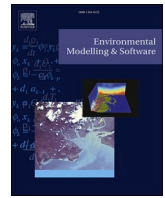


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# Environmental Modelling and Software

journal homepage: [www.elsevier.com/locate/envsoft](http://www.elsevier.com/locate/envsoft)

## PoolTestR: An R package for estimating prevalence and regression modelling for molecular xenomonitoring and other applications with pooled samples

Angus McLure<sup>a,\*</sup>, Ben O'Neill<sup>a</sup>, Helen Mayfield<sup>a,b</sup>, Colleen Lau<sup>a,b</sup>, Brady McPherson<sup>a,c</sup>

<sup>a</sup> Research School of Population Health, Australian National University, Canberra, Australia

<sup>b</sup> School of Population Health, Faculty of Medicine, University of Queensland, Brisbane, Australia

<sup>c</sup> Australian Defence Force Malaria and Infectious Disease Institute, Brisbane, Australia

### ARTICLE INFO

#### Keywords:

R  
Group testing  
Molecular xenomonitoring  
Open source software  
Pooled testing  
Mixed effect regression

### ABSTRACT

Pooled testing (also known as group testing), where diagnostic tests are performed on pooled samples, has broad applications in the surveillance of diseases in animals and humans. An increasingly common use case is molecular xenomonitoring (MX), where surveillance of vector-borne diseases is conducted by capturing and testing large numbers of vectors (e.g. mosquitoes). The R package PoolTestR was developed to meet the needs of increasingly large and complex molecular xenomonitoring surveys but can be applied to analyse any data involving pooled testing. PoolTestR includes simple and flexible tools to estimate prevalence and fit fixed- and mixed-effect generalised linear models for pooled data in frequentist and Bayesian frameworks. Mixed-effect models allow users to account for the hierarchical sampling designs that are often employed in surveys, including MX. We demonstrate the utility of PoolTestR by applying it to a large synthetic dataset that emulates a MX survey with a hierarchical sampling design.

### 1. Introduction

Pooled testing, also known as group testing, where diagnostic tests are performed on pooled samples, has broad applications for the detection of traits, defects, or diseases with low prevalence. Recently, pooled testing has been used to efficiently and rapidly test for SARS-CoV-2019 (Sunjaya and Sunjaya, 2020), but this approach has long been used for surveillance of other infectious diseases, e.g. to detect pathogens in food animals (Arnold et al., 2011) and conduct surveillance of vector-borne diseases (Pilotte et al., 2017; Rodríguez-Pérez et al., 2006). The World Health Organization has for decades been running global elimination programs to reduce the impact of many vector-borne diseases, including lymphatic filariasis (LF) (World Health Organisation, 2019). A key challenge for programs of this scale is to optimise the efficiency and accuracy of surveillance, especially as the disease becomes rarer or more localised over time. Molecular xenomonitoring (MX), the surveillance of pathogens or molecular markers in vector populations, is emerging as an alternative or adjunct to human-based surveillance of vector-borne diseases such as LF, and typically employs pooled testing of mosquitoes for filarial DNA (Lau et al., 2016; Rao et al., 2016;

Schmaedick et al., 2014; Subramanian et al., 2020). In the right setting, MX with pooled testing could potentially be very sensitive, avoid or reduce the need to test humans, and provide insights to assist in vector control strategies.

Large scale MX surveys, such as those needed to support decision making for elimination programs, involve capturing and testing large numbers of vectors. It is not always feasible to test every vector captured individually, so vectors are routinely pooled to reduce cost and improve efficiency. If using a sufficiently sensitive and specific test, each pool returns a positive result if any individual in the pool is positive and a negative result otherwise. The presence/absence of infected vectors provides a proxy measure of ongoing transmission, and could potentially help identify geographic areas where further surveillance or interventions may be required. The prevalence of infection amongst vectors provides useful information for decision makers when assessing and communicating risk and can be used to prioritise the allocation of resources, observe the effects of interventions, and identify spatial and temporal trends. However, estimating prevalence from pooled samples requires specialised statistical methods (Chen and Swallow, 1990; Farrington, 1992; Hepworth, 2005; Walter et al., 1980).

\* Corresponding author. 62 Mills Road, Acton, ACT 2611, Australia.  
E-mail address: [angus.mclure@anu.edu.au](mailto:angus.mclure@anu.edu.au) (A. McLure).

<https://doi.org/10.1016/j.envsoft.2021.105158>

Accepted 28 July 2021

Available online 2 August 2021

1364-8152/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Many MX surveys employ a hierarchical sampling structure. For instance, to conduct a MX study in a particular region, one may select a number of representative villages in the region, followed by a number of representative sites within each village. While hierarchical sampling designs like this provide an effective means of collecting a representative sample of vectors across the study area, they call for specialised analytical methods to accurately estimate infection prevalence. Analytical methods that do not account for hierarchical sampling structures will tend to underestimate the uncertainty in prevalence when applied to data collected using these sampling methods (Birkner et al., 2013).

There are a number of extant software packages for working with pooled testing models — e.g. PoolScreen (Katholi and Unnasch, 2006), the excel add-in *PooledInfRate* (Biggerstaff, 2009), and the R packages pooling (Van Domelen, 2020), binGroup (Zhang et al., 2018), and its successor binGroup2 (Hitt et al., 2020). PoolScreen is a stand-alone application that has been used in many MX projects and provides a graphical user interface for estimating prevalence in both frequentist and Bayesian paradigms. However, PoolScreen is impractical for very large and complex surveys where one may wish to estimate prevalence for many subsets of the data (e.g. estimating prevalence by vector species, by country/region/village, by sampling year, etc.). The Microsoft Excel add-in, *PooledInfRate*, is able to automate some of these steps, however neither PoolScreen nor *PooledInfRate* have functionality for regression modelling. The R package pooling is designed primarily for case-control studies with pooled assays and not applicable to MX studies. The R packages, binGroup and its successor binGroup2, provide tools for fixed-effect regression modelling with pooled data, though only in a frequentist framework. However, none of these software have functionality to account for hierarchical sampling frames which are common in large-scale MX surveys and therefore will tend to underestimate the uncertainty in prevalence estimates associated with the sampling design. The lack of a tool that is tailored for large-scale hierarchical MX surveys limits the efficiency of data analyses and the amount of information to be gleaned from these studies, particularly in resource-poor settings with limited technical capacity.

To fill this gap, we developed PoolTestR, a package for the R language (R Core Team, 2020) which provides a user-friendly and extensible framework for estimating prevalence with pooled data, and performing fixed and mixed-effect regression modelling for both hierarchical and non-hierarchical surveys. All analyses can be conducted in frequentist or Bayesian frameworks. Though our package can be applied to generic pooled testing datasets, we demonstrate its use through examples based on simulated MX data with known prevalence.

