Web-based supplementary material for "Optimizing Pooled Testing for Estimating the Prevalence of Multiple Diseases"

Md S. Warasi[1,*], **Laura L. Hungerford**[2], **Kevin Lahmers**[2]

[1]Department of Mathematics and Statistics, Radford University, Radford, VA 24142

[2]Virginia-Maryland College of Veterinary Medicine, Virginia Tech, Blacksburg, VA 24061

[*]*email*: msarker@radford.edu

Web Appendix A. *Additional information about the efficiency measures in Section 4.*

Pooled testing only: With only pooled responses, the relative measures exist in closed form. As discussed in Section 4, $\text{RTE}(\widehat{\mathbf{p}}_G, \widehat{\mathbf{p}}_I) = 1/k$, where k is the pool size. We herein show how the expressions of $\text{REE}(\widehat{\mathbf{p}}_G, \widehat{\mathbf{p}}_I)$ and $\text{RCE}(\widehat{\mathbf{p}}_G, \widehat{\mathbf{p}}_I)$ can be obtained.

Li et al. (2017) provided a closed-form expression of the expected information matrix $\mathcal{I}(\mathbf{p})$. For brevity, we denote the 3×3 information matrix of Li et al. (2017) as

$$\mathcal{I}(\mathbf{p}) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

Then the covariance matrix is

$$\mathcal{I}(\mathbf{p})^{-1} = \frac{1}{|\mathcal{I}(\mathbf{p})|} \begin{vmatrix} a_{22}a_{33} - a_{23}a_{32} & a_{13}a_{32} - a_{12}a_{33} & a_{12}a_{23} - a_{13}a_{22} \\ a_{23}a_{31} - a_{21}a_{33} & a_{11}a_{33} - a_{13}a_{31} & a_{13}a_{21} - a_{11}a_{23} \\ a_{21}a_{32} - a_{22}a_{31} & a_{12}a_{31} - a_{11}a_{32} & a_{11}a_{22} - a_{12}a_{21} \end{vmatrix}$$

where the determinant of $\mathcal{I}(\mathbf{p})$ is

$$|\mathcal{I}(\mathbf{p})| = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}).$$

The diagonal elements of $\mathcal{I}(\mathbf{p})^{-1}$ are the variance components $\text{var}(\widehat{p}_{00})$, $\text{var}(\widehat{p}_{10})$, and $\text{var}(\widehat{p}_{01})$, which can be easily calculated using the $\mathcal{I}(\mathbf{p})$ from Li et al. (2017). Then, for any pool size k, we find the expressions of

$$E_G[(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})] = \text{var}(\widehat{p}_{00}) + \text{var}(\widehat{p}_{10}) + \text{var}(\widehat{p}_{01})$$
$$E_G[T(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})] = T[\text{var}(\widehat{p}_{00}) + \text{var}(\widehat{p}_{10}) + \text{var}(\widehat{p}_{01})].$$

Using $k = 1$ in the above, we can find the expressions for individual testing.

Hierarchical testing: As mentioned in Section 4, for hierarchical testing, we evaluate $\text{RTE}(\widehat{\mathbf{p}}_G, \widehat{\mathbf{p}}_I)$ analytically using the explicit expressions provided in Tebbs et al. (2013). These authors also presented the EM algorithm with a Gibbs sampler, which we use in step 3 of our computation algorithm for approximating $\mathcal{I}(\mathbf{p})$ and $E_G[T(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})]$. Here we briefly describe how the approximation is performed.

