Predictive distributions of performance in future studies

# 1 Goals

In a future study of 100,000 participants, what is our predictive distribution for the following: - number individuals screened - liver cancers detected by delfi - liver cancers detected by ultrasound (US) + alpha fetal protein (AFP) - false positives leading to unnecessary follow-up procedures by these approaches - individuals with liver cancer not identified ( false negatives )

The size of the screening study is determined by the number of samples collected in a finite period of time. Studies with lower adherence to surveillance guidelines have fewer participants.

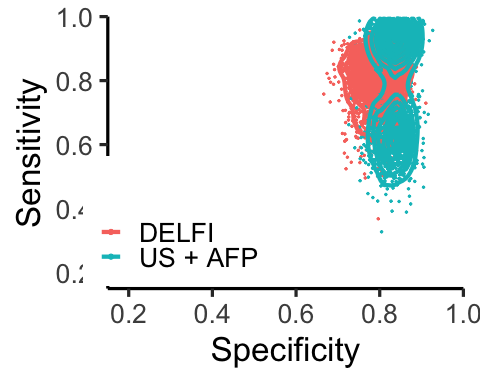
# 2 Best guesses

## 2.1 Performance of surveillance tools

find\_params <- function(i, dat){  
 dat <- dat[i, ]  
 params <- epi.betabuster(mode=dat$mode,  
 conf=0.975,  
 greaterthan=TRUE,  
 x=dat$lower,  
 max.shape1=500,  
 conf.level=0.95)  
 dat$shape1 <- params$shape1  
 dat$shape2 <- params$shape2  
 dat  
}  
params <- seq\_len(nrow(tools)) %>%  
 map\_dfr(find\_params, dat=tools)

simulate\_performance <- function(i, params, N=1000){  
 p <- params[i, ]  
 x <- rbeta(N, shape1=p$shape1, shape2=p$shape2)  
 x  
}  
simvalues <- seq\_len(nrow(tools)) %>%  
 map(simulate\_performance, params=params)  
params$simulation <- simvalues  
params2 <- params %>%  
 filter(metric != "adherence") %>%  
 pivot\_wider(id\_cols=c("tool", "population"),  
 names\_from="metric",  
 values\_from="simulation") %>%  
 unnest(c("sensitivity", "specificity"))  
adhere <- params %>%  
 filter(metric=="adherence") %>%  
 unnest(simulation) %>%  
 select(tool, population, simulation) %>%  
 set\_colnames(c("tool", "population", "adherence")) %>%  
 nest(adherence=adherence)  
params3 <- params2 %>%  
 nest(performance=c(sensitivity, specificity))  
params4 <- left\_join(params3, adhere,  
 by=c("tool", "population"))  
panelA.data <- params2 %>%  
 mutate(population=factor(population,  
 rev(c("High risk", "Early stage"))))

panelA <- panelA.data %>%  
 filter(tool %in% c("DELFI", "US + AFP")) %>%  
 ggplot(aes(specificity, sensitivity)) +  
 geom\_point(aes(color=tool), size=0.5) +  
 theme\_classic(base\_size=25) +  
 theme(panel.grid=element\_blank()) +  
 geom\_density2d(aes(color=tool), size=1.5) +  
 scale\_y\_continuous(expand=c(0, 0), limits=c(0.15, 1)) +  
 scale\_x\_continuous(expand=c(0, 0), limits=c(0.15, 1)) +  
 ##xlim(c(0.15, 1)) +  
 ##ylim(c(0.15, 1)) +  
 ##facet\_wrap(~population, ncol=1) +  
 theme(strip.background=element\_blank(),  
 legend.position=c(0.2, 0.25)) +  
 guides(color=guide\_legend(title="",  
 override.aes=list(size=1.5))) +  
 xlab("Specificity") +  
 ylab("Sensitivity")  
panelA