## 2. Pooled testing and the pooled binomial GLMM

Suppose we have a molecular marker of infection in a population, with unknown prevalence  $p$ . We can estimate the prevalence by taking a random sample of binary outcomes showing whether or not the molecular marker is present in a sampled unit (e.g. testing each mosquito caught). If the molecular marker has a low prevalence in the population, and the unit cost of each test is much more than the unit cost of procuring samples (e.g. trapping mosquitoes), pooled testing may be a more cost-effective means of estimating prevalence. For simplicity, assume that the test can detect the marker with 100% sensitivity and specificity. We also assume that the marker is independent across each individual in the sample (and therefore also in the pools) and that the total number of samples is much smaller than the population. While larger pools may ‘dilute’ the marker of interest and thus affect the sensitivity of the test, for the purposes of our models we will assume that the dilution is insignificant for the range of pool sizes used. Under these assumptions, for a pool of size  $s$ , the probability of a positive result is:

$$\varphi_s(p) = 1 - (1 - p)^s$$

In some cases a fixed pool size is used, however we consider the

general case where there may be a mixture of pool sizes. Suppose we have a set of observations where we sample from a population where the marker of interest has prevalence  $p$  pool them into pools of size  $s_i$  and observe the indicator outcomes  $y_i$ , with 1 indicating a positive test result and 0 indicating a negative test result. Given the pool sizes  $s$ , the outcomes  $\mathbf{y}$  is a sufficient statistic for the prevalence  $p$  and follows a “pooled Bernoulli distribution”, with probability mass function given by:

$$\begin{aligned} \text{PoolBern}(\mathbf{y}|\mathbf{s}, p) &\equiv \prod_i \text{PoolBern}(y_i|s_i, p) = \prod_i \text{Bern}(y_i|\varphi_{s_i}(p)) \\ &= (1 - p)^{\sum_i s_i(1 - y_i)} \prod_i (1 - (1 - p)^{s_i})^{y_i} \end{aligned}$$

An equivalent formulation can be made in terms of the “pooled binomial distribution” described in O’Neill and McLure (2021), which reduces the sufficient statistic of interest to the counts of cases and counts of positive outcomes in each pool size. Other than in trivial situations where all the positive pools have the same size, the maximum likelihood estimate (MLE) for  $p$  does not have a closed-form expression and is computed numerically. Under some weak regularity conditions, the standardised MLE converges in distribution to the standard normal distribution allowing for Wald confidence intervals. Other confidence intervals for  $p$  have been proposed (Hepworth, 2005).

In our package we implement a generalised linear mixed model (GLMM) using the pooled Bernoulli distribution building on mixed modelling frameworks provided by the packages *brms* and *lme4*. We model the outcomes  $\mathbf{y}$  of tests on pools of size  $\mathbf{s}$  as

$$p(\mathbf{y}|\mathbf{s}, \boldsymbol{\eta}) = \prod_i \text{PoolBern}(y_i|s_i, p_i),$$

$$\boldsymbol{\eta} = f^{-1}(\boldsymbol{\eta}),$$

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \sum_j \mathbf{Z}_j \mathbf{u}_j + \sum_k s_k(x_k) \quad \mathbf{u}_j \sim N(0, \boldsymbol{\Sigma}_j),$$

where  $f$  is a link function,  $\boldsymbol{\beta}$  are fixed/population effect coefficients,  $\mathbf{u}_j$  are random/group effect coefficients associated with grouping factor  $j$ ,  $\mathbf{X}$  is a design matrix for the fixed/population effects,  $\mathbf{Z}_j$  are design matrices for each grouping factor and  $\boldsymbol{\Sigma}_j$  are unknown covariance matrices. In the simplest case  $\boldsymbol{\Sigma}_j$  are diagonal matrices, however, in general we allow for random/group effects to be correlated between covariates. Further details of possible structuring of these random/group effects can be found in (Bürkner, 2018). The  $s_k(x_k)$  are smooth functions of covariates  $x_k$  — either splines or gaussian processes with a squared exponential kernel.

Our package accommodates two link functions for the prevalence parameter: the logistic and complementary log-log ( $\text{CLL}(p) = \log(-\log(1 - p))$ ) functions. The logistic link function produces more readily interpretable coefficients and is the default in our package. However, the CLL link function leads to a simpler mathematical exposition since  $\text{CLL}(\varphi_s(p)) = \log(s) + \text{CLL}(p)$ , allowing the separation of the pool size from the prevalence parameter, i.e. reducing the model to a GLMM with a Bernoulli response and offset of  $\log(s)$ :

$$p(\mathbf{y}|\boldsymbol{\eta}) = \prod_i \text{Bern}(y_i|\text{CLL}^{-1}(\eta_i)),$$

$$\boldsymbol{\eta} = \log(\mathbf{s}) + \mathbf{X}\boldsymbol{\beta} + \sum_j \mathbf{Z}_j \mathbf{u}_j + \sum_k s_k(x_k) \quad \mathbf{u}_j \sim N(0, \boldsymbol{\Sigma}_j).$$

Our package provides functions for fitting these models in either frequentist or Bayesian framework. In the Bayesian case, the functions in our package have flexibility for the user to input any (differentiable) prior for any of the unknown model parameters, or use default uninformative priors.

Our model extends the generalised linear model discussed in Farington (1992); our model with a logistic link function is also similar to a bird-nesting model used in Shaffer (2004). The regression models

implemented in the R package, *BinGroup2* (Hitt et al., 2020), and those presented in McMahan et al. (2017) and Joyner et al. (2020), are similar to our model but their models allow items within a single pool to have different covariates. However, the models in McMahan et al. (2017) and Joyner et al. (2020) omit the smooth terms  $s_k(x_k)$ , and *BinGroup2* can only model fixed effects.

The covariates in these models can represent any characteristic of pools. For a MX study of a mosquito-borne disease, this may include sample collection time and location, or attributes of the site where the sample is collected (e.g. interventions in place at the site, altitude, vegetation index, distance to housing). It may also include any attributes of the mosquitoes shared by the entire pool; e.g. mosquito species, in a survey design where trapped mosquitoes are sorted by species before being pooled. Mixed-effect terms can be used to account for intra-site variation not captured by other covariates in studies with hierarchical sampling frames.

### 3. The PoolTestR package

The PoolTestR package has been designed to be a simple, user-friendly and extensible way to analyse test results from pooled samples. The package has four primary functions for the estimation of prevalence: PoolPrev, HierPoolPrev, PoolReg, and PoolRegBayes. PoolPrev produces unadjusted estimates of prevalence of a marker based on the outcome of tests on pooled samples, optionally stratifying the dataset by one or more user-specified covariates. HierPoolPrev is like PoolPrev but allows users to adjust prevalence estimates for hierarchical structure in sampling frames.

