Supplementary material of A mixed model approach to estimate the survivor average causal effect in cluster-randomized trials

Wei Wang[1], Fan Li[2,3], Shashivadan P. Hirani[4], Stanton P. Newman[4], Scott D. Halpern[1,6], Dylan S. Small[5], Guangyu Tong[2], and Michael O. Harhay[1,6]

[1]Clinical Trials Methods and Outcomes Lab, Palliative and Advanced Illness Research (PAIR) Center, Perelman School of Medicine, University of Pennsylvania

[2]Department of Biostatistics, Yale University School of Public Health

[3]Center for Methods in Implementation and Prevention Science, Yale University

[4]School of Health Sciences, City University London

[5]Department of Statistics and Data Science, The Wharton School, University of Pennsylvania

[6]Department of Biostatistics, Epidemiology and Informatics,

Perelman School of Medicine, University of Pennsylvania

Contents

S1 Estimation

S1.1 Complete data log-likelihood

The complete data log-likelihood of the i-th treated cluster follows as

$$\sum_{j \leq m_{1,i,1}} Z_{ss,ij} \left[\mathbf{x}'_{ij} \boldsymbol{\alpha}_{ss} - \frac{\left(y_{ij} - \mathbf{x}'_{ij} \boldsymbol{\beta}_{ss,1} - u_{1i}\right)^2}{2\sigma^2} \right]$$

$$+ \sum_{j \leq m_{1,i,1}} Z_{sn,ij} \left[\mathbf{x}'_{ij} \boldsymbol{\alpha}_{sn} - \frac{\left(y_{ij} - \mathbf{x}'_{ij} \boldsymbol{\beta}_{sn} - u_{1i}\right)^2}{2\sigma^2} \right]$$

$$-m_{1,i,1} \left[\frac{\log(\sigma^2)}{2} + \frac{\log(\tau^2)}{2} + \frac{u_{1i}^2}{2\tau^2} \right] - \sum_{j \leq m_{1,i}} \log \left[1 + \exp(\mathbf{x}'_{ij} \boldsymbol{\alpha}_{ss}) + \exp(\mathbf{x}'_{ij} \boldsymbol{\alpha}_{sn}) \right].$$

Similarly, the log-likelihood of the i-th control cluster follows as

$$\sum_{j\le m_{0,i,1}} Z_{ss,ij}\left[\mathbf{x}'_{ij}\boldsymbol{\alpha}_{ss}-\frac{\left(y_{ij}-\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,0}-u_{0i}\right)^2}{2\sigma^2}\right]+\sum_{j\le m_{0,i,0}} Z_{sn,ij}\mathbf{x}'_{ij}\boldsymbol{\alpha}_{sn}$$

$$-\sum_{j\le m_{0,i}}\log\left[1+\exp(\mathbf{x}'_{ij}\boldsymbol{\alpha}_{ss})+\exp(\mathbf{x}'_{ij}\boldsymbol{\alpha}_{sn})\right]-m_{0,i,1}\left[\frac{\log(\sigma^2)}{2}+\frac{\log(\tau^2)}{2}+\frac{u_{0i}^2}{2\tau^2}\right].$$

S1.2 E-step

In the i-th treated cluster, the joint density of (u_{1i}, y_{ij}) and marginal density of y_{ij} follows as

$$\begin{aligned}
f(u_{1i},y_{ij}) &= \left[\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,1}+u_{1i},\sigma^2)p_{ss,ij}+\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{sn}+u_{1i},\sigma^2)p_{sn,ij}\right]f(u_{1i})\\
f(y_{ij}) &= \mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,1},\tau^2+\sigma^2)p_{ss,ij}+\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{sn},\tau^2+\sigma^2)p_{sn,ij}.
\end{aligned}$$

Standard calculation in a normal mixture model gives

$$\eta_{ss,ij}\equiv \mathrm{P}(Z_{ss,ij}=1|y_{ij})=\frac{\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,1},\sigma^2+\tau^2)p_{ss,ij}}{\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,1},\tau^2+\sigma^2)p_{ss,ij}+\mathcal{N}(\mathbf{x}'_{ij}\boldsymbol{\beta}_{sn},\tau^2+\sigma^2)p_{sn,ij}},$$

and $\mathrm{P}(Z_{sn,ij}=1|y_{ij})=1-\eta_{ss,ij}$. In the i-th treated cluster, the conditional log-likelihood of the complete data given the observed data then follows as