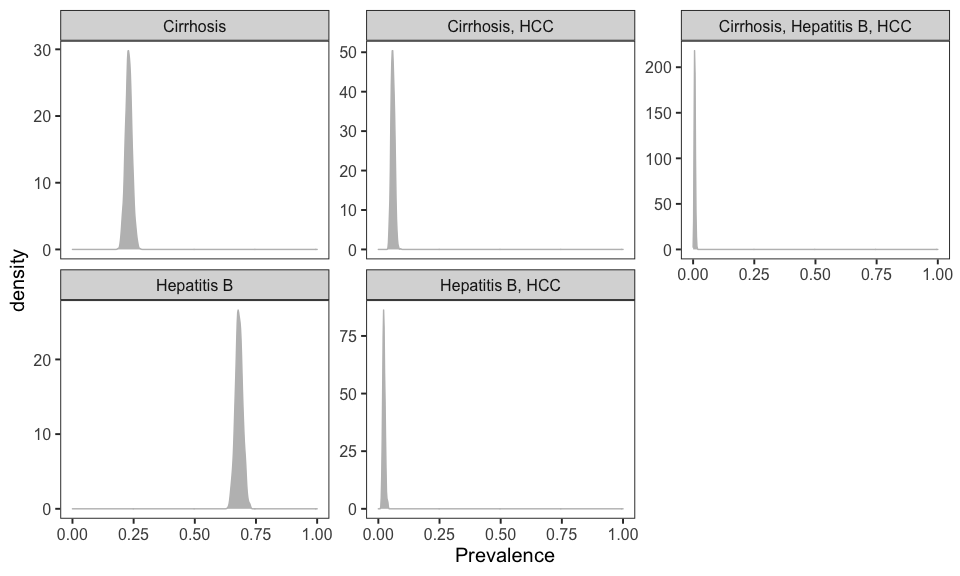


panelA <- ggplotGrob(panelA)

## 2.2 Prevalence of HCC, cirrhosis, Hep B

Prevalence is probably fairly precise depending on population

prev.wide <- rdirichlet(1000, cohorts$prevalence\*1000) %>%  
 as\_tibble() %>%  
 set\_colnames(cohorts$disease)  
prev.long <- prev.wide %>%  
 pivot\_longer(cols=all\_of(cohorts$disease), names\_to="Disease",  
 values\_to="Prevalence")  
prev.long %>%  
 ggplot(aes(Prevalence)) +  
 geom\_density(color="gray", fill="gray") +  
 theme\_bw(base\_size=15) +  
 facet\_wrap(~Disease, scales="free\_y") +  
 theme(panel.grid=element\_blank()) +  
 xlim(c(0, 1))



prev3 <- tibble(prevalence=prev.wide[["Cirrhosis, HCC"]] +  
 prev.wide[["Hepatitis B, HCC"]] +  
 prev.wide[["Cirrhosis, Hepatitis B, HCC"]])  
params4$prevalence <- rep(prev3, 6)

## 2.3 Simulations

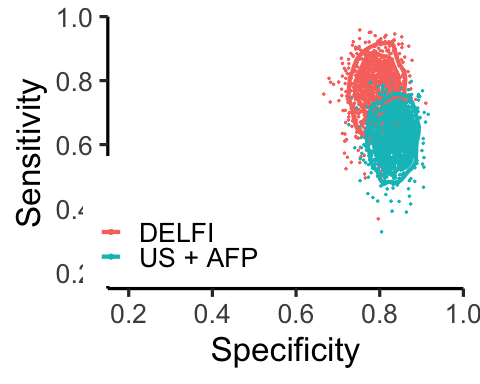
1. sample 100,000 people from high risk population (multinomial)
2. subsample based on adherence
3. calculate performance statistics for delfi and US+AFP
4. repeat 1-3 1000 times