Let $\widetilde{\mathbf{Y}}_i = (\widetilde{Y}_{i1}, \widetilde{Y}_{i2})'$ denote the vector of individual true statuses, for $i = 1, 2, ..., N$, where $\widetilde{Y}_{i1} = 1$ ($\widetilde{Y}_{i2} = 1$) if individual i is truly positive for *T. orientalis* (*A. marginale*) and $\widetilde{Y}_{i1} = 0$ ($\widetilde{Y}_{i2} = 0$) if individual i is truly negative for *T. orientalis* (*A. marginale*). The true statuses $\widetilde{\mathbf{Y}}_i$'s are unobservable due to potential testing error and thus regarded as missing data. Let $\mathbf{p}^{(d)}$ be the most recent estimate of $\mathbf{p}$. This estimate is updated in the M-step of the EM

algorithm using the unique solution $\mathbf{p}^{(d+1)} = (p_{00}^{(d+1)}, p_{10}^{(d+1)}, p_{01}^{(d+1)})'$, where

$$p_{00}^{(d+1)} = \frac{1}{N}\sum_{i=1}^{N} E(\widetilde{V}_{(00)i}|\mathbf{Z}, \mathbf{Y}; \mathbf{p}^{(d)})$$
$$p_{10}^{(d+1)} = \frac{1}{N}\sum_{i=1}^{N} E(\widetilde{V}_{(10)i}|\mathbf{Z}, \mathbf{Y}; \mathbf{p}^{(d)})$$
$$p_{01}^{(d+1)} = \frac{1}{N}\sum_{i=1}^{N} E(\widetilde{V}_{(01)i}|\mathbf{Z}, \mathbf{Y}; \mathbf{p}^{(d)})$$

$\widetilde{V}_{(00)i} = (1-\widetilde{Y}_{i1})(1-\widetilde{Y}_{i2})$, $\widetilde{V}_{(10)i} = \widetilde{Y}_{i1}(1-\widetilde{Y}_{i2})$, $\widetilde{V}_{(01)i} = (1-\widetilde{Y}_{i1})\widetilde{Y}_{i2}$, $\mathbf{Z}$ is a vector of all pooled test responses from stage 1, and $\mathbf{Y}$ is a vector of all individual retest responses from stage 2. For $\omega \in \{00, 10, 01\}$ and $i = 1, 2, ..., N$, the expectations $E(\widetilde{V}_{(\omega)i}|\mathbf{Z}, \mathbf{Y}; \mathbf{p}^{(d)})$ involved in $\mathbf{p}^{(d+1)}$ are approximated in the E-step using a Gibbs sampler. That is, a large number of Markov Chain Monte Carlo (MCMC) samples of $\widetilde{V}_{(\omega)i}$ are generated from its conditional distribution using the Gibbs sampler, and then the sample mean is used as an estimate of $E(\widetilde{V}_{(\omega)i}|\mathbf{Z}, \mathbf{Y}; \mathbf{p}^{(d)})$. We use 3000 MCMC samples after discarding the initial 1000 as a burn-in period. Upon convergence of the EM algorithm, $\mathbf{p}^{(d+1)}$ is taken as the MLE $\widehat{\mathbf{p}}$ and used in step 3(b) of our computational algorithm.

The observed information matrix $I(\mathbf{p})$ can be calculated by an appeal to the missing data principle and Louis's (1982) method as

$$I(\mathbf{p}) = -E\left\{\frac{\partial^2 \ln L_C(\mathbf{p}|\mathbf{Z}, \mathbf{Y}, \widetilde{\mathbf{Y}})}{\partial \mathbf{p} \partial \mathbf{p}'}\middle|\mathbf{Z}, \mathbf{Y}; \mathbf{p}\right\} - \mathrm{cov}\left\{\frac{\partial \ln L_C(\mathbf{p}|\mathbf{Z}, \mathbf{Y}, \widetilde{\mathbf{Y}})}{\partial \mathbf{p}}\middle|\mathbf{Z}, \mathbf{Y}; \mathbf{p}\right\}, \quad \text{(A.1)}$$

where $\widetilde{\mathbf{Y}}$ is a vector of individual true statuses $\widetilde{\mathbf{Y}}_i$'s and $L_C(\mathbf{p}|\mathbf{Z}, \mathbf{Y}, \widetilde{\mathbf{Y}})$ is the associated