$$\begin{aligned}
&\sum_{j\le m_{1,i,1}}\eta_{ss,ij}\left[\mathbf{x}'_{ij}\boldsymbol{\alpha}_{ss}-\frac{\left[y_{ij}-\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,1}-\mathrm{E}(u_{i1}|\mathbf{y}_i)\right]^2}{2\sigma^2}-\frac{\mathrm{Var}(u_{i1}|\mathbf{y}_i)}{2\sigma^2}\right]\\
&+\sum_{j\le m_{1,i,1}}\eta_{sn,ij}\left[\mathbf{x}'_{ij}\boldsymbol{\alpha}_{sn}-\frac{\left[y_{ij}-\mathbf{x}'_{ij}\boldsymbol{\beta}_{sn}-\mathrm{E}(u_{i1}|\mathbf{y}_i)\right]^2}{2\sigma^2}-\frac{\mathrm{Var}(u_{i1}|\mathbf{y}_i)}{2\sigma^2}\right]\\
&-m_{1,i,1}\left[\frac{\log(\sigma^2)}{2}+\frac{\log(\tau^2)}{2}+\frac{\mathrm{E}(u_{1i}^2|\mathbf{y}_i)}{2\tau^2}\right]-\sum_{j\le m_{1,i}}\log\left[1+\exp(\mathbf{x}'_{ij}\boldsymbol{\alpha}_{ss})+\exp(\mathbf{x}'_{ij}\boldsymbol{\alpha}_{sn})\right].
\end{aligned} \tag{1}$$

In the i-th control cluster, by Bayes' theorem, we have

$$\eta_{sn,ij}\equiv \mathrm{P}(Z_{sn,ij}=1|S_{ij}=0)=\frac{p_{sn,ij}}{p_{sn,ij}+p_{nn,ij}},\quad \mathrm{P}(Z_{ss,ij}=1|y_{ij})=1.$$

The joint distribution of u_{0i} and $\mathbf{y}_i$ is a multivariate normal distribution, and so is the conditional distribution of u_{0i} given $\mathbf{y}_i$. The conditional log-likelihood is then

$$\sum_{j\leq m_{0,i,1}}\left[\mathbf{x}_{ij}'\boldsymbol{\alpha}_{ss}-\frac{\left[y_{ij}-\mathbf{x}_{ij}'\boldsymbol{\beta}_{ss,0}-\mathrm{E}(u_{0i}|\mathbf{y}_i)\right]^2}{2\sigma^2}-\frac{\mathrm{Var}(u_{0i}|\mathbf{y}_i)}{2\sigma^2}\right]+\sum_{j\leq m_{0,i,0}}\eta_{sn,ij}\mathbf{x}_{ij}'\boldsymbol{\alpha}_{sn}$$
$$-m_{0,i,1}\left[\frac{\log(\sigma^2)}{2}+\frac{\log(\tau^2)}{2}+\frac{\mathrm{E}(u_{0i}^2|\mathbf{y}_i)}{2\tau^2}\right]-\sum_{j\leq m_{0,i}}\log\left[1+\exp(\mathbf{x}_{ij}'\boldsymbol{\alpha}_{ss})+\exp(\mathbf{x}_{ij}'\boldsymbol{\alpha}_{sn})\right]. \tag{2}$$

S1.3 M-step

The conditional complete data log-likelihood is the sum of (1) over all treated clusters plus the sum of (2) over all control clusters. We find estimates of $\boldsymbol{\beta}_{ss,1}$, $\boldsymbol{\beta}_{sn}$, $\boldsymbol{\beta}_{ss,0}$, σ^2 and τ^2 by standard calculation of the first and the second order derivatives. A closed form solution is not available for $\boldsymbol{\alpha}_{ss}$ or $\boldsymbol{\alpha}_{sn}$. Hence, we find their estimates using the Newton-Rapson algorithm.

S1.4 Proof of Proposition 1

We express the survivor average causal effect (SACE) as

$$\begin{aligned}E[Y_{ij}(1)-Y_{ij}(0)|S_{ij}(1)=S_{ij}(0)=1] &= \mathrm{E}[Y_{ij}(1)|S_{ij}(1)=S_{ij}(0)=1]\\ &\quad -\mathrm{E}[Y_{ij}(0)|S_{ij}(1)=S_{ij}(0)=1]\end{aligned}$$