PoolReg and PoolRegBayes provide flexible and extensible frameworks to fit mixed or fixed effect regression models in either a frequentist or Bayesian framework, allowing users to identify variables associated with higher prevalence or fit predictive models, while accounting for hierarchical sampling frames. Table 1 provides an overview of the differences between the four main functions. The following sections provide more details of these functions, Boxes A and B provide example code, and Figs. 1 and 2 compare the outputs of these functions when applied to a synthetic dataset.

#### 3.1. PoolPrev

PoolPrev was designed to produce comparable results to the popular stand-alone application PoolScreen (Katholi and Unnasch, 2006) for

**Table 1**

A summary of the four main functions in PoolTestR, with example function calls applied to a hypothetical dataset called Data with columns: NumInPool (number of specimens in each pool), Result (1/0 result of test for each pool), Cov1 and Cov2 (two covariate variables), and Level1 and Level2 (variables identifying sample location at two levels of the sampling frame hierarchy; e.g. village ID and site ID).

Function	Example function call	Stratified or adjusted for covariates?	Adjusted for hierarchical sampling?	Bayesian/ Frequentist	Output class
PoolPrev	PoolPrev(Data, Result, NumInPool)	Neither	No	Both	tibble
HierPoolPrev	PoolPrev(Data, Result, NumInPool, Cov1, Cov2)	Neither	Yes	Bayesian	tibble
	HierPoolPrev(Data, Result, NumInPool, c('Level1', 'Level2'))	Stratified			
PoolReg	PoolReg(Data, Result, NumInPool, Cov1, Cov2)	Adjusted	No	Frequentist	glmfit <sup>a</sup>
	PoolReg(Data, Result, NumInPool, Cov1, Cov2, (1 Level1/Level2))		Yes		glmerMod <sup>a</sup>
PoolRegBayes	PoolRegBayes(Data, Result, NumInPool, Cov1, Cov2)	Adjusted	No	Bayesian	brmsfit <sup>a</sup>
	PoolRegBayes(Data, Result, NumInPool, Cov1, Cov2, (1 Level1/Level2))		Yes		

<sup>a</sup> Applying the function getPrevalence to these outputs extracts the prevalence for each unique combination of covariates and sampling sites into a list of one or more data.frame objects.

familiarity to existing users of the software, and to enable direct comparison of results from the many studies that used PoolScreen (e.g. Helmy et al. (2004); Rodríguez-Pérez et al. (2006); Schmaedick et al. (2014); Subramanian et al. (2020); Tewari et al. (2004)). Stratifying a dataset and calculating prevalence for each subgroup of the data using PoolScreen requires many manual steps to import data, run analyses and export results. Using our function PoolPrev, this same task can be achieved in a few lines of R code with a simple syntax.

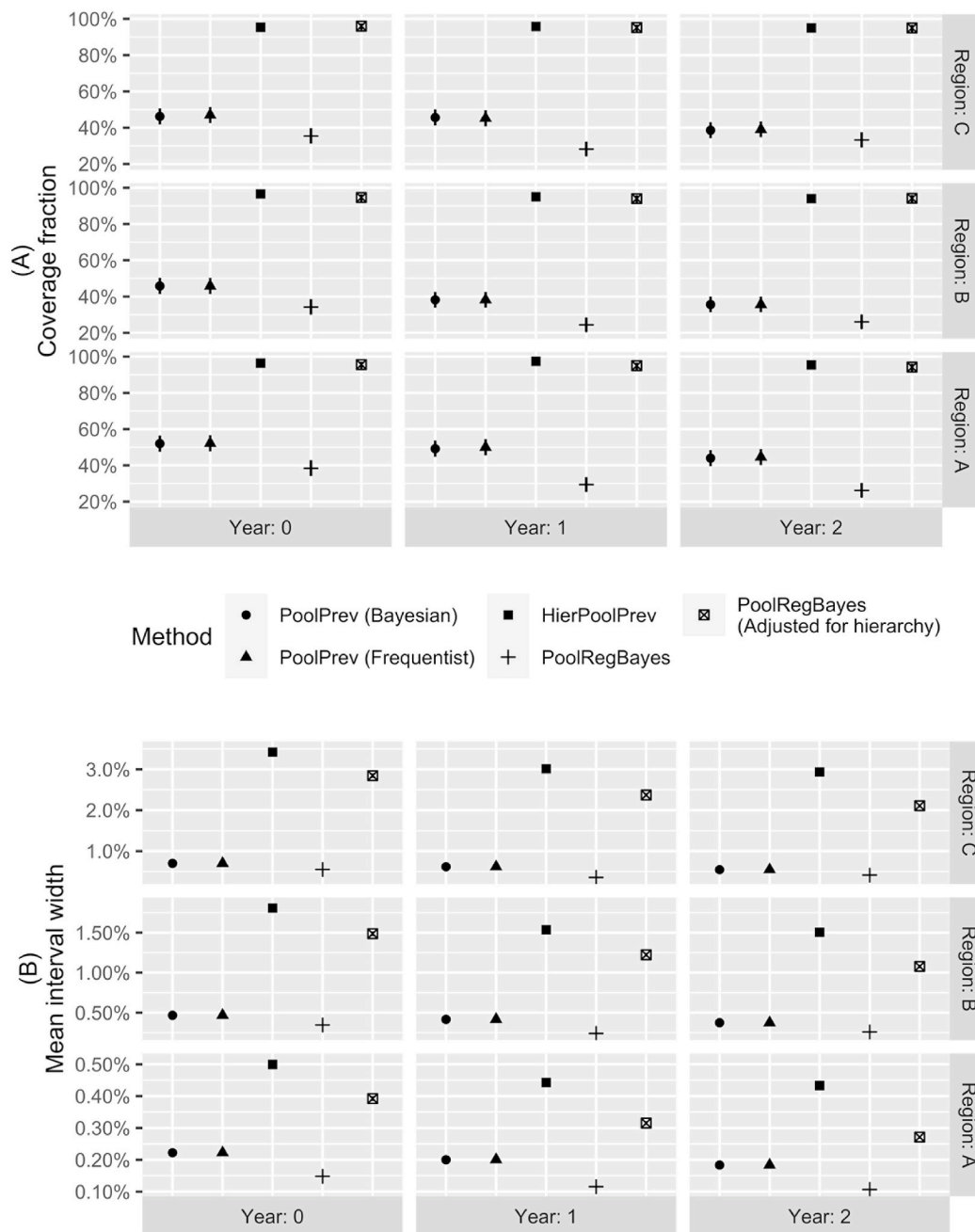
Given a dataset containing the number of samples per pool and the test results for each pool, PoolPrev returns Bayesian and maximum likelihood estimates of the prevalence together with uncertainty intervals. Efficient Bayesian inference is performed with Hamiltonian Markov Chain Monte Carlo using the Stan programming language (Stan Development Team, 2020b) and the R packages rstan (Stan Development Team, 2020a) and rstantools (Gabry et al., 2020). Users can specify their prior belief for the prevalence from the Beta distribution or use the default uninformative 'Jeffrey's' prior. Users can also optionally specify the prior probability that the marker of interest is entirely absent from the population, in which case PoolPrev also returns the probability of absence given the data. As we assume the test performed on the pooled samples does not produce false positive or negatives, the probability of absence is always zero if any of the pools test positive. In most cases the credible interval (CrI) of level  $\gamma$  (e.g. 95%) is the  $(1-\gamma)/2$  (e.g. 2.5%) and  $(1+\gamma)/2$  (e.g. 97.5%) quantiles of the posterior distribution. However, if all tests are positive the upper bound of the CrI is 1 and the lower bound is the  $1-\gamma$  (e.g. 5%) quantile of the posterior. Similarly, if all tests are negative the lower bound of the CrI is 0 and the upper bound is the  $\gamma$  (e.g. 95%) quantile of the posterior. A confidence interval of level  $\gamma$  is calculated using the likelihood ratio method (i.e. a Wilk's confidence interval). As with the Bayesian CrI, the lower or upper bound of the confidence intervals are zero or one when all pools are negative or positive, respectively.