perf <- function(i, object, N=100e3){  
 dat <- unnest(object[i, ], c("performance",  
 "adherence",  
 "prevalence"))  
 L <- nrow(dat)  
 set.seed(149491) ## use same random number seed to capture differences in parameters  
 screened <- rbinom(L, size=N, prob=dat$adherence)  
 ## prevalence  
 P <- rbinom(L, size=screened, prob=dat$prevalence) ## P = FN + TP  
 N <- screened - P ## N = TN + FP  
 TP <- rbinom(L, size=P, prob=dat$sensitivity)  
 FP <- rbinom(L, size=N, prob=(1-dat$specificity))  
 TN <- N - FP ## N = FP + TN  
 FN <- P - TP  
 fpr <- FP/N  
 fnr <- FN/P  
 tnr <- TN/N  
 tpr <- TP/P ## sensitivity  
 acc <- (TP+TN)/(P+N)  
 err <- (FP+FN)/(P+N)  
 ppv <- TP/(TP+FP)  
 npv <- TN/(TN+FN)  
 stats <- tibble("P"=P, "N"=N, "TP"=TP, "FP"=FP, "TN"=TN,  
 "FN"=FN, "acc"=acc, "err"=err,  
 "fpr"=fpr, "fnr"=fnr, "tnr"=tnr,  
 "tpr"=tpr, "ppv"=ppv, "npv"=npv,  
 "number\_screened"=screened)  
 stats2 <- bind\_cols(dat, stats)  
 stats2  
}

## for now, we only care about HCC versus not HCC  
params5 <- seq\_len(nrow(params4)) %>%  
 map\_dfr(perf, params4) %>%  
 filter(tool %in% c("DELFI", "US + AFP"))  
montecarlo <- params5

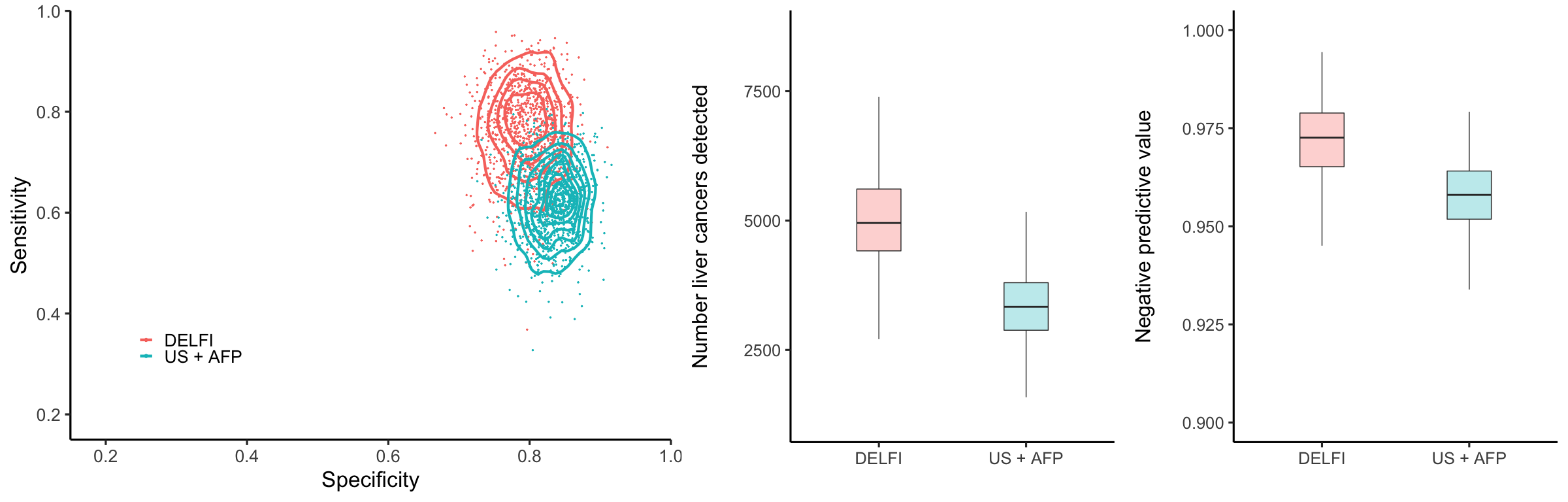
##colors <- c("steelblue", "gray")  
npvfig <-  
 params5 %>%  
 ggplot(aes(tool, npv)) +  
 geom\_boxplot(aes(fill=tool),  
 alpha=0.3, width=0.3,  
 outlier.shape=NA) +  
 theme\_classic(base\_size=24) +  
 theme(panel.grid=element\_blank()) +  
 ylab("Negative predictive value\n") +  
 xlab("") +  
 ylim(c(0.9, 1)) +  
 guides(fill="none")  
npvfig2 <- ggplotGrob(npvfig)  
  