Using double expectation, we have

$$\begin{aligned}\mathrm{E}\left[Y_{ij}(1)|S_{ij}(1)=S_{ij}(0)=1\right] &= \mathrm{E}\left[\mathrm{E}\left(Y_{ij}(1)|\mathbf{x}_{ij},u_{i1},S_{ij}(1)=S_{ij}(0)=1\right)|S_{ij}(1)=S_{ij}(0)=1\right]\\ &= \mathrm{E}\left[\mathbf{x}_{ij}'\boldsymbol{\beta}_{ss,1}+u_{i1}|S_{ij}(1)=S_{ij}(0)=1\right]\\ &= \frac{\mathrm{E}\left[\left(\mathbf{x}_{ij}'\boldsymbol{\beta}_{ss,1}+u_{i1}\right)I_{\{S_{ij}(1)=S_{ij}(0)=1\}}\right]}{\mathrm{P}\left[S_{ij}(1)=S_{ij}(0)=1\right]}\\ &= \frac{\mathrm{E}\left[\left(\mathbf{x}_{ij}'\boldsymbol{\beta}_{ss,1}+u_{i1}\right)p_{ss,ij}\right]}{\mathrm{P}\left[S_{ij}(1)=S_{ij}(0)=1\right]},\end{aligned}$$

where $p_{ss,ij}$ is independent of u_{i1}. Similarly, we get

$$\mathrm{E}[Y_{ij}(0)|S_{ij}(1) = S_{ij}(0) = 1] \quad = \quad \frac{\mathrm{E}[(\mathbf{x}'_{ij}\boldsymbol{\beta}_{ss,0} + u_{i0})p_{ss,ij}]}{\mathrm{P}[S_{ij}(1) = S_{ij}(0) = 1]}.$$

S2 Simulation results

Table S1: MSE ($\times 10^{-2}$) of estimating each regression coefficient between the fixed effects approach (FE) and the developed mixed effects approach (ME) where the number of clusters is $n_c = 30$. The average cluster size is $m = 25$ and 50, with the ICC $\rho = 0.01$, 0.05 and 0.1.

	True	m=25						m=50					
		0.01		0.05		0.1		0.01		0.05		0.1	
	value	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE
$\boldsymbol{\beta}_{ss,1}$	-0.5	1.91	2.05	2.04	2.44	2.53	3.47	1.28	1.48	1.28	1.63	1.58	2.10
	1	2.53	2.62	2.02	2.32	2.21	2.84	1.40	1.54	1.27	1.53	1.11	1.44
	1.5	0.57	0.58	0.67	0.75	0.51	0.64	0.37	0.39	0.27	0.33	0.24	0.31
$\boldsymbol{\beta}_{sn}$	−0.3	25.81	29.80	27.18	37.01	19.09	36.06	14.38	17.22	10.79	16.61	9.93	19.00
	0.8	109.45	123.84	102.43	125.42	70.85	123.20	58.93	64.16	59.99	76.91	36.78	65.41
	1.3	10.38	11.57	11.47	14.85	8.55	14.10	5.35	6.32	4.90	6.96	4.25	6.76
$\boldsymbol{\beta}_{ss,0}$	−0.2	0.92	0.92	1.25	1.25	1.67	1.74	0.53	0.53	0.80	0.81	1.13	1.17
	1	1.47	1.47	1.51	1.57	1.30	1.42	0.73	0.75	0.83	0.85	0.85	0.91
	1	0.40	0.40	0.36	0.34	0.44	0.47	0.19	0.19	0.20	0.21	0.16	0.18
$\boldsymbol{\alpha}_{ss}$	1	1.42	1.43	1.21	1.22	1.15	1.15	0.73	0.74	0.62	0.63	0.83	0.83
	2	4.23	4.19	4.93	4.94	4.35	4.39	2.59	2.61	2.65	2.67	2.41	2.38
	1	1.45	1.45	1.29	1.29	1.45	1.45	0.64	0.64	0.66	0.66	0.69	0.69
$\boldsymbol{\alpha}_{sn}$	−0.5	14.74	15.21	16.07	16.84	15.71	16.08	7.60	7.91	6.81	7.07	7.24	7.37
	−1.5	69.32	74.68	67.71	73.36	88.28	106.56	28.46	30.26	30.69	32.26	36.72	39.37
	−1	11.16	11.37	11.63	11.99	11.77	12.01	5.48	5.65	5.44	5.57	5.09	5.11

Table S2: MSE ($\times 10^{-2}$) of estimating each regression coefficient between the fixed effects approach (FE) and the developed mixed effects approach (ME) where the number of clusters is $n_c = 60$. The average cluster size is $m = 25$ and 50, with the ICC $\rho = 0.01$, 0.05 and 0.1.

