**Supplementary Information S3**

**Bayesian Latent Class Model**

We use latent class models to estimate test accuracy in the absence of a gold standard (1, 2). We adjust for the possibility of conditional dependence between ≥2 tests in ≥2 populations using Bayesian inference, as proposed by Dendukuri and Joseph [[45](#_ENREF_45)], building on previous work [[46](#_ENREF_46)] [[47](#_ENREF_47)] [[48](#_ENREF_48)].

In this analysis, we evaluate the accuracy of two methods for STH, Kato-Katz and Mini-FLOTAC, based on a sample of 525 individuals from 18 schools/communities. For 12 schools/communities, data were available from one day only, whilst for six schools tests were repeated on consecutive days increasing the number results per school to three (day 1, day 2 and combined). Here, we investigate positivity for each of the STH species individually (hookworm, *Ascaris lumbricoides* and *Trichuris trichiura)* and positivity for any STH species. When considering results from consecutive days, individuals were classified as infected if they were positive on either day, and uninfected if testing negative on both days.

For the analysis, we consider each school/community as a separate population *k* with its own (true but unobserved) infection prevalence (π*k*). Each population is subjected to two diagnostic tests, *j* (*j* = 1,2); and denote positive and negative test results from test *j,* and and denote true numbers of infected and non-infected. We define *Sj* and *C* to be the sensitivity and specificity of test *j* where and ; common sensitivities and specificities of each diagnostic test are assumed across all populations, and in the first instance are assumed to be conditionally independent. Two-day sensitivity was defined based as 1 – (1-*Sj*)\*(1-*Sj*) and two-day specificity as *Cj\*Cj.* For each population, results from each diagnostic test were cross-classified in 2x2 tables as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | ***J1*** | |
|  |  | *+* | - |
| ***J2*** | + | X*k++* | X*k+-* |
| - | X*k-+* | X*k--* |

The joint distribution of the results () of the two tests is multinomial, with the multinomial probabilities calculated as shown below using the law of total probability:

The model described above assumes that both tests are independent from one another conditional on the true disease status. However, given that they are both based upon a common biological phenomenon this assumption is unacceptable and may lead to misleading results. For example, those individuals experiencing high worm burdens are also likely to have high faecal egg counts, making infection easier to detect by either method. Dendukuri and Joseph (2001) generalized the Bayesian inference approach described above to allow for conditional dependence by directly incorporating the covariance between the two tests (ref). Specifically, the covariance between tests for infected individuals is given by , where , and for non-infected individuals by where . By this parameterization, the probability of a concordant result in an infected individual is increased by an amount , and in a non-infected individual by . Conversely, the probabilities of discordant results are penalized by the same amount. Multinomial probabilities are thus revised to include the additional parameters:

The conditional correlation between the two test outcomes for infected individuals is given as:

And for non-infected individuals as:

Finally, positive predictive value (PPV), negative predictive value (NPV) and accuracy for each test were estimated based upon overall prevalence (π), and test sensitivity and specificity:

Models were fit using MCMC simulation methods in the Winbugs 1.4.1 software (Imperial College and MRC, UK). The two sensitivities and two specificities were assumed to have independent prior distributions and the two covariances were assumed to be distributed according to generalized beta distributions, specified as uniform priors that satisfy the statements: and . Uncertainty around the values of π*k*, and the two sensitivities and two specificities were modeled using independent beta prior distributions. Reasonably informative prior information is required for at least four of the seven model parameters because of the non-identifiability of the model. In this instance, specificity of both tests is assumed to be high as experienced technicians are unlikely to mistake the shape of an egg. Thus, the beta distributions used for specificity of both tests were selected to have a mode = 0.95 and 5th percentile 0.7; informative (mode = 0.7 and a 5th percentile at 0.5) and non-informative gamma priors were used for sensitivity, and a non-informative gamma prior for . Following a burn-in of 9,000 iterations, the values for the intercept and coefficients were stored for 1,000 iterations and model convergence was assessed using diagnostic tests and by visually inspecting the time series plots. Convergence was successfully achieved after 10,000 iterations, and the model was run for a further 5,000 with thinning every five, during which predictions were made.

A sensitivity analysis was performed to investigate model assumptions and the influence of prior distributions. Model fit was compared using the DIC.

The code for the final model is given below:

**MODEL {**

for (i in 1:6) { # (for each of the schools with two days)

# results for day 1:

y1[i,1] <- (prev[i]\*(Se[1]\*Se[2] + CovSe)) + (((1-prev[i])\*(1-Sp[1])\*(1-Sp[2]) + CovSp)) # positive in both

y1[i,2] <- (prev[i]\*(Se[1]\*(1-Se[2]) - CovSe)) + (((1-prev[i])\*(1-Sp[1])\*Sp[2] - CovSp)) # positive in test1

y1[i,3] <- (prev[i]\*((1-Se[1])\*Se[2] - CovSe)) + (((1-prev[i])\*Sp[1] \* (1-Sp[2]) - CovSp)) # positive in test2

y1[i,4] <- (prev[i]\*((1-Se[1])\*(1-Se[2]) + CovSe)) + (((1-prev[i]) \*Sp[1]\*Sp[2] + CovSp)) # negative in both

result1[i,1:4] ~ dmulti(y1[i,1:4], testedb[i]) # observed data

# results for day 2:

y2[i,1] <- (prev[i]\*(Se[1]\*Se[2] + CovSe)) + (((1-prev[i])\*(1-Sp[1])\*(1-Sp[2]) + CovSp)) # positive in both