complete likelihood function. The expectation in Equation (A.1) is

$$
\begin{aligned}
&E\left\{\left.\frac{\partial^2 \ln L_C(\mathbf{p}|\mathbf{Z},\mathbf{Y},\widetilde{\mathbf{Y}})}{\partial\mathbf{p}\partial\mathbf{p}'}\right|\mathbf{Z};\mathbf{p}\right\} \\
&\qquad = -\sum_{i=1}^{N}\begin{pmatrix} p_{00}^{-2}E(\widetilde{V}_{(00)i}|\mathbf{Z},\mathbf{Y};\mathbf{p}) & 0 & 0 \\ 0 & p_{10}^{-2}E(\widetilde{V}_{(10)i}|\mathbf{Z},\mathbf{Y};\mathbf{p}) & 0 \\ 0 & 0 & p_{11}^{-2}E(\widetilde{V}_{(01)i}|\mathbf{Z},\mathbf{Y};\mathbf{p}) \end{pmatrix} \\
&\qquad\qquad + (1-p_{00}-p_{10}-p_{01})^{-2}E(\widetilde{V}_{(11)i}|\mathbf{Z},\mathbf{Y};\mathbf{p})\mathbf{J}_3,
\end{aligned}
$$

where $\widetilde{V}_{(11)i} = \widetilde{Y}_{i1}\widetilde{Y}_{i2}$ and $\mathbf{J}_3$ is a 3×3 matrix of 1's. Again, $E(\widetilde{V}_{(\omega)i}|\mathbf{Z},\mathbf{Y};\mathbf{p})$, for $\omega \in \{00, 10, 01, 11\}$, are approximated using the Gibbs sampler. The covariance in Equation (A.1) is estimated by its sample covariance matrix using the Gibbs sampler; for more information, refer to Tebbs et al. (2013, Web Appendix B). Note that $I(\mathbf{p})$ estimated in this manner is used in step 3(a) of our computational algorithm.

Web Appendix B. *Convergence tests.*

For hierarchical testing, convergence of the sample averages in step 4 of the computational algorithm is demonstrated in Figures B.1-B.2. The results are shown with $N = 500, 1000, 2000$, pool sizes $k = 4$ (Figure B.1), 10 (Figure B.2), and $G = 20000$ repetitions. Overall, convergence is established at $G = 5000$ repetitions or faster. Thus, we use $G = 5000$ for the hierarchical testing results throughout.