		m=25						m=50					
	True	0.01		0.05		0.1		0.01		0.05		0.1	
	value	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE
$\boldsymbol{\beta}_{ss,1}$	-0.5	1.18	1.33	1.10	1.39	1.01	1.63	0.63	0.70	0.66	0.83	0.80	1.11
	1	1.27	1.39	1.23	1.43	0.92	1.44	0.65	0.70	0.65	0.78	0.57	0.80
	1.5	0.34	0.36	0.29	0.34	0.30	0.40	0.19	0.21	0.16	0.20	0.15	0.19
$\boldsymbol{\beta}_{sn}$	-0.3	17.21	20.10	12.38	19.53	10.04	21.52	7.56	8.90	6.61	10.34	5.06	10.40
	0.8	68.57	75.93	45.87	64.85	39.42	72.83	29.13	36.16	26.29	38.36	18.29	33.77
	1.3	7.45	8.41	4.99	7.45	4.59	8.17	3.02	3.48	2.42	3.52	2.09	3.65
$\boldsymbol{\beta}_{ss,0}$	-0.2	0.54	0.54	0.68	0.67	0.72	0.76	0.27	0.27	0.44	0.45	0.61	0.60
	1	0.82	0.81	0.72	0.72	0.72	0.83	0.39	0.39	0.39	0.40	0.29	0.32
	1	0.19	0.19	0.20	0.21	0.25	0.25	0.10	0.10	0.09	0.10	0.08	0.09
$\boldsymbol{\alpha}_{ss}$	1	0.76	0.76	0.68	0.68	0.66	0.66	0.35	0.35	0.34	0.34	0.33	0.33
	2	3.11	3.12	2.05	2.07	2.44	2.46	1.28	1.29	1.10	1.10	1.06	1.07
	1	0.73	0.73	0.61	0.61	0.64	0.64	0.42	0.42	0.30	0.30	0.30	0.30
$\boldsymbol{\alpha}_{sn}$	-0.5	8.76	8.94	5.84	6.05	6.93	7.09	3.33	3.44	3.85	3.90	3.24	3.36
	-1.5	38.53	40.57	41.40	44.25	34.09	35.53	13.31	14.22	14.32	15.01	11.35	12.09
	-1	5.65	5.75	4.45	4.58	4.32	4.41	2.37	2.44	2.42	2.44	2.40	2.46

Table S3: Coverage proportion (%) of each regression coefficient between the fixed effects approach (FE) and the developed mixed effects approach (ME) where the number of clusters is $n_c = 30$. The average cluster size is $m = 25$ and 50, with the ICC $\rho = 0.01$, 0.05 and 0.1.

		m=25						m=50					
	True	0.01		0.05		0.1		0.01		0.05		0.1	
	value	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE
$\boldsymbol{\beta}_{ss,1}$	-0.5	95	96	96.5	97	94	90	94.5	95	92	92	90	87
	1	92.5	95	94.5	97	94	94.5	93	96	89.5	93	95	97
	1.5	95	96.5	89	92.5	93.5	97.5	94	93	96	96	95	97
$\boldsymbol{\beta}_{sn}$	-0.3	91.5	95	95.5	96	95.5	94	94	95.5	94.5	97.5	95	96
	0.8	90	92.5	91	92.5	95	94	93	95	92.5	93	91.5	94.5
	1.3	91.5	94.5	94	93	95.5	94	95	95	92.5	95.5	93	95.5
$\boldsymbol{\beta}_{ss,0}$	-0.2	94.5	94.5	93.5	91.5	92.5	85.5	94	94	90	85	92	78
	1	93	95.5	95	95	91.5	94.5	92	92	90	92.5	88.5	92
	1	95	97	93.5	96	90.5	91.5	94	96	93.5	96.5	93.5	94.5
$\boldsymbol{\alpha}_{ss}$	1	93.5	93.5	92	94	97	97	92	91.5	94	96	90	93
	2	93.5	94.5	91.5	94	92	94.5	92	92.5	91.5	92.5	93	93.5
	1	91	92.5	93.5	94.5	91.5	90.5	93	93.5	93.5	96	94.5	95.5
$\boldsymbol{\alpha}_{sn}$	-0.5	96.5	96.5	97	96	95.5	97	91.5	94	93.5	96	93	93.5
	-1.5	95.5	96	94.5	98	94.5	93.5	92	95	91.5	94	95	94
	-1	96	96	92.5	95.5	94	95	93	94	91	90	93.5	91.5

Table S4: Coverage proportion (%) of each regression coefficient between the fixed effects approach (FE) and the developed mixed effects approach (ME) where the number of clusters is $n_c = 60$. The average cluster size is $m = 25$ and 50, with the ICC $\rho = 0.01$, 0.05 and 0.1.