y2[i,2] <- (prev[i]\*(Se[1]\*(1-Se[2]) - CovSe)) + (((1-prev[i])\*(1-Sp[1])\*Sp[2] - CovSp)) # positive in test1

y2[i,3] <- (prev[i]\*((1-Se[1])\*Se[2] - CovSe)) + (((1-prev[i])\*Sp[1] \* (1-Sp[2]) - CovSp)) # positive in test2

y2[i,4] <- (prev[i]\*((1-Se[1])\*(1-Se[2]) + CovSe)) + (((1-prev[i]) \*Sp[1]\*Sp[2] + CovSp)) # negative in both

result2[i,1:4] ~ dmulti(y2[i,1:4], testedb[i]) # observed data

# aggregated results:

yb[i,1] <- (prev[i]\*(Se[3]\*Se[4] + CovSe2)) + (((1-prev[i])\*(1-Sp[3])\*(1-Sp[4]) + CovSp2)) # positive in both

yb[i,2] <- (prev[i]\*(Se[3]\*(1-Se[4]) - CovSe2)) + (((1-prev[i])\*(1-Sp[3])\*Sp[4] - CovSp2)) # positive in test1

yb[i,3] <- (prev[i]\*((1-Se[3])\*Se[4] - CovSe2)) + (((1-prev[i])\*Sp[3] \* (1-Sp[4]) - CovSp2)) # positive in test2

yb[i,4] <- (prev[i]\*((1-Se[3])\*(1-Se[4]) + CovSe2)) + (((1-prev[i]) \*Sp[3]\*Sp[4] + CovSp2)) # negative in both

resultb[i,1:4] ~ dmulti(yb[i,1:4], testedb[i]) # observed data

totalb[i] <- prev[i] \* testedb[i] # true infected

}

Se[3] <- (1-((1-Se[1])\*(1-Se[1]))) # Se[3] as a function of Se[1]

Sp[3] <- Sp[1]\*Sp[1]

Se[4] <- (1-((1-Se[2])\*(1-Se[2]))) # Se[4] as a function of Se[2]

Sp[4] <- Sp[2]\*Sp[2]

for (i in 1:12) { # (for each of the schools with only 1 day)

y[i,1] <- (prevA[i]\*(Se[1]\*Se[2] + CovSe)) + (((1-prevA[i])\*(1-Sp[1])\*(1-Sp[2]) + CovSp)) # positive in both

y[i,2] <- (prevA[i]\*(Se[1]\*(1-Se[2]) - CovSe)) + (((1-prevA[i])\*(1-Sp[1])\*Sp[2] - CovSp)) # positive in test1

y[i,3] <- (prevA[i]\*((1-Se[1])\*Se[2] - CovSe)) + (((1-prevA[i])\*Sp[1] \* (1-Sp[2]) - CovSp)) # postivie in test2

y[i,4] <- (prevA[i]\*((1-Se[1])\*(1-Se[2]) + CovSe)) + (((1-prevA[i]) \*Sp[1]\*Sp[2] + CovSp)) # negative in both

result[i,1:4] ~ dmulti(y[i,1:4], tested[i]) # observed data

total[i] <- prevA[i] \* tested[i] # true infected

}

sum.total <- sum(total[1:12]) + sum(totalb[1:6])

sum.tested <- sum(tested[1:12]) + sum(testedb[1:6])

Pi <- sum.total/sum.tested

for (i in 1: 4) {

PPV[i] <- (Se[i] \* Pi) / ((Se[i]\*Pi) + ((1-Sp[i])\*(1-Pi)))

NPV[i] <- (Sp[i] \* (1-Pi)) / ((Sp[i] \* (1-Pi)) + ((1 -Se[i]) \* Pi))

Accuracy[i] <- (Se[i]\*Pi) + (Sp[i]\*(1-Pi))

}

**## Priors**

for (i in 1:12) { prevA[i] ~ dbeta(1,1) }

for (i in 1:6) { prev[i] ~ dbeta(1,1) }

for (i in 1:2 ) {Se[i] ~ dbeta(13.3,6.3) }

for (i in 1:2 ) {Sp[i] ~ dbeta(21.2,2.06) }

CovSe ~ dunif(SeL, SeU)

CovSp ~ dunif(SpL, SpU)

CovSe2 ~ dunif(SeL2, SeU2)

CovSp2 ~ dunif(SpL2, SpU2)

SeL <- (Se[1]-1)\*(1-Se[2])

SeU <- min(Se[1],Se[2]) - (Se[1]\*Se[2])

SeL2 <- (Se[3]-1)\*(1-Se[4])

SeU2 <- min(Se[3],Se[4]) - (Se[3]\*Se[4])

SpL <- (Sp[1]-1)\*(1-Sp[2])

SpU <- min(Sp[1],Sp[2]) - (Sp[1]\*Sp[2])

SpL2 <- (Sp[3]-1)\*(1-Sp[4])

SpU2 <- min(Sp[3],Sp[4]) - (Sp[3]\*Sp[4])

rhoDplus <- CovSe / sqrt(Se[1] \* (1-Se[1])\*Se[2]\*(1-Se[2]))

rhoDminus <- CovSp / sqrt(Sp[1] \* (1-Sp[1])\*Sp[2]\*(1-Sp[2]))

**}**