All estimates can be optionally stratified by variables (e.g. vector species or location) by providing the name(s) of the columns in the dataset containing the variable(s). Estimation of prevalence proceed independently for each subgroup of the data defined by the variable(s).

Box A demonstrates the use of PoolPrev on a synthetic dataset (described in Section 4).

#### 3.2. HierPoolPrev

HierPoolPrev is designed to account for the hierarchical sampling

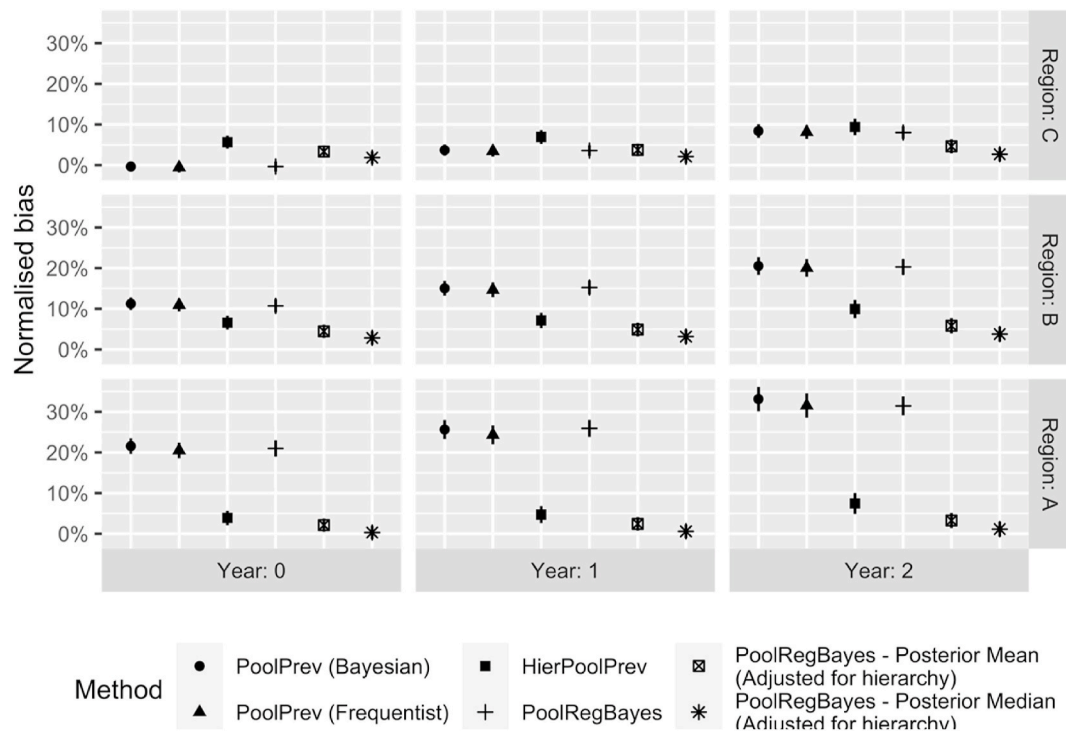


**Fig. 1.** Comparisons of the 95% confidence/credible intervals of five different methods for estimating prevalence for pooled samples. The methods were compared on 500 simulated MX surveys from a single simulated population. (A) Coverage fraction i.e. the fraction of simulated datasets for which the 95% intervals included the true value. (B) Mean interval width: i.e. the mean difference between the upper and lower bounds of the 95% intervals. Only methods that account for the hierarchical nature of the sampling frame (solid or hatched square) achieved the nominal coverage fraction (95%), but these methods also had wider intervals. R code for each approach can be found in [Boxes A and B](#). The function PoolPrev (circles and triangles) uses the same underlying methodology as the software PoolScreen (Katholi and Unnasch, 2006).

structures that are common in MX studies. It assumes that samples were taken from a number of sites across the study area, and these sites can be nested within one or more hierarchical levels (e.g. sites within villages, villages within regions, regions within provinces, etc.). HeirPoolPrev estimates prevalence by fitting an intercept-only hierarchical generalised linear mixed model with a logistic link function. The syntax and outputs are very similar to PoolPrev. There is only one additional argument, 'hierarchy', which requires the user to list the variables that encode the hierarchical structure of the sampling frame (e.g. the names of columns containing site IDs, village IDs etc.). The output provides the

Bayesian posterior mean and CrI for the prevalence, but unlike PoolPrev does not provide frequentist outputs (i.e. maximum likelihood estimates or likelihood ratio confidence intervals). As with PoolPrev, users can specify their prior belief for the prevalence and specify variables that stratify the dataset into subgroups. If subgroups of the data are specified, estimation of prevalence and random effect variances proceed independently for each subgroup.

[Box A](#) demonstrates the use of HierPoolPrev on a synthetic dataset (described in Section 4).



**Fig. 2.** The bias of six different methods for determining prevalence from pooled samples. The methods were compared on 500 simulated MX surveys from a single simulated population. The bias has been divided by the true prevalence (normalised) to allow comparison between years and regions. All methods were somewhat positively biased and relative bias was highest where prevalence was lowest (region A and year 2) but methods that accounted for the hierarchical nature of the sampling frame (solid square, hatched square, and star) achieved consistently low bias across regions and years. R code for each approach can be found in [Boxes A and B](#). The function PoolPrev (circles and triangles) uses the same underlying methodology as the software PoolScreen (Katholi and Unnasch, 2006).