	True	m=25						m=50					
		0.01		0.05		0.1		0.01		0.05		0.1	
	value	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE	ME	FE
$\boldsymbol{\beta}_{ss,1}$	-0.5	93	93.5	94.5	95.5	96.5	95.5	92.5	94.5	95	94.5	89.5	90
	1	94	97	94.5	95	95	96	95	95	91.5	94.5	93.5	94
	1.5	90.5	93.5	95	96.5	93.5	94	92	94	93	96.5	93.5	94.5
$\boldsymbol{\beta}_{sn}$	-0.3	92	94.5	93	93	91.5	94.5	91.5	91.5	93.5	94.5	92	94
	0.8	89	91	95	94.5	92	93.5	96	96	92.5	94	91.5	93.5
	1.3	91.5	91.5	91.5	92.5	91.5	94.5	91	93	94	95	92.5	94
$\boldsymbol{\beta}_{ss,0}$	-0.2	93	93.5	91	89.5	91	86	93	92.5	90	82	91.5	79
	1	93	94.5	93	94	93.5	94.5	92	92	91	93.5	95	95.5
	1	95	95	95	95.5	91	91	93.5	96	93.5	94	95.5	94.5
$\boldsymbol{\alpha}_{ss}$	1	90.5	89.5	95	94.5	94.5	94	94	93.5	93	93	92	93.5
	2	87.5	89	96.5	97	92.5	94.5	92.5	92.5	93.5	94	95	97
	1	92.5	92	94.5	96	92.5	94	92.5	92	96.5	96	93.5	92.5
$\boldsymbol{\alpha}_{sn}$	-0.5	90.5	92	96.5	96.5	94.5	94.5	93	92.5	90.5	91	95	94
	-1.5	92.5	94.5	93	92.5	91	94	93.5	95	93.5	95.5	95.5	95.5
	-1	93	94	95	95.5	96.5	96	93.5	94	94	95	93	94.5