tpfig <- params5 %>%  
 ggplot(aes(tool, TP)) +  
 geom\_boxplot(aes(fill=tool), alpha=0.3,  
 width=0.3,  
 outlier.shape=NA) +  
 ##scale\_y\_log10(limits=c(10, 1200)) +  
 theme\_classic(base\_size=24) +  
 theme(panel.grid=element\_blank(),  
 legend.position=c(0.2, 0.8)) +  
 guides(fill="none") +  
 ylab("Number liver cancers detected\n") + xlab("")  
tpfig2 <- ggplotGrob(tpfig)  
tpfig2$widths <- npvfig2$widths

params.backup <- params5  
panelA.data.backup <- panelA.data  
params5 <- filter(params5, population=="Early stage")  
panelA.data <- filter(panelA.data, population=="Early stage")  
##colors <- c("steelblue", "gray")  
npvfig <-  
 params5 %>%  
 ggplot(aes(tool, npv)) +  
 geom\_boxplot(aes(fill=tool),  
 alpha=0.3, width=0.3,  
 outlier.shape=NA) +  
 theme\_classic(base\_size=24) +  
 theme(panel.grid=element\_blank()) +  
 ylab("Negative predictive value\n") +  
 xlab("") +  
 ylim(c(0.9, 1)) +  
 guides(fill="none")  
npvfig2 <- ggplotGrob(npvfig)  
  
tpfig <- params5 %>%  
 ggplot(aes(tool, TP)) +  
 geom\_boxplot(aes(fill=tool), alpha=0.3,  
 width=0.3,  
 outlier.shape=NA) +  
 ##scale\_y\_log10(limits=c(10, 1200)) +  
 theme\_classic(base\_size=24) +  
 theme(panel.grid=element\_blank(),  
 legend.position=c(0.2, 0.8)) +  
 guides(fill="none") +  
 ylab("Number liver cancers detected\n") + xlab("")  
tpfig2 <- ggplotGrob(tpfig)  
tpfig2$widths <- npvfig2$widths  
panelA <- panelA.data %>%  
 filter(tool %in% c("DELFI", "US + AFP")) %>%  
 ggplot(aes(specificity, sensitivity)) +  
 geom\_point(aes(color=tool), size=0.5) +  
 theme\_classic(base\_size=25) +  
 theme(panel.grid=element\_blank()) +  
 geom\_density2d(aes(color=tool), size=1.5) +  
 scale\_y\_continuous(expand=c(0, 0), limits=c(0.15, 1)) +  
 scale\_x\_continuous(expand=c(0, 0), limits=c(0.15, 1)) +  
 ##xlim(c(0.15, 1)) +  
 ##ylim(c(0.15, 1)) +  
 ##facet\_wrap(~population, ncol=1) +  
 theme(strip.background=element\_blank(),  
 legend.position=c(0.2, 0.25)) +  
 guides(color=guide\_legend(title="",  
 override.aes=list(size=1.5))) +  
 xlab("Specificity") +  
 ylab("Sensitivity")  
panelA