### 3.3. Regression - PoolReg and PoolReg Bayes

Our package provides tools for mixed-effect regression models in both frequentist and Bayesian frameworks. PoolReg fits a frequentist mixed- or fixed-effect generalised linear model that adjusts for the sizes of pools, building on glm from the stats package (R Core Team, 2020) for fixed-effect models and the glmer function from the lme4 package (Bates et al., 2015) for mixed-effect models. For a model with only fixed effects the output is an S3-object of class glmfit, while the output for a model with random effects is an S4-object of class glmerMod which supports that same methods (e.g. summary, predict, plot, confint, anova) as any other object returned by the glm or glmer functions. PoolRegBayes provides functionality to perform the same analyses in a Bayesian framework and returns a brmsfit object. By building on these existing statistical packages, PoolTestR leverages the extensive suite of diagnostics tools available for working with models fitted with these functions and uses paradigms that will be familiar to existing users of R. These frameworks allow for a very broad range of linear models (e.g. polynomial regression, spline regression, gaussian process models). In addition, PoolTestR includes the function getPrevalence, which provides a convenient way to extract estimates of prevalence from regression models fitted with PoolReg or PoolRegBayes. The function getPrevalence, is in many cases able to detect whether a model includes adjustments for hierarchical random/group effect terms, and automatically estimate prevalence at every level of the sampling hierarchy.

[Box B](#) applies PoolReg and PoolRegBayes to the same synthetic dataset used to demonstrate PoolPrev, estimating the trend of decline in prevalence over time. [Box C](#) shows the model summaries for these regression models.

### 4. Comparison of methods on a synthetic dataset

PoolTestR provides a number of approaches to estimate prevalence:

frequentist or Bayesian, stratifying or adjusting for covariates, adjusting for or ignoring hierarchical sampling frame (Table 1). We compare the approaches with a simulation study of 500 synthetic datasets. Each synthetic dataset emulated a large MX survey with mosquitoes sampled across three years with a realistic hierarchical sampling design (e.g. Schmaedick et al. (2014); Subramanian et al. (2020)): three regions, ten randomly chosen villages per region, and ten randomly chosen sites per village, for an average of approximately 180,000 mosquitoes per dataset split across an average of 6,770 pools. We assume, as is common, that the primary purpose of the surveys is to inform interventions or assessments that will be applied at the region level — i.e. in sampled and unsampled villages. Therefore, the primary outcome of interest is the overall prevalence in each region over time. However, region-level estimates of prevalence will need to be adjusted for the hierarchical sampling frames used at the village and site level.

Each synthetic dataset was generated by simulating samples taken from across three regions (A, B, and C) in which the vectors had a low (0.5%), medium (2%), and high (4%) prevalence of the marker of interest. We then emulated a multi-level cluster survey with ten villages chosen randomly from each 'region' and traps placed at ten random sites in each village. We sampled from the same locations once a year over three years (0, 1, and 2). Prevalence was not uniform within each region or over time. At baseline (year 0), prevalence varied between villages within each region (standard deviation on the log-odds scale: 0.5), and prevalence varied between sites within each village (standard deviation on the log-odds scale: 0.5). Consequently, though the prevalence was different for each site, two sites within the same village were likely to have a more similar prevalence than two sites in different villages or two sites in different regions. On average, the prevalence was declining over time (odds ratio of 0.8 per year or equivalently a coefficient for Year of  $-0.22$  on the log-odds scale), however, the rate of change in prevalence varied between villages (standard deviation on log-odds scale: 0.2). Consequently, two sites in different villages with similar prevalence at

**Box A**

Example R code applying functions PoolPrev and HierPoolPrev to a synthetic dataset, to get prevalence estimates for the whole dataset, estimates stratified by year and/or region, and estimates with and without adjustments for hierarchical sampling.

```
#Looking at a few rows of the synthetic dataset to see structure
ExampleData[c(1:3,4001:4003),]

##      Year Region Village  Site NumInPool Result
## 1      0      A    A-1  A-1-1      25      0
## 2      0      A    A-1  A-1-1      25      0
## 3      0      A    A-1  A-1-1      25      0
## 4001    0      B    B-6  B-6-10      5      0
## 4002    1      B    B-6  B-6-10     25      1
## 4003    1      B    B-6  B-6-10     25      0

#Prevalence across the whole synthetic dataset (ignoring hierarchical sampling)
Prevs <- PoolPrev(ExampleData, Result, NumInPool)

#Prevalence for each Region (ignoring hierarchical sampling)
PrevsRegion <- PoolPrev(ExampleData, Result, NumInPool, Region)

#Prevalence for each Year (ignoring hierarchical sampling)
PrevsYear <- PoolPrev(ExampleData, Result, NumInPool, Year)

#Prevalence for each combination of Region and Year (ignoring hierarchical sampling)
PrevsYearRegion <- PoolPrev(ExampleData, Result, NumInPool, Region, Year)
PrevsYearRegion

## # A tibble: 9 x 11
##   Region Year PrevMLE  CILow  CIHigh PrevBayes  CrILow  CrIHigh ProbAbsent
## 1 A      0 0.00617 0.00511 0.00737 0.00621 0.00508 0.00740 NA
## 2 A      1 0.00461 0.00368 0.00569 0.00463 0.00364 0.00569 NA
## 3 A      2 0.00595 0.00492 0.00711 0.00598 0.00493 0.00710 NA
## 4 B      0 0.0197 0.0176 0.0220 0.0198 0.0176 0.0221 NA
## 5 B      1 0.0196 0.0175 0.0218 0.0196 0.0177 0.0217 NA
## 6 B      2 0.0172 0.0152 0.0194 0.0172 0.0153 0.0194 NA
## 7 C      0 0.0380 0.0348 0.0414 0.0380 0.0348 0.0414 NA
## 8 C      1 0.0303 0.0274 0.0334 0.0304 0.0277 0.0335 NA
## 9 C      2 0.0260 0.0235 0.0286 0.0260 0.0235 0.0286 NA
## # ... with 2 more variables: NumberOfPools <int>, NumberPositive <dbl>

#Prevalence for each Region and Year accounting for hierarchical sampling
HierPrevsYearRegion <- HierPoolPrev(ExampleData, Result, NumInPool,
                                     c("Village", "Site"), Region, Year)
#Similar to the above but stratifying down to village level
HierPrevsYearRegionVillage <- HierPoolPrev(ExampleData, Result, NumInPool,
                                           c("Site"), Region, Year, Village)
```

baseline typically had different prevalence by the third year and prevalence even went up in some villages.

We modelled the total number of mosquitoes trapped at each site and each year as independent negative binomial random variables (mean 200, dispersion 5) of vectors. Though a wide range of pool sizes may lead to better estimates of prevalence (Gu et al., 2004), we simulated a simple and practical pooling strategy similar to those used in practice (e.g. Schmaedick et al. (2014); Subramanian et al. (2020)): each year, the catches at each site were pooled into groups of 25 with an additional pool for any remainder (e.g. a catch of 53 vectors would be pooled into two pools of 25 and one pool of three). Every pool was tested once for the marker of interest using a test with perfect sensitivity and specificity.