S3 R code

```
# principal stratum probability computation
pmod=function(xi,alphass,alphasn){
# xi: observed covariate values of the subjects in the i-th cluster
ess=exp(xi%*%alphass)
esn=exp(xi%*%alphasn)
dsn=(1+ess+esn)
pnn=1/dsn
pss=ess*pnn
psn=esn*pnn
sdess=pss*(1-pss)
sdesn=psn*(1-psn)
eta0=esn/(esn+1) # E-step calculation
return(list(pss=pss,psn=psn,pnn=pnn,
sdesn=sdesn,sdess=sdess,eta0=eta0)) }

# E step calculation, normal mixture model
eta1fun=function(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1){
mss=x1i1%*%betass1
msn=x1i1%*%betasn
jointsd=sqrt(sat+taut)
nume=dnorm(y1i,mean=mss,sd=jointsd)*pss1
dsn=dnorm(y1i,mean=msn,sd=jointsd)*psn1
eta1i=nume/(nume+dsn)
return(eta1i)}

# MC EM calculation,
```

```
# conditional distribution of the random effect in a treated cluster
eu1fun=function(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1,nms){
mss=x1i1%*%betass1
msn=x1i1%*%betasn
sdtau=sqrt(taut)
sdsa=sqrt(sat)
nb=nms
sui=rnorm(nb,mean=0,sd=sdtau)
fyij=rep(0,nb)
for(k in 1:nb){
uk=sui[k]
d1=dnorm(y1i,mean=mss+uk,sd=sdsa)*pss1
d2=dnorm(y1i,mean=msn+uk,sd=sdsa)*psn1
fij=d1+d2
fyij[k]=prod(fij)}
fyi=mean(fyij)
if(fyi==0) {eui=eui2=vui=0} else {
eui=mean(sui*fyij)/fyi
eui2=mean(sui^2*fyij)/fyi
vui=eui2-(eui)^2}
return(list(vui=vui,eui2=eui2,eui=eui))}

# E-step, conditional distribution of the random effect in a control cluster
eu0fun=function(x0i1,y0i,m0i1,betass0,taut,sat){
mss0=x0i1%*%betass0
coef=taut/(m0i1*taut+sat)
eui=coef*sum(y0i-mss0)
vui=coef*sat
```

```
eui2=vui+eui^2
return(list(vui=vui,eui2=eui2,eui=eui))}

# parameter estimation in treated clusters
trfun=function(xm1,sind1,y1,n1,m1v,m1v1,
alphass,alphasn,betass1,betasn,taut,sat,p,nms,xnames){
bssp=bsnp=matrix(0,p,p)
bssy=bsny=matrix(0,p,1)
dass1=dasn1=matrix(0,p,1)
dass2=dasn2=matrix(0,p,p)
aess=aesn=matrix(0,p,1)
for(i in 1:n1){
cind=which(xm1$cid==i)
x1i=as.matrix(xm1[cind,xnames]) # m_{1,i} x p
tx1i=t(x1i) # p x m_{1,i}
m1i=m1v[i]
m1i1=m1v1[i]
pmod1=pmod(x1i,alphass,alphasn)
pss=pmod1$pss
psn=pmod1$psn
sdss=pmod1$sdess
sdsn=pmod1$sdesn
pnn=pmod1$pnn
if(m1i1>0){
sind1i=sind1[[i]]
x1i1=x1i[sind1i,,drop=F] # m_{1,i,1} x p
tx1i1=t(x1i1)  # p x m_{1,i,1}
y1i=y1[cind][sind1i]
```

```
pss1=pss[sind1i]
psn1=psn[sind1i]
etai=eta1fun(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1)
euo=eu1fun(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1,nms)
cyi=y1i-euo$eui
metai=matrix(etai,nrow=p,ncol=m1i1,byrow=T)
etaix1i1=tx1i1*metai
etai1x1i1=tx1i1-etaix1i1
bssp=bssp+etaix1i1%*%x1i1
bssy=bssy+etaix1i1%*%cyi
bsnp=bsnp+etai1x1i1%*%x1i1
bsny=bsny+etai1x1i1%*%cyi
aess=aess+rowSums(etaix1i1)
aesn=aesn+rowSums(etai1x1i1)
} else {bssp=bssp
        bssy=bssy
        bsnp=bsnp
        bsny=bsny
        aess=aess
        aesn=aesn}
mpss=matrix(pss,nrow=p,ncol=m1i,byrow=T)
msdss=matrix(sdss,nrow=p,ncol=m1i,byrow=T)
dass1=dass1-rowSums(tx1i*mpss)
dass2=dass2-(tx1i*msdss)%*%x1i
mpsn=matrix(psn,nrow=p,ncol=m1i,byrow=T)
msdsn=matrix(sdsn,nrow=p,ncol=m1i,byrow=T)
dasn1=dasn1-rowSums(tx1i*mpsn)
dasn2=dasn2-(tx1i*msdsn)%*%x1i}
```

```
nbetass1=lm.fit(bssp,bssy)$coef
nbetasn=lm.fit(bsnp,bsny)$coef
return(list(nbetass1=nbetass1,nbetasn=nbetasn,dass1=dass1,
dass2=dass2,dasn1=dasn1,dasn2=dasn2,
aess=aess,aesn=aesn))}

# parameter estimation in control clusters
confun=function(xm0,sind0,y0,n0,m0v,m0v1,
alphass,alphasn,betass0,taut,sat,p,xnames){
bssp0=matrix(0,p,p)
bssy0=matrix(0,p,1)
