

1 Appendix

Example in Section 4.1, the PFS biomarker subgroup analysis

R code for columns 3 and 4 in Table 7.

```
library("dplyr")
library("survival")
library("RISCA")
library("survRM2")
```

#####

```
biomark_mut_h220 <- read.csv('yourroute.csv')
```

Our response is TRT01P, the planned treatment

Our adjusted variables is HER2TRT, ECOGBL,

HRRECP, REGION, and DISLOC

We use Integer Encoding to encode the levels of covariates

#####

Below is the code for unadjusted RMST(column 3)

```
rmst_result_sub1_unadj <-
rmst2(biomark_mut_h220$AVAL_PFS,
      biomark_mut_h220$status_PFS,
      biomark_mut_h220$TRT01P_encode, tau=24 )
```

#####

Below is the code for adjusted RMST(column 4 in our table)

```
Pr0 <- glm(TRT01P_encode ~ 1, family = binomial(link="logit"),
          data=biomark_mut_h220)$fitted.values[1]
Pr1 <- glm(TRT01P_encode ~ HER2TRT_encode
          +ECOGBL_encode+HRRECP_encode+
          REGION_encode+DISLOC_encode,
          data=biomark_mut_h220,
          family=binomial(link = "logit"))$fitted.values
W <- (biomark_mut_h220$TRT01P_encode==1) * (1/Pr1) +
      (biomark_mut_h220$TRT01P_encode==0) * (1)/(1-Pr1)
```

```
res.akm <- ipw.survival(times=biomark_mut_h220$AVAL_PFS,
                       failures=biomark_mut_h220$status_PFS,
```

```

variable=biomark_mut_h220$TRT01P_encode, weights=W)$table.surv
adjusted.LR(times=biomark_mut_h220$AVAL_PFS,
             failures=biomark_mut_h220$status_PFS,
             variable=biomark_mut_h220$TRT01P_encode, weights=W)

##### only if you want to plot the two adjusted KM curves #####
lines(res.akm$times[res.akm$variable==1],
      res.akm$survival[res.akm$variable==1],
      type="s", col=2, lwd=2)
lines(res.akm$times[res.akm$variable==0],
      res.akm$survival[res.akm$variable==0],
      type="s", col=1, lwd=2)

#####
biomark_mut_h220$W <- W
survdat1 <- res.akm[res.akm$variable==1,]
wtdat1 <- biomark_mut_h220[biomark_mut_h220$TRT01P_encode==1,]
wtdat1$time <-
biomark_mut_h220[biomark_mut_h220$TRT01P_encode==1,]$AVAL_PFS

wtdat1$status <-
biomark_mut_h220[biomark_mut_h220$TRT01P_encode==1,]$status_PFS

survdat0 <- res.akm[res.akm$variable==0,]
wtdat0 <- biomark_mut_h220[biomark_mut_h220$TRT01P_encode==0,]

wtdat0$time <-
biomark_mut_h220[biomark_mut_h220$TRT01P_encode==0,]$AVAL_PFS

wtdat0$status <-
biomark_mut_h220[biomark_mut_h220$TRT01P_encode==0,]$status_PFS

## Use coded function to compute the ARMST,
## slightly different than original version in GitHub
rmstlakme_1 <- rmst1_AKME(survdata =survdat1,wtdata =wtdat1,tau=24)
rmstlakme_0 <- rmst1_AKME(survdata =survdat0,wtdata =wtdat0,tau=24)

rmstlakme_1$rmst[1:2]
rmstlakme_0$rmst[1:2]

#Use the coded function to compute the ARMST difference and ratio
armst <- ARMST_diff(rmst_trt = rmstlakme_1$rmst[1],
                   rmst_ctrl = rmstlakme_0$rmst[1],
                   rmst_var_trt = rmstlakme_1$rmst.var,
                   rmst_var_ctrl = rmstlakme_0$rmst.var)