The code for generating these synthetic datasets and the first of these datasets (accessible as ExampleData) are distributed with the PoolTestR package. The example dataset has been used to illustrate the package in Boxes A, B, and C. Box A demonstrates the use of the functions PoolPrev

and HierPoolPrev, by estimating prevalence stratified by year and region with or without adjustments for the hierarchical sampling frame. Box B demonstrates the functions PoolReg and PoolRegBayes and fits logistic-type regression models with Year and Region as covariates with and without adjustment for sampling hierarchy in frequentist and Bayesian frameworks. Box C shows the model summaries for the simple frequentist fixed-effect regression model for Region and Year, and a more complex Bayesian model with fixed/population effects for Region and Year and random/group effects for village and site. While both correctly identified that prevalence declined over the three sampling years (i.e. negative coefficient for Year), and that baseline prevalence was lowest in region A (i.e. positive coefficients for regions B and C) the mixed-effect model also estimated the degree of variation between villages and sites, resulting in differing point estimates.

Figs. 1 and 2 compare these different approaches for estimating prevalence on these simulated datasets. Since our example had adequate

**Box B**

Mixed and fixed effect regression modelling for pooled data using PoolReg and PoolRegBayes

```
# Logistic regression model - no adjustment for sampling frame hierarchy
ModFreq <- PoolReg(Result ~ Region + Year,
  ExampleData, NumInPool)
coefficients(ModFreq) # estimated model coefficients
## (Intercept)   RegionB   RegionC     Year
## -5.0555273   1.2260503   1.7476965  -0.1287181

# Logistic regression model - adjusting for sampling frame hierarchy
ModFreqHier <- PoolReg(Result ~ Region + Year + (1|Village/Site),
  ExampleData, NumInPool)

# Same as above but in a Bayesian framework
ModBayes <- PoolRegBayes(Result ~ Region + Year,
  ExampleData, NumInPool)
ModBayesHier <- PoolRegBayes(Result ~ Region + Year + (1|Village/Site),
  ExampleData, NumInPool)

# A more complex model: estimate temporal trend for each village
ModBayesHier2 <- PoolRegBayes(Result ~ Region + Year + (1 + Year|Village) + (1|Site),
  ExampleData, NumInPool)

# We can use any of these models to estimate prevalence e.g.
getPrevalence(ModFreq)
## $PopulationEffects
##   Region Year Estimate      CILow      CIHigh
## 1      A    0 0.006333634 0.005617496 0.007140413
## 2      A    1 0.005572933 0.004985287 0.006229414
## 3      A    2 0.004903144 0.004335404 0.005544817
## 4      B    0 0.021259202 0.019664559 0.022980127
## omitted 5 more rows

# For hierarchical models, getPrevalence returns prevalence at every level
getPrevalence(ModBayesHier2)
## $PopulationEffects
##   Region Year Estimate      CrILow      CrIHigh
## 1      A    0 0.005233794 0.003579397 0.007534288
## 2      A    1 0.004482318 0.003065639 0.006399554
## 3      A    2 0.003842876 0.002572213 0.005569832
## 4      B    0 0.019507027 0.013687156 0.026432815
## omitted 5 more rows
## $Village
##   Region Year Village Estimate      CrILow      CrIHigh
## 1      A    0      A-1 0.003073011 0.001690074 0.004999869
## 2      A    1      A-1 0.002665134 0.001510287 0.004283468
## 3      A    2      A-1 0.002342421 0.001232661 0.003917210
## 4      A    0      A-2 0.011871139 0.007584781 0.017769973
## omitted 86 more rows
## $Site
##   Region Year Village Site Estimate      CrILow      CrIHigh
## 1      A    0      A-1 A-1-1 0.002662369 0.0008586564 0.005724619
## 2      A    1      A-1 A-1-1 0.002310121 0.0007825112 0.005028522
## 3      A    2      A-1 A-1-1 0.002031356 0.0006330185 0.004518543
## 4      A    0      A-1 A-1-2 0.002877637 0.0010513109 0.005875640
## omitted 896 more rows

# We can also predict prevalence for new datapoints. For example, this projects
# the temporal trend forward to estimate prevalence in region A for years 3-5.
PredictData <- data.frame(Region = "A", Year = c(3,4,5))
getPrevalence(ModFreq, newdata = PredictData)
## $PopulationEffects
##   Region Year Estimate      CILow      CIHigh
## 1      A    3 0.004313506 0.003711471 0.005012704
## 2      A    4 0.003794505 0.003148847 0.004571946
## 3      A    5 0.003337741 0.002658735 0.004189427
```