dasn1c=dass1c=matrix(0,p,1)
dasn2c=dass2c=matrix(0,p,p)
aessc=aesnc=matrix(0,p,1)
for(i in 1:n0){
cind=which(xm0$cid==i)
x0i=as.matrix(xm0[cind,xnames]) # m_{0,i} x p
m0i=m0v[i]
m0i1=m0v1[i]
m0i0=m0i-m0i1
pmod0=pmod(x0i,alphass,alphasn)
pss0=pmod0$pss
psn0=pmod0$psn
sdss0=pmod0$sdess
sdsn0=pmod0$sdesn
pnn0=pmod0$pnn
tx0i=t(x0i) # p x m_{0,i}
sind0i=sind0[[i]]
```

```
if(m0i1>1){
x0i1=x0i[sind0i,,drop=F] # m_{0,i,1} x p
tx0i1=t(x0i1)  # p x m_{0,i,1}
y0i=y0[cind][sind0i]
eu0=eu0fun(x0i1,y0i,m0i1,betass0,taut,sat)
cyi=y0i-eu0$eui
bssp0=bssp0+tx0i1%*%x0i1
bssy0=bssy0+tx0i1%*%cyi
aessc=aessc+rowSums(tx0i1)
} else {bssp0=bssp0
        bssy0=bssy0
        aessc=aessc}
if(m0i0>0) {
if(m0i1>=1){
x0i0=x0i[-sind0i,,drop=F] # m_{0,i,0} x p
etai=pmod0$eta0[-sind0i] } else { x0i0=x0i
                                  etai=pmod0$eta0}
tx0i0=t(x0i0)
metai=matrix(etai,nrow=p,ncol=m0i0,byrow=T)
p0=rowSums(tx0i0*metai)
aesnc=aesnc+p0 } else {aesnc=aesnc}
mpss0=matrix(pss0,nrow=p,ncol=m0i,byrow=T)
msdss0=matrix(sdss0,nrow=p,ncol=m0i,byrow=T)
dass1c=dass1c-rowSums(tx0i*mpss0)
dass2c=dass2c-(tx0i*msdss0)%*%x0i
mpsn0=matrix(psn0,nrow=p,ncol=m0i,byrow=T)
msdsn0=matrix(sdsn0,nrow=p,ncol=m0i,byrow=T)
dasn1c=dasn1c-rowSums(tx0i*mpsn0)
```

```
dasn2c=dasn2c-(tx0i*msdsn0)%*%x0i}
nbetass0=lm.fit(bssp0,bssy0)$coef
return(list(nbetass0=nbetass0,dass1c=dass1c,dass2c=dass2c,
dasn1c=dasn1c,dasn2c=dasn2c,aessc=aessc,aesnc=aesnc))}

# fixed effects estimation
fefun=function(xm1,sind1,y1,n1,m1v,m1v1,alphass,alphasn,betass1,betasn,
xm0,sind0,y0,n0,m0v,m0v1,m01,betass0,taut,sat,p,nms,xnames){
to=trfun(xm1,sind1,y1,n1,m1v,m1v1,
alphass,alphasn,betass1,betasn,taut,sat,p,nms,xnames)
co=confun(xm0,sind0,y0,n0,m0v,m0v1,
alphass,alphasn,betass0,taut,sat,p,xnames)

# Newton-Raphson estimates
nalphass=alphass-solve(to$dass2+co$dass2c,to$dass1+co$dass1c+to$aess+co$aessc)
nalphasn=alphasn-solve(to$dasn2+co$dasn2c,to$dasn1+co$dasn1c+to$aesn+co$aesnc)

return(list(nbetass1=to$nbetass1,nbetasn=to$nbetasn,
nbetass0=co$nbetass0,nalphass=nalphass,nalphasn=nalphasn))}

# variance parameters estimation
vfun=function(xm1,sind1,y1,n1,m1v1,xm0,sind0,y0,n0,m0v1,
betass1,betasn,betass0,alphass,alphasn,
taut,sat,nms,xnames,m0,m1){
den1=den0=0
eu02=rep(0,n0)
eu12=rep(0,n1)
m11=sum(m1v1)
```

```
m01=sum(m0v1)
pssi1=fyi1=psni1=0

for(i in 1:n1){
cind=which(xm1$cid==i)
x1i=as.matrix(xm1[cind,xnames,drop=F])# m_{1,i} x p
m1i1=m1v1[i]
pmod1=pmod(x1i,alphass,alphasn)
pss=pmod1$pss
psn=pmod1$psn
pssi1=pssi1+sum(pss)
psni1=psni1+sum(psn)

if(m1i1>0){
sind1i=sind1[[i]]
x1i1=x1i[sind1i,,drop=F]
y1i=y1[cind][sind1i]
pss1=pss[sind1i]
psn1=psn[sind1i]
etai=eta1fun(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1)
etai1=1-etai
euo=eu1fun(x1i1,y1i,betass1,betasn,taut,sat,pss1,psn1,nms)
eu1i=euo$eui
eu12[i]=euo$eui2
ty1ss=etai*(y1i-x1i1%*%betass1-eu1i)^2
ty1sn=etai1*(y1i-x1i1%*%betasn-eu1i)^2
den1=den1+sum(ty1ss+ty1sn+euo$vui)
fyi1=fyi1+sum(pss*(x1i%*%betass1+eu1i))} else {den1=den1
```

```
        fyi1=fyi1}}

pssi0=fyi0=psni0=0
for(i in 1:n0){
cind=which(xm0$cid==i)
x0i=as.matrix(xm0[cind,xnames])
m0i1=m0v1[i]
pmod0=pmod(x0i,alphass,alphasn)
pss0=pmod0$pss
psni0=psni0+sum(pmod0$psn)
pssi0=pssi0+sum(pss0)

if(m0i1>0){
sind0i=sind0[[i]]
x0i1=x0i[sind0i,,drop=F]
y0i=y0[cind][sind0i]
eu0=eu0fun(x0i1,y0i,m0i1,betass0,taut,sat)
eu02[i]=eu0$eui2
ty0ss=(y0i-x0i1%*%betass0-eu0$eui)^2
den0=den0+sum(ty0ss+eu0$vui)