```

```

#ARMST function
#This is a modification of the R function 'rmst1' in 'survRM2'
rmst1_AKME=function(survdata, wtdata, tau, alpha=0.05){
  #-- time
  #-- statuts
  #-- tau -- truncation time
  #-- alpha -- gives (1-alpha) confidence interval
  #-- W weighting from AKME

  wtdata<-wtdata[order(wtdata$time),]

  idx=survdata$time<=tau
  wk.time=sort(c(survdata$time[idx],tau))

  w1 <-sapply(wk.time,function(x){sum(wtdata$W[wtdata$time >= x])})
  w2 <-sapply(wk.time,function(x){sum((wtdata$W[wtdata$time >= x])^2)})

  wk.surv=survdata$survival[idx]
  wk.n.risk =survdata$n.risk[idx]
  wk.n.event=survdata$n.event[idx]

  time.diff <- diff(c(0, wk.time))
  areas <- time.diff * c(1, wk.surv)
  rmst = sum(areas)
  rmst

  wk.var <- ifelse((wk.n.risk-wk.n.event)==0, 0,
                  wk.n.event /((wk.n.risk - wk.n.event)))
  wk.var =c(wk.var,0)
  rmst.var =
  sum(((cumsum(rev(areas)[-1]))^2*rev(wk.var)[-1])*rev(w2)[-1])/rev(w1)[-1]^2)
  rmst.se = sqrt(rmst.var)

  #--- check ---
  # print(ft, rmean=tau)
  #--- output ---
  out=matrix(0,2,4)
  out[1,]=c(rmst, rmst.se, rmst-qnorm(1-alpha/2)*rmst.se,
            rmst+qnorm(1-alpha/2)*rmst.se)
  out[2,]=c(tau-out[1,1], rmst.se, tau-out[1,4], tau-out[1,3])
  rownames(out)=c("RMST", "RMTL")
  colnames(out)
  =c("Est.", "se", paste("lower .", round((1-alpha)*100, digits=0), sep=""),

```

```

        paste("upper .",round((1-alpha)*100, digits=0), sep=""))

Z=list()
Z$result=out
Z$rmst = out[1,]
Z$rmstl = out[2,]
Z$tau=tau
Z$rmst.var = rmst.var

return(Z)

}

##### ARMST difference and ratio
ARMST_diff <- function(rmst_trt=rmst_trt,
                      rmst_ctrl=rmst_ctrl,
                      rmst_var_trt=rmst_var_trt,
                      rmst_var_ctrl=rmst_var_ctrl,
                      alpha=0.05){

rmst.diff.10 = rmst_trt - rmst_ctrl
rmst.diff.10.se = sqrt(rmst_var_ctrl + rmst_var_trt)
rmst.diff.10.low = rmst.diff.10 - qnorm(1 - alpha/2) * rmst.diff.10.se

rmst.diff.10.upp = rmst.diff.10 + qnorm(1 - alpha/2) * rmst.diff.10.se

rmst.diff.pval = pnorm(-abs(rmst.diff.10)/rmst.diff.10.se) * 2

rmst.diff.result=cbind(round(rmst.diff.10,6),round(rmst.diff.10.se,6),
                      round(rmst.diff.10.low,6), round(rmst.diff.10.upp,6),
                      round(rmst.diff.pval,6))

rmst.log.ratio.10 = log(rmst_trt) - log(rmst_ctrl)

rmst.log.ratio.10.se=
  sqrt(rmst_var_trt/rmst_trt/rmst_trt + rmst_var_ctrl/rmst_ctrl/rmst_ctrl)

rmst.log.ratio.10.low =
  rmst.log.ratio.10 - qnorm(1-alpha/2)*rmst.log.ratio.10.se

rmst.log.ratio.10.upp =
  rmst.log.ratio.10 + qnorm(1-alpha/2)*rmst.log.ratio.10.se

rmst.log.ratio.pval=pnorm(-abs(rmst.log.ratio.10)/rmst.log.ratio.10.se)*2
rmst.ratio.result =

