions in the package can be applied to whole-genome SNP data. We applied the functions of caseonly, spmle, and mele to a more comprehensive dataset from WHI trial to estimate the interaction between biomarkers (SNPs) and hormone therapy (estrogen plus progestin) on type II diabetes. In total 21047 applicants in the trial were included, and 3147 of them have genome-wide SNP data. We used 78081 SNPs on chromosome 1 to show the package is scalable to whole-genome analysis.

The results are shown in the below. The quantile-quantile plots in the upper panels (Figure A, B, C) compare the distribution of observed p-values with that of a uniform-distributed p-values. Although there is no signficant p-value, the q-q line is right in the diagonal direction, suggesting the algorithm works well in estimation for all three methods. The first two graphics in the lower panels of Figure 1 (Figure D, E) shows the estimated variances of SNP-treatment interaction, using or without the independence between treatment and the SNP, suggesting that using independence yields a much more precise estimates of interaction. The last graph in the lower panel (Figure F) shows the comparison of the case-only estimator and the SPMLE estimator, suggesting the two agrees well in efficiency of estimation since type II diabete is relative rare in the WHI hormone trial.



6 session information

The version number of R and packages loaded for generating the vignette were:

```
R version 4.4.1 (2024-06-14)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04 LTS

Matrix products: default
```

```
BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: Etc/UTC
tzcode source: system (glibc)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] TwoPhaseInd_1.1.2

loaded via a namespace (and not attached):
 [1] compiler_4.4.1   Matrix_1.7-0     tools_4.4.1      survival_3.7-0
 [5] maketools_1.3.0  buildtools_1.0.0 splines_4.4.1    grid_4.4.1
 [9] knitr_1.48       xfun_0.47        sys_3.4.2        lattice_0.22-6
```

References

[1] Susan P Buchbinder, Devan V Mehrotra, Ann Duerr, Daniel W Fitzgerald, Robin Mogg, David Li, Peter B Gilbert, Javier R Lama, Michael Marmor, Carlos del Rio, M Juliana McElrath, Danilo R Casimiro, Keith M Gottesdiener, Jeffrey A Chodakewitz, Lawrence Corey, and Michael N Robertson. Efficacy assessment of a cell-mediated immunity hiv-1 vaccine (the step study): a double-blind, randomised, placebo-controlled, test-of-concept trial. *Lancet*, 372(9653):1881–1893, 11 2008.

[2] J. Y. Dai, C. Kooperberg, M. LeBlanc, and R. L. Prentice. Two-stage testing procedures with independent filtering for genome-wide gene-environment interaction. *Biometrika*, 99(4):929–944, 2012.

[3] J. Y. Dai, M. LeBlanc, and C. Kooperberg. Semiparametric estimation exploiting covariate independence in two-phase randomized trials. *Biometrics*, 65(1):178–187, Mar 2009.

[4] J. Y. Dai, S. S. Li, and P. B. Gilbert. Case-only methods for competing risks models with application to assessing differential vaccine efficacy by viral and host genetics. *Biostatistics*, 15(1):196–203, 2014.

[5] J. Y. Dai, X. C. Zhang, C. Y. Wang, and C. Kooperberg. Augmented case-only designs for randomized clinical trials with failure time endpoints. *Biometrics*, 2016.

[6] C. Kooperberg, M. Cushman, J. Hsia, J. G. Robinson, A. K. Aragaki, J. K. Lynch, A. E. Baird, K. C. Johnson, L. H. Kuller, S. A. Beresford, and B. Rodriguez. Can biomarkers identify women at increased stroke risk? the women's health initiative hormone trials. *PLoS clinical trials*, 2(6):e28, Jun 15 2007.

[7] J. P. Pandey, A. M. Namboodiri, S. Bu, J. Tapsoba, A. Sato, and J. Y. Dai. Immunoglobulin genes and the acquisition of hiv infection in a randomized trial of recombinant adenovirus hiv vaccine. *Virology*, 441(1):70–74, 2013.