fyi0=fyi0+sum(pss0*(x0i%*%betass0+eu0$eui))
} else {den0=den0
        fyi0=fyi0}}
nsat=(den1+den0)/(m11+m01)
ntaut=(sum(m1v1*eu12)+sum(m0v1*eu02))/(m11+m01)
sace1=fyi1/pssi1-fyi0/pssi0 # SACE estimate

pss=(pssi0+pssi1)/(m1+m0)
```

```
psn=(psni0+psni1)/(m1+m0)
pnn=1-pss-psn
return(list(nsat=nsat,ntaut=ntaut,sace1=sace1,
pss=pss,psn=psn,pnn=pnn))}

# main function
ffun=function(da1,da0,xnames){
# da1: the treatment group data set
# da0: the control group data set
# xnames: names of the covariates

n1=length(unique(da1$cid))
# number of treated clusters
m1v=as.vector(table(da1$cid))
# cluster size
m1v1=as.vector(table(da1$cid,da1$yind)[,2])
# number of survivors by cluster
xm1=da1[,c("cid",xnames),drop=F]
p=ncol(xm1)-1
y1=da1[,"y"]
sind1=tapply(da1$yind,da1$cid,function(x){which(x==1)})
# survivor index by cluster
m1=sum(m1v)

n0=length(unique(da0$cid))
m0v=as.vector(table(da0$cid))
m0v1=as.vector(table(da0$cid,da0$yind)[,2])
xm0=da0[,c("cid",xnames),drop=F]
```

```
y0=da0[,"y"]
sind0=tapply(da0$yind,da0$cid,function(x){which(x==1)})
m0=sum(m0v)

# starting values
alphass=(1:p)/(100*p)
alphasn=(p:1)/(90*p)
s1=lm(y1~as.matrix(xm1[,-1])-1)
s0=lm(y0~as.matrix(xm0[,-1])-1)
betass1=s1$coef
betass0=s0$coef
betasn=(betass1+betass0)/2
sat=(summary(s1)$sigma^2+summary(s0)$sigma^2)/2
taut=sat/5

#initation
dif=1
nit=35 # number of maximum iterations
thr=1e-3 # threshold
nuit=0
nms=100 # number of Monte Carlo EM samples

# iteration
while(any(dif>thr)&(nuit<nit)){
# fixed effects estimation
fo=fefun(xm1,sind1,y1,n1,m1v,m1v1,alphass,alphasn,betass1,
betasn,xm0,sind0,y0,n0,m0v,m0v1,m01,betass0,taut,sat,p,nms,xnames)
```

```
dif.betass1=betass1-fo$nbetass1
dif.betasn=betasn-fo$nbetasn
dif.betass0=betass0-fo$nbetass0
dif.beta=c(dif.betass1,dif.betasn,dif.betass0)
betass1=fo$nbetass1
betasn=fo$nbetasn
betass0=fo$nbetass0

dif.alphass=alphass-fo$nalphass
dif.alphasn=alphasn-fo$nalphasn
dif.alpha=c(dif.alphass,dif.alphasn)
alphass=fo$nalphass
alphasn=fo$nalphasn

# variance parameters estimation
vo=vfun(xm1,sind1,y1,n1,m1v1,xm0,sind0,y0,n0,m0v1,betass1,
betasn,betass0,alphass,alphasn,taut,sat,nms,xnames,m0,m1)
dif.v=c(sat-vo$nsat,taut-vo$ntaut)
sat=vo$nsat
taut=vo$ntaut
dif=c(dif.v,dif.alpha,dif.beta)
nuit=nuit+1
}
return(obj=c(betass1,betasn,betass0,
alphass,alphasn,sat,taut,vo$sace1,vo$pss,vo$psn,vo$pnn))
# returned values:
# coefficient vectors of the outcome models, betass1,betasn,betass0
# coefficient vectors of the membership models, alphass,alphasn
```

```
# variance parameters, sat, taut
# SACE: sace1
# principal stratum proportions, pss, psn, pnn
}

# SACE estimation
csace=function(da,y.name,tr.name,xnames,cluster.name){
# da: the data set
# y.name: the name of the outcome
# tr.name: name of the treatment variable
# xnames: names of the covariates
# cluster.name: name of the cluster variable
da$y=da[,y.name]
da$yind=1 # survivors
da$yind[is.na(da$y)]=0 # non-survivors

da0=da[which(da[,tr.name]==0),] # control group
da1=da[which(da[,tr.name]==1),] # treatment group

# new cluster index

da0$cid=da0[,cluster.name]
ugp0=sort(unique(da0[,cluster.name]))
lugp0=length(ugp0)
for(i in 1:lugp0){
da0$cid[which(da0[,cluster.name]==ugp0[i])]=i}
da1$cid=da1[,cluster.name]
ugp1=sort(unique(da1[,cluster.name]))
```

```
lugp1=length(ugp1)
for(i in 1:lugp1){
da1$cid[which(da1[,cluster.name]==ugp1[i])]=i}

# estimation
fobj=ffun(da1,da0,xnames)
return(fobj)}
```