```

```

cbind(round(exp(rmst.log.ratio.10),6),round((rmst.log.ratio.10.se),6),
round(exp(rmst.log.ratio.10.low),6),round(exp(rmst.log.ratio.10.upp),6),
round(rmst.log.ratio.pval,6))

colnames(rmst.ratio.result) <- c('est','se','lcl','ucl','p-value')
colnames(rmst.diff.result) <- c('est','se','lcl','ucl','p-value')

return_list <- list(rmst.diff.result = rmst.diff.result,
                    rmst.ratio.result = rmst.ratio.result)
return(return_list)
}

## SAS code for columns 5 and 6 in Table 7

libname akme \Please provide your location";
proc import datafile=
    "Please_provide_your_location\biomark_mut_h220.csv"
out=hs_ge220 dbms=csv replace;
getnames=yes;
run;

proc rmstreg data=hs_ge220 tau=24;
class TRT01P_encode(ref=first) HRRECP_encode(ref=first)
    ECOGBL_encode(ref=first) REGION_encode(ref=first)
    HER2TRT_encode(ref=first) DISLOC_encode(ref=first);
model AVAL_PFS*status_PFS(0) =
    TRT01P_encode HRRECP_encode ECOGBL_encode
    REGION_encode HER2TRT_encode DISLOC_encode/ link
    =linear method=ipcw;
lsmeans TRT01P_encode/cl;
run;

proc rmstreg data=hs_ge220 tau=24;
class TRT01P_encode(ref=first) HRRECP_encode(ref=first)
    ECOGBL_encode(ref=first) REGION_encode(ref=first)
    HER2TRT_encode(ref=first) DISLOC_encode(ref=first);
model AVAL_PFS*status_PFS(0) =
    TRT01P_encode HRRECP_encode ECOGBL_encode
    REGION_encode HER2TRT_encode DISLOC_encode/ link
    =linear method=pv;
lsmeans TRT01P_encode/cl;
run;

```

```

proc rmstreg data=hs_ge220 tau=24;
class TRT01P_encode(ref=first) HRRECP_encode(ref=first)
      ECOGBL_encode(ref=first) REGION_encode(ref=first)
      HER2TRT_encode(ref=first) DISLOC_encode(ref=first);
model AVAL_PFS*status_PFS(0) =
      TRT01P_encode HRRECP_encode ECOGBL_encode
      REGION_encode HER2TRT_encode DISLOC_encode/ link
      =log method=ipcw;
lsmeans TRT01P_encode/cl;
run;

proc rmstreg data=hs_ge220 tau=24;
class TRT01P_encode(ref=first) HRRECP_encode(ref=first)
      ECOGBL_encode(ref=first) REGION_encode(ref=first)
      HER2TRT_encode(ref=first) DISLOC_encode(ref=first);
model AVAL_PFS*status_PFS(0) =
      TRT01P_encode HRRECP_encode ECOGBL_encode
      REGION_encode HER2TRT_encode DISLOC_encode/ link
      =log method=pv;
lsmeans TRT01P_encode/cl;
run;

```

Example in Section 4.2, NALA CNS analysis

```

library(cmpsrk)
library(dplyr)
library(RISCA)
library(ggplot2)

```

function to calculate the area under the curve.

```

AUC<-function(x,y,tau){
  indx<-which(x<=tau)
  x<-sort(c(x[indx],tau))
  y<-y[indx]
  dx<-diff(x)
  return(sum(dx*y))
}