**Box C**

Inspecting regression coefficients for fixed and mixed-effect regression models. Both identify the declining trend in prevalence (coefficient for year is negative), while the mixed-effect model additionally estimates the degree of variation in these coefficients between villages and sites.

```
# We can use the 'summary' function to summarise regression model coefficients, with
# standard errors, p-values, deviance and information criteria
summary(ModFreq)
## Call: stats::glm(formula = formula,
##                 family = stats::binomial(PoolLink(data[[poolSize]])),
##                 data = data)
## Deviance Residuals:
##   Min       1Q   Median       3Q      Max
## -1.3406  -0.9726  -0.5286   1.0986   2.9066
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.05553    0.06158  -82.100 < 2e-16 ***
## RegionB      1.22605    0.06639   18.468 < 2e-16 ***
## RegionC      1.74770    0.06384   27.377 < 2e-16 ***
## Year         -0.12872    0.02488   -5.173 2.3e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##   Null deviance: 9705.3  on 7603  degrees of freedom
## Residual deviance: 8524.8  on 7600  degrees of freedom
## AIC: 8532.8
##
## Number of Fisher Scoring iterations: 5

# A similar summary for a Bayesian mixed-effect model that includes MCMC convergence
# diagnostics, estimates and 95% credible intervals for fixed effect parameters and the
# standard deviations and correlations of the of mixed effects
summary(ModBayesHier2)
## Family: pool_bernoulli_logit
## Links: mu = identity
## Formula: Result | vint(NumInPool) ~ Region + Year + (1 + Year | Village) + (1 | Site)
## Data: SimData (Number of observations: 7649)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~Site (Number of levels: 300)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.47     0.04    0.40    0.55 1.00    1775    2897
##
## ~Village (Number of levels: 30)
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)    0.51     0.10    0.35    0.72 1.00    1230    1817
## sd(Year)          0.28     0.05    0.19    0.40 1.00    1215    2280
## cor(Intercept,Year) -0.33    0.22   -0.70    0.16 1.00     876    1550
##
## Population-Level Effects:
##           Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     -5.40     0.18   -5.76   -5.04 1.00    1561    1930
## RegionB        1.49     0.24    1.00    1.96 1.00    1157    1547
## RegionC        2.26     0.23    1.79    2.72 1.00    1254    1716
## Year           -0.22     0.06   -0.34   -0.10 1.00    1171    1487
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

sample size, the estimates using a frequentist framework were very similar to estimates in a Bayesian framework using non-informative priors; compare frequentist and Bayesian outputs of PoolPrev in Fig. 1. As true prevalence in the synthetic datasets was moderately variable between sites and villages, methods that did not account for hierarchical sampling frame resulted in confidence/credible intervals that included the true value for  $\leq 50\%$  of synthetic datasets. Meanwhile, methods that accounted for hierarchical sampling frames resulted in 95% intervals that included the true value for approximately 95% of simulated datasets (Fig. 1A). This difference can be seen whether stratifying other covariates (the approach in Box A), or adjusting for them (the approach in Box B). Stratifying the data by Year and Region produced estimates with wider confidence/credible intervals than in a regression framework; compare in Fig. 1B PoolPrev to PoolRegBayes (without adjustment for hierarchy), or the results of HierPoolPrev to PoolRegBayes (with adjustments for hierarchy). This effect is particularly pronounced where prevalence is low (e.g. region A). Consequently, without adjustments for hierarchical sampling, using a regression framework further reduced the fraction of the confidence/credible intervals that contain the true value (Fig. 1A). However, the regression model with adjustments for hierarchical sampling frame had the narrowest intervals that included the true value in approximately 95% of simulated datasets.

The maximum likelihood estimates of prevalence from pooled samples are known to have positive bias (Hepworth and Biggerstaff, 2017), and other regression models of pool-tested data are known to produce biased estimates (Bilder and Tebbs, 2009). Consequently, it is not surprising that the estimates of prevalence in our simulation study also exhibited positive bias (Fig. 2). The normalised bias (bias divided by true value) increased with decreasing prevalence. However, the bias was minimised in nearly all cases by adjusting for sampling frame hierarchy and adjusting for covariates rather than stratifying. In Bayesian analyses, bias was further reduced by using the posterior median rather than the posterior mean as the point estimate. Moreover, while normalised bias of the posterior median in regression analyses adjusting for sampling frame hierarchy was consistently  $<4\%$ , the bias of point estimates from models without these adjustments was sensitive to true prevalence, ranging from approximately 0% normalised bias for Region C in year 0 (true prevalence: 4.0%; mean estimated prevalence 4.0%), to approximately +32% normalised bias for region A in year 2 (true prevalence: 0.32%; mean estimated prevalence: 0.42%).

## 6. Discussion

PoolTestR is a cross-platform, user-friendly, flexible, and extensible R package for estimating prevalence and regression modelling with tests on pooled samples. PoolTestR offers substantial advantages over existing software for pooled testing such as Poolscreen (Katholi and Unnasch, 2006) and PooledInfRate (Biggerstaff, 2009) especially for hierarchical sampling designs such as those used in MX surveys. While each analysis in PoolScreen requires many manual steps to import data and export results, PoolTestR integrates with diverse ecosystem of R packages simplifying the importation of data, visualisation and exportation of results to a number of common formats (e.g. csv, xls). Existing R packages with some functionality to work with pool-tested data include BinomSamSize (Höhle, 2017), PEGroupTesting (Zhang and Li, 2016), binGroup (Zhang et al., 2018), binGroup2 (Hitt et al., 2020) and pooling (Van Domelen, 2020). BinomSamSize, can only accommodate equal sized pools and neither PEGroupTesting nor BinomSamSize has functionality for regression modelling. pooling, binGroup and its successor binGroup2 have functionality for simple regression models but cannot fit mixed-effect models. None of these software or R packages are able to account for hierarchical sampling frames. However, other authors have published mixed effect regression models for pooled data, sometimes accompanied by software (e.g. Joyner et al. (2020); McMahan et al. (2017)), however, these software have included closed-source platform-specific components, or otherwise have not been designed for

ease-of-use for non-programming specialists.

When conducting MX surveys, collecting a simple random sample of vectors across a large area is operationally infeasible. Many MX studies will therefore involve a hierarchical sampling frame involving representative sample sites distributed across the study area. If the study area and the distance between traps are smaller than the movement range of the vector being studied, it may be fair to assume that all traps are sampling from the same population and that there is no variation in prevalence between trap sites. In such cases the method implemented in Poolscreen, PooledInfRate, and the PoolPrev function in our package are appropriate for estimating prevalence. However, when aggregating data to estimate prevalence in a study area substantially larger than the typical movement range of vectors, these methods which do not account for heterogeneity between sample sites may have unreasonably narrow confidence intervals that often fail to contain the true value (Birkner et al., 2013). Instead, the function HierPoolPrev or a hierarchical mixed-effect regression model using PoolReg or PoolRegBayes should be preferred in these situations. While accounting for hierarchical sampling frames will increase the width of confidence intervals for prevalence estimates, failing to do so may result in confidence intervals which frequently fail to include the true prevalence value.