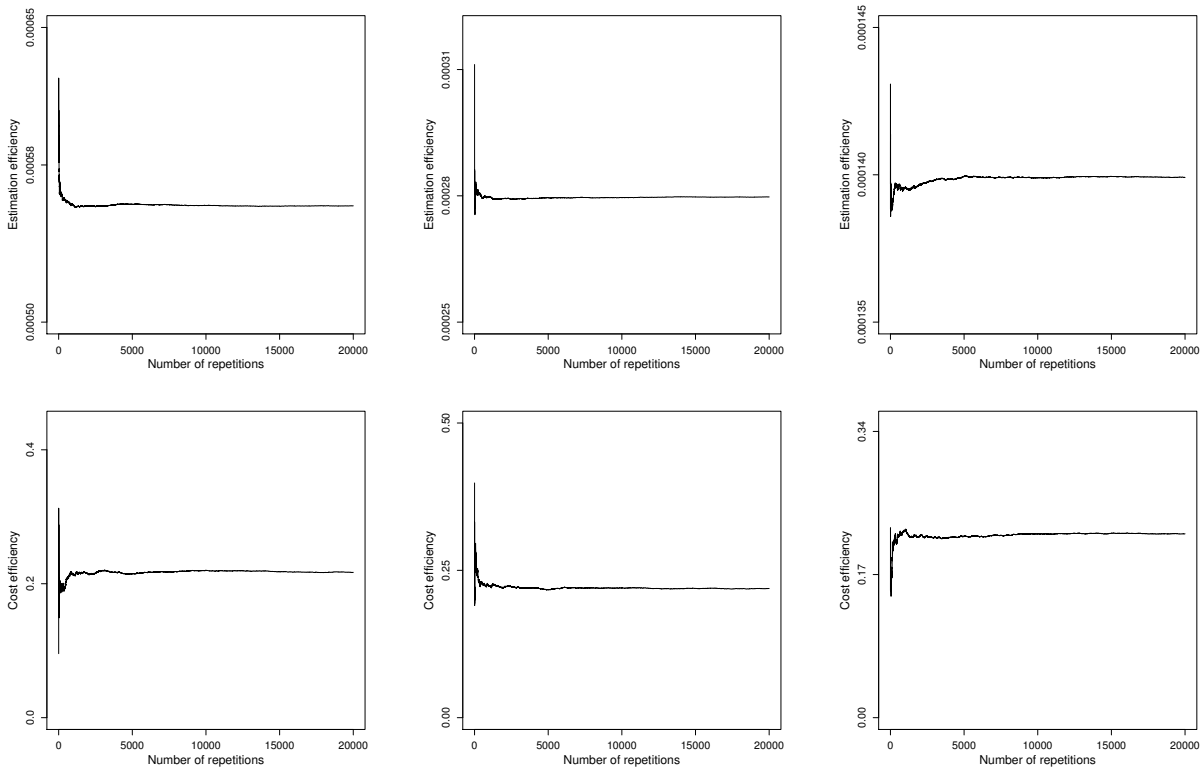

Figure B.1: Convergence of the estimates of estimation efficiency, $E[(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})]$, and cost efficiency, $E[T(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})]$, for hierarchical testing. The number of repetitions, G, is shown on the horizontal axis. Results are shown with sample size $N = 500$ (left), 1000 (middle), and 2000 (right) and pool size $k = 4$.