```

calculate the cumulative incidence function and

```

# adjusted cumulative incidence function
cumulative_incidence<-function(time,status,censor,event, trt, w=NULL)
{

  status0<-ifelse(status==censor,0,1)
  status1<-ifelse(status==event,1,0)
  km0<-ipw.survival(time,status0,trt,w)$table.surv
  %>% rename(trt=variable)
  km1<-ipw.survival(time,status1,trt,w)$table.surv
  %>% rename(trt=variable)
  %>% rename(n.event_j=n.event)
  %>% select(times,n.event_j,trt)

  km<-left_join(km0,km1,by=c("trt","times"))
  km$n.event_j[is.na(km$n.event_j)]<-0
  km$s<-lag(km$survival)
  km$s[which(km$times==0)]<-1
  km$dyis<-km$n.event_j/km$n.risk*km$s

  arm0<-km %>% subset(trt==0) %>% arrange(.,by_group=times)
  %>% mutate(cif=cumsum(dyis)) %>% rename(time=times)
  arm1<-km %>% subset(trt==1) %>% arrange(.,by_group=times)
  %>% mutate(cif=cumsum(dyis)) %>% rename(time=times)
  outdata<-rbind(arm0,arm1)

}

#prepare CNS dataset
cns<-read.csv('cns2.csv')
%>% mutate(trt=ifelse(TRT01P=='L1250 + C2000',0,1))
%>% mutate(strat1=ifelse(PREVHER2=='N=2',0,1))
%>% mutate(strat2=ifelse(HRRECP=="Negative",0,1))
%>% mutate(strat3=ifelse(DISLOC=='VISCERAL',0,1))

adsl<-read.csv('adsl.csv')
%>% mutate(trt=ifelse(TRT01P=='L1250 + C2000',0,1))
%>% mutate(IHC=ifelse(HER2IHC=="2+",0,1)) %>%
  mutate(BRAIN=ifelse(BRAMETBL=='N',0,1))

adsl2<-adsl %>% select(SUBJID,IHC,BRAIN)
cns<-left_join(cns,adsl2,by="SUBJID")

#set up tau
tau<-48

```

```

#set up time and event status.  CNS=0, death=1, censor=2
time<-cns$AVAL
status<-cns$CNSR
censor<-2
event<-0
trt<-cns$trt

# unadjusted model for CNS.  for "Death", change "event" to 1
m1<-cumulative_incidence(time,status,censor,event, trt, w=NULL)

arm0<-m1 %>% subset(trt==0)
      %>% mutate(group='L+C')
      %>% subset(time<=tau)

arm1<-m1 %>% subset(trt==1)
      %>% mutate(group='N+C')
      %>% subset(time<=tau)

tmp<-arm0[nrow(arm0),] %>% mutate(time=tau)
arm0<-rbind(arm0,tmp)
tmp<-arm1[nrow(arm1),] %>% mutate(time=tau)
arm1<-rbind(arm1,tmp)

rmtl0<-AUC(arm0$time,arm0$cif,tau) # rmtl for control
rmtl1<-AUC(arm1$time,arm1$cif,tau) # rmtl for treatment

rmtl1-rmtl0 # rmtl difference

# adjusted model by HER2IHC, and stratification factors.
# For "death", change event to 1.

Pr1 <- glm(trt ~ strat1+strat2+strat3+IHC,
           family = binomial(link="logit"),data=cns)$fitted.values
w <- (cns$trt==1) * (1/Pr1) + (cns$trt==0) * (1)/(1-Pr1)

m2<-cumulative_incidence(time,status,censor,event, trt, w=w)

arm0_adj<-m2 %>% subset(trt==0)
              %>% mutate(group='L+C_adjusted')
              %>% subset(time<=tau)