panelA <- ggplotGrob(panelA)

widths <- c(0.5, 0.325, 0.325) %>%  
 "/"(sum(.))  
gl <- grid.layout(1, 3, widths=unit(widths, "npc"),  
 just=c("left", "bottom"))  
grid.newpage()  
pushViewport(viewport(layout=gl))  
pushViewport(viewport(layout.pos.row=1, layout.pos.col=1))  
##pushViewport(viewport(height=unit(0.5, "npc")))  
grid.draw(panelA)  
popViewport()  
pushViewport(viewport(layout.pos.row=1, layout.pos.col=2))  
grid.draw(tpfig2)  
popViewport()  
pushViewport(viewport(layout.pos.row=1, layout.pos.col=3))  
grid.draw(npvfig2)



mc <- filter(montecarlo, population=="Early stage",  
 tool %in% c("DELFI", "US + AFP"))  
nscreened <- mc %>%  
 group\_by(tool) %>%  
 summarize(mu=mean(number\_screened),  
 `0.025`=quantile(number\_screened, prob=0.025),  
 `0.975`=quantile(number\_screened, prob=0.975)) %>%  
 mutate\_at(2:4, round, 0) %>%  
 mutate\_at(2:4, prettyNum, big.mark=",") %>%  
 set\_colnames(c("tool", "mu", "q2.5", "q97.5")) %>%  
 unite("ci", c("q2.5", "q97.5"), sep="-")  
ndetect <- mc %>%  
 group\_by(tool) %>%  
 summarize(mu=mean(TP),  
 `0.025`=quantile(TP, prob=0.025),  
 `0.975`=quantile(TP, prob=0.975)) %>%  
 mutate\_at(2:4, round, 0) %>%  
 mutate\_at(2:4, prettyNum, big.mark=",") %>%  
 set\_colnames(c("tool", "mu", "q2.5", "q97.5")) %>%  
 unite("ci", c("q2.5", "q97.5"), sep="-")  
tmp <- mc %>%  
 group\_by(tool) %>%  
 select(tool, TP) %>%  
 nest()  
addl <- tibble(TP.delfi=tmp$data[[1]]$TP, TP.usafp=tmp$data[[2]]$TP) %>%  
 mutate(diff=TP.delfi-TP.usafp,  
 fold=TP.delfi/TP.usafp) %>%  
 summarize(mu=mean(diff),  
 mu.fold=mean(fold),  
 q2.5=quantile(fold, prob=0.025),  
 q97.5=quantile(fold, prob=0.975)) %>%  
 mutate\_at(1, round, 0) %>%  
 mutate\_at(2:4, round, 2) %>%  
 mutate\_at(1, prettyNum, big.mark=",") %>%  
 unite("ci", c("q2.5", "q97.5"), sep="-")  
tmp <- mc %>%  
 group\_by(tool) %>%  
 select(tool, npv) %>%  
 nest()  
npv <- mc %>%  
 group\_by(tool) %>%  
 summarize(mu=mean(npv),  
 `0.025`=quantile(npv, prob=0.025),  
 `0.975`=quantile(npv, prob=0.975)) %>%  
 mutate\_at(2:4, round, 3) %>%  
 mutate\_at(2:4, scales::percent) %>%  
 set\_colnames(c("tool", "mu", "q2.5", "q97.5")) %>%  
 unite("ci", c("q2.5", "q97.5"), sep="-")

params <- round(cohorts$prevalence\*1000, 1)  
prev <- paste(params, collapse=", ")