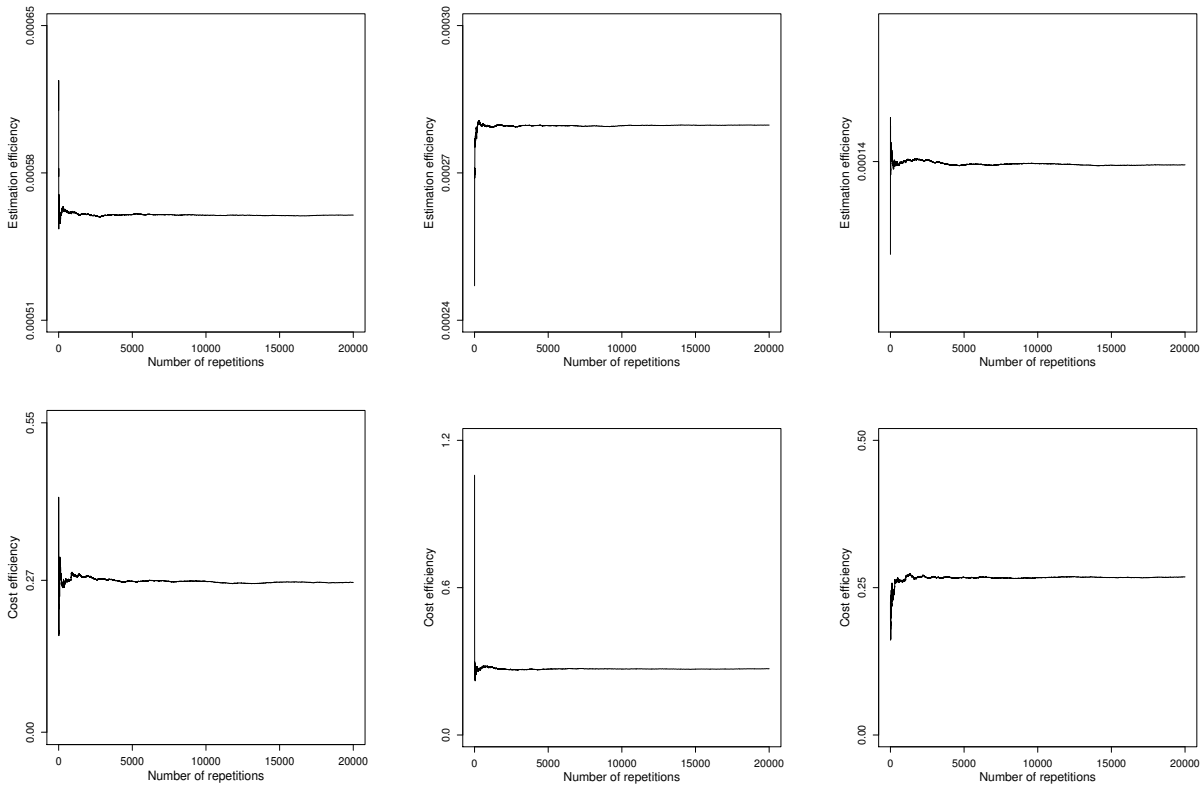

Figure B.2: Convergence of the estimates of estimation efficiency, $E[(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})]$, and cost efficiency, $E[T(\widehat{\mathbf{p}} - \mathbf{p})'(\widehat{\mathbf{p}} - \mathbf{p})]$, for hierarchical testing. The number of repetitions, G, is shown on the horizontal axis. Results are shown with sample size $N = 500$ (left), 1000 (middle), and 2000 (right) and pool size $k = 10$.

Web Appendix C. *Software installation information.*

We introduced the software application `multGT-app` in Section 4.4. We herein describe how it can be installed in a local computer. Note that the application involves our R package `multGT`, which we provide in two formats: binary package `multGT_1.0.0` (.zip file) and source package `multGT_1.0.0.tar` (.gz file). Also, we will make the application available at the free, public distribution site `https://www.shinyapps.io` that does not require any installation. A limitation of the free distribution is that the application can be used for a limited number of hours per month and can be somewhat slower in computing.

Follow the steps below to install the application in a Windows computer.

- Download and install the R (64-bit) software available at

`https://cran.r-project.org/bin/windows/base`.

- Download and install the RStudio (optional) available at `https://www.rstudio.com/products/rstudio`.
- Install either `multGT_1.0.0` or `multGT_1.0.0.tar` in R. This can be easily done in RStudio from the Tools menu.
- Create a folder, say, poolingShinyApp on your computer, and then download the application folder `multGT-app` inside poolingShinyApp.
- To launch the application, run the following code in an R console.

```
## Specify the working directory as shown in the example below:
   setwd(dir = "C:/poolingShinyApp")
## Load the multGT package:
   library(multGT)
## Install and load the shiny package:
   install.packages("shiny")
   library(shiny)
## Install and load the binGroup2 package:
   install.packages("binGroup2")
   library(binGroup2)
## Launch the application:
   runApp("multGT-app")
```

Web Appendix D. *Efficiency with misspecified values of* $\mathbf{p}$.

The efficiency results in Section 4 are calculated at the true value $\mathbf{p} = (0.834, 0.075, 0.078)'$. We now explore how much different those results become when $\mathbf{p}$ is misspecified. With this goal, we use two values of the parameter: (a) $\mathbf{p} = (0.75, 0.12, 0.11)'$ and (b) $\mathbf{p} = (0.90, 0.05, 0.04)'$.

In the first case, the prevalence of each disease is larger than the true prevalence. In the second, the prevalence of each disease is smaller. As in Section 4, we have again calculated the efficiency measures under three scenarios: (a) N is fixed, (b) T is fixed, and (c) the minimum level of precision is fixed. The overall finding from these results is expected and consistent with the pattern found in Section 4. Thus, for the sake of brevity, we did not report them. However, one can easily obtain the results using our software application.

We find that misspecification of $\mathbf{p}$ affects the optimality results, and the magnitude of the effects depends on the level of misspecification. We provide a few overall remarks to summarize the finding.

- When the prevalence of disease is smaller (i.e., the first case), pooling offers more savings in testing and more gain in estimator precision. Also, the optimal efficiency in both estimation and case identification is achieved with larger pools.
- With higher prevalence of disease and a larger pool size, more information is concealed (i.e., more loss in estimator information) due to pooling. In this situation, more tests are necessary to achieve the same level of precision in estimation.
- As found in Figure 2, the relative estimation efficiency curves for hierarchical testing remain flat; i.e., estimator precision is again independent of the pool size.