```



```

arm1_adj<-m2 %>% subset(trt==1)
                                %>% mutate(group='N+C_adjusted')
                                %>% subset(time<=tau)

tmp<-arm0_adj[nrow(arm0_adj),] %>% mutate(time=tau)
arm0_adj<-rbind(arm0_adj,tmp)
tmp<-arm1_adj[nrow(arm1_adj),] %>% mutate(time=tau)
arm1_adj<-rbind(arm1_adj,tmp)

# adjusted rmtl for control
rmtl0_adj<-AUC(arm0_adj$time,arm0_adj$cif,tau)

# adjusted rmtl for treatment
rmtl1_adj<-AUC(arm1_adj$time,arm1_adj$cif,tau)

rmtl1_adj-rmtl0_adj # adjusted rmtl difference

# bootstrap for confidence interval

rmtl<-NULL

set.seed(1234)

for (i in 1:10000){

  indx<-sample(1:621,621,replace=T)
  cns2<-cns[indx,]

  time<-cns2$AVAL
  status<-cns2$CNSR
  censor<-2
  event<-0
  trt<-cns2$trt

  m1<-cumulative_incidence(time,status,censor,event, trt, w=NULL)
  arm0<-m1 %>% subset(trt==0)
  arm1<-m1 %>% subset(trt==1)

  rmtl0<-AUC(arm0$time,arm0$cif,tau)
  rmtl1<-AUC(arm1$time,arm1$cif,tau)

  diff<-rmtl1-rmtl0

  Pr1 <- glm(trt ~ strat1+strat2+strat3+IHC,

```

```

        family = binomial(link="logit"),
        data=cns2)$fitted.values
w <- (cns2$trt==1) * (1/Pr1) + (cns2$trt==0) * (1)/(1-Pr1)

m2<-cumulative_incidence(time,status,censor,event, trt, w=w)

arm0_adj<-m2 %>% subset(trt==0)
arm1_adj<-m2 %>% subset(trt==1)

rmtl0_adj<-AUC(arm0_adj$time,arm0_adj$cif,tau)
rmtl1_adj<-AUC(arm1_adj$time,arm1_adj$cif,tau)

diff_adj<-rmtl1_adj-rmtl0_adj

m2<-cumulative_incidence(time,status,censor,event=1, trt)

arm0_death<-m2 %>% subset(trt==0) %>% mutate(group='L+C_death')
%>% subset(time<=tau)
arm1_death<-m2 %>% subset(trt==1) %>% mutate(group='N+C_death')
%>% subset(time<=tau)

b0<-AUC(arm0_death$time,arm0_death$cif,tau)
b1<-AUC(arm1_death$time,arm1_death$cif,tau)

m2<-cumulative_incidence(time,status,censor,event=1, trt, w=w)

arm0_death<-m2 %>% subset(trt==0) %>% mutate(group='L+C_death')
%>% subset(time<=tau)
arm1_death<-m2 %>% subset(trt==1) %>% mutate(group='N+C_death')
%>% subset(time<=tau)

b0_adj<-AUC(arm0_death$time,arm0_death$cif,tau)
b1_adj<-AUC(arm1_death$time,arm1_death$cif,tau)

rmtl<-rbind(rmtl,
            data.frame(rmtl0=rmtl0,rmtl1=rmtl1,diff=diff,
                      rmtl0_adj=rmtl0_adj,rmtl1_adj=rmtl1_adj,
                      diff_adj=diff_adj,b0=b0,b1=b1,
                      c0=tau-b0-rmtl0,c1=tau-b1-rmtl1,

```

```

        b0_adj=b0_adj,b1_adj=b1_adj,
        c0_adj=tau-b0_adj-rmtl0_adj,
        c1_adj=tau-b1_adj-rmtl1_adj))
}

```

```

rmtl$diff_c<-rmtl$c1-rmtl$c0
rmtl$diff_c_adj<-rmtl$c1_adj-rmtl$c0_adj
rmtl$diff_b<-rmtl$b1-rmtl$b0
rmtl$diff_b_adj<-rmtl$b1_adj-rmtl$b0_adj

```

```

sapply(rmtl,mean)
sapply(rmtl,sd)
sapply(rmtl, function(x) return(quantile(x,0.025)))
sapply(rmtl, function(x) return(quantile(x,0.975)))

```