# 3 text for Results

To evaluate how our approach would perform for surveillance and detection in patients at high-risk for liver cancer, we evaluated the DELFI model in a theoretical population of 100,000 high-risk individuals using Monte Carlo simulations. Given the importance of detection of early-stage cancers, we focused our modeling on the detection of stage 0/A disease. We compared the DELFI approach to the current standard of care, concurrent ultrasound and AFP, and modeled the uncertainty of sensitivity and specificity of these surveillance modalities in this theoretical population through probability distributions centered at empirical estimates from our cohort or from previous reports (Singal et al., 2012, see Methods). Despite surveillance recommendations adherence in the US is low, with the most generous estimates suggesting 61% (46), resulting in an average of 60,203 individuals tested (95% CI, 41,181-76,953). As blood tests offer high accessibility and compliance, with adherence rates of 80–90% reported for blood-based biomarkers(47,48), we conservatively assumed an average of 75% (95% CI, 60-90%) of this population would be tested using the DELFI approach. Similarly, as the prevalence of cirrhosis, hepatitis B, and the co-occurrence of these co-morbidities with HCC could vary by geographic region due to differences in the ancestral composition, environmental exposures, and socioeconomic factors, we used a prior probability distribution to reflect our uncertainty of the composition of these diseases and possible regional differences. Monte Carlo simulations from these probability distributions (Methods) revealed that ultrasound and AFP detected an average of 3,348 individuals (95% CI, 2,071-4,765) with liver cancer (Supplemental Figure 14). Using DELFI, we would detect on average 1,679 additional liver cancer cases, or a 1.56-fold increase (95% CI, 0.98-2.53 fold increase) compared to ultrasound AFP alone (Supplemental Figure 14). The DELFI approach would not only substantially improve detection of liver cancer, but would be expected to increase the negative predictive value of the test (NPV) from 95.7% for ultrasound and AFP (95% CI, 93.8%-97.4%) to 97.1% for DELFI (95% CI, 94.8%-99.0%). These analyses suggest a significant population-wide benefit for using a high-specificity blood-based early detection test as a tool for the detection of liver cancer.

# 4 text for Methods

Below, I’m assuming that we will add data/simulation.xlsx as a Supplemental Table.

Monte Carlo simulations were used to compare the DELFI approach to ultrasound and AFP in a theoretical surveillance population. We used estimated 95% confidence intervals of sensitivity and specificity for DELFI and published 95% confidence intervals for US and AFP @ ref. The R package epiR was used to derive prior predictive probability distributions (beta distributions) from these confidence intervals (Carstensen B, Plummer M, Laara E, Hills M (2022). Epi: A Package for Statistical Analysis in Epidemiology. R package version 2.47, <https://CRAN.R-project.org/package=Epi>). @ author et al (year) reported that adherence to US and AFP surveillance was 61% (95% CI: 42%-79%). As other noninvasive blood-based tests have a reported adherence of more than 75% (@ ref), we assumed the probability that adherence to DELFI would be less than 60% would 2.5% or less. Using these confidence estimates, epiR was used to derive beta prior predictive distributions for adherence. We simulated multinomial probabilities for prevalence of hepatitis B, cirrhosis, hepatitis B + HCC, cirrhosis + HCC, and hepatitis B + cirrhosis + HCC from a Dirichlet with parameters 230, 680, 60, 23, 7, respectively. For a single Monte Carlo simulation for US and AFP testing, we (i) sampled the probability of adherence () from the prior predictive distribution, (ii) simulated the number of 100,000 individuals () who participated in surveillance (), (iii) sampled probabilities of co-morbidities (Dirichlet(230, 680, 60, 23, 7)), (iv) computed the prevalence of HCC (), (v) simulated HCC cases and computed the number of individuals without cancer (), (vi) sampled the sensitivity () and specificity () from the corresponding prior predictive distributions, and (vii) sampled the true positives ( and false positives (). Given the true positives and false positives, we calculated the NPV as (true negatives)/(true negatives + false negatives), where true negatives = and false negatives = . We repeated the above simulation 1000 times, obtaining a distribution of TP, FP and NPV. Using parameters for sensitivity, specificity, and adherence for the DELFI approach, we repeated the same Monte Carlo analysis to allow comparisons between these two surveillance methodologies.