Molecular xenomonitoring surveys utilising pooled testing are often paired with human surveys utilising un-pooled testing (Pilotte et al., 2017). Though regression modelling is commonly used in the human components of these surveys (e.g. Subramanian et al. (2020)), regression modelling with pooled MX data has been hampered by the lack of suitable software; the only method for looking at differences by groups in PoolScreen is to manually stratify the data and re-run the analysis, and the regression models in binGroup2 cannot account for hierarchical sampling frames. The regression functions in the PoolTestR package fill this gap, allowing users to identify variables associated with infection (e.g. region, survey year, vector species, environmental covariates), test the statistical significance of these associations, and produce predictive models. Moreover, where appropriate, regression models can produce more precise estimates (narrower confidence intervals) compared to simple stratification. Regression models could be used for predictive prevalence mapping, however further development is required to allow for models with spatial correlation to be easily accessible to users. One limitation of the class of regression models implemented in our package is that covariates must be equal for every individual in a pool. For instance, to use our package to model possible differences between vector species in an MX study, each pool must include only vectors of a single species. The R packages, binGroup and binGroup2 can handle cases where covariates may differ between individuals in a single pool, but only for a restricted set of fixed-effect regression models. However, study designs where covariates are the same for all members of the pool allow for better estimates of prevalence and regression coefficients, so should be preferred where practical (Bilder and Tebbs, 2009). There are currently no tools that readily allow for the comparison or synthesis of both the human and MX components of surveys (e.g. model predictions of prevalence in humans based on prevalence in vectors). This functionality may be added in future releases of PoolTestR.

Maximum likelihood estimates of prevalence based on pool-tested data are known to positively biased (Bilder and Tebbs, 2009; Hepworth and Biggerstaff, 2017). A number of bias-corrected estimates have been proposed (Hepworth and Biggerstaff, 2017) and these may be incorporated in future releases of the package. Bias is not typically used to assess estimators in a Bayesian context where point estimates depend not only on the model but also on the choice of prior. However, posterior mean prevalence when using the default uninformative priors in our package will likely be positively biased in many settings. This bias can be alleviated by using the posterior median instead of the posterior mean and/or an informative prior appropriate to the study setting.

As with all models, estimates made with PoolTestR will be unreliable if the implicit assumptions about the test characteristics, sampling frame, population, or covariates are substantially violated. All the

models in our package currently assume that the tests applied to each pool have perfect sensitivity and specificity. While tests may be imperfectly sensitive or specific even when testing individual samples, test sensitivity and specificity may also decline with pool size. Statistical methods that estimate test sensitivity or specificity from the data, test for the existence of diluting effect in larger pools, or otherwise adjust for imperfect test specificity and sensitivity have been proposed (Tu et al., 1995) and may be incorporated in future versions of PoolTestR. All of our models also assume that vector catch numbers are either fixed by the sampling design or random and independent of the prevalence of the marker of interest and any modelled covariates. One common survey design is to set out traps for a fixed period of time and test all vectors trapped at each site. The relationship between vector density, transmission rate, and prevalence is dependent on complex host, agent, and environment relationships, and so there may be correlation between catch numbers and disease prevalence at a given sampling site. However, we anticipate that this kind of correlation, if not accounted for may bias estimates if sample sizes are not fixed ahead of time. While a predetermined sample size for each site could avoid this bias, it may require sampling to be prolonged at some sites and vectors to be discarded at others. The best way to detect and adjust for bias related to sampling designs that do not use a predetermined sample size remains an open question.

Another key consideration in MX studies is the appropriate sample size and pooling strategy (Katholi and Unnasch, 2006). When designing a sampling strategy using pooled samples, there is a trade-off between cost and precision. Using fewer, larger pools makes it cheaper and faster to conduct laboratory tests, but greater numbers of smaller pools improves the power of the data and the precision of estimates. For a fixed number of pools, distributing the specimens into a number of fixed size pools is likely to result in poorer estimates than using pools of various sizes (Gu et al., 2004). However, there are currently no practical rules or tools for determining an optimal or near-optimal strategies for sampling or pooling. A tool that — given a sampling design, testing constraints, and catch size — determines the optimal number of pools and the optimal distribution of samples across these pools, would further improve the cost-effectiveness of pooled MX surveys and may be incorporated in future updates of PoolTestR.

## 7. Conclusion

PoolTestR is a software package born out of the need for a simple, flexible and freely available tool to analyse large and complex datasets to estimate infection prevalence from pooled samples. PoolTestR allows users to conduct the most common analyses required for MX, whilst to being able to adjust for hierarchical sampling design and conduct a broad range of regression analyses. MX is increasingly being used as a surveillance method around the world and we hope that PoolTestR can assist researchers and program managers in disease surveillance in a range of control settings and other contexts using pooled data.

## Software availability

Name of software: PoolTestR.

Type of software: Add-on package for R.

First available: 2020.

Programming languages: R, stan.

License: GPL 3.

Code Repository: CRAN (<https://cran.r-project.org/web/packages/PoolTestR/index.html>); GitHub (<https://github.com/AngusMcLure/PoolTestR>).

Developers: Angus McLure Contact Address: [angus.mclure@anu.edu.au](mailto:angus.mclure@anu.edu.au).

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

Angus McLure was supported by an Australian Research Council Discovery Project Grant (DP180100246). Colleen Lau was supported by the National Health and Medical Research Council Fellowships (1109035 & 1193826).

This work received financial support from the Coalition for Operational Research on Neglected Tropical Diseases (COR-NTD) (Grant number OPP1053230), which is funded at The Task Force for Global Health primarily by the Bill & Melinda Gates Foundation, by the UK aid from the British government, and by the United States Agency for International Development through its Neglected Tropical Diseases Program.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

## References

- Arnold, M.E., Carrique-Mas, J.J., McLaren, I., Davies, R.H., 2011. A comparison of pooled and individual bird sampling for detection of Salmonella in commercial egg laying flocks. *Prev. Vet. Med.* 99 (2–4), 176–184.
- Bates, D., Mächler, M., Bolker, B., Walker, S., 2015. Fitting linear mixed-effects models using lme4. *J. Stat. Software* 67 (1), 1–48.
- Birkner, T., Aban, I.B., Katholi, C.R., 2013. Evaluation of a frequentist hierarchical model to estimate prevalence when sampling from a large geographic area using pool screening. *Commun. Stat. Theor. Methods* 42 (19), 3571–3595.
- Chen, C.L., Swallow, W.H., 1990. Using group testing to estimate a proportion, and to test the binomial model. *Biometrics* 46 (4), 1035.
- Farrington, C.P., 1992. Estimating prevalence by group testing using generalized linear models. *Stat. Med.* 11 (12), 1591–1597.
- Gabry, J., Goodrich, B., Lysy, M., 2020. RstanTools: Tools for Developing R Packages Interfacing with Stan.
- Gu, W., Lampman, R., Novak, R.J., 2004. Assessment of arbovirus vector infection rates using variable size pooling. *Med. Vet. Entomol.* 18 (2), 200–204.
- Hepworth, G., 2005. Confidence intervals for proportions estimated by group testing with groups of unequal size. *J. Agric. Biol. Environ. Stat.* 10 (4), 478–497.
- Hitt, B., Bilder, C., Schaarschmidt, F., Biggerstaff, B., McMahan, C., Tebbs, J., 2020. binGroup2: Identification and Estimation Using Group Testing.
- Katholi, C.R., Unnasch, T.R., 2006. Important experimental parameters for determining infection rates in arthropod vectors using pool screening approaches. *Am. J. Trop. Med. Hyg.* 74 (5), 779–785.
- Lau, C.L., Won, K.Y., Lammie, P.J., Graves, P.M., 2016. Lymphatic filariasis elimination in American Samoa: evaluation of molecular xenomonitoring as a surveillance tool in the endgame. *PLoS Neglected Trop. Dis.* 10 (11), e0005108.
- O'Neill, Ben, McLure, Angus, 2021. An examination of the generalised pooled binomial distribution and its information properties. Preprint platform: arXiv, arXiv ID: 2108.04396. <https://arxiv.org/abs/2108.04396>.
- Pilotte, N., Unnasch, T.R., Williams, S.A., 2017. The current status of molecular xenomonitoring for lymphatic filariasis and onchocerciasis. *Trends Parasitol.* 33 (10), 788–798.
- R Core Team, 2020. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
- Rao, R.U., Samarasekera, S.D., Nagodavithana, K.C., Punchihewa, M.W., Dassanayaka, T. D.M., Dg, P.K., Ford, E., Ranasinghe, U.S.B., Henderson, R.H., Weil, G.J., 2016. Programmatic use of molecular xenomonitoring at the level of evaluation units to assess persistence of lymphatic filariasis in Sri Lanka. *PLoS Neglected Trop. Dis.* 10 (5), e0004722.
- Rodríguez-Pérez, M.A., Katholi, C.R., Hassan, H.K., Unnasch, T.R., 2006. Large-scale entomologic assessment of onchocerca volvulus transmission by poolscreen PCR in Mexico the. *Am. J. Trop. Med. Hyg.* 74 (6), 1026–1033.
- Schmaedick, M.A., Koppel, A.L., Pilotte, N., Torres, M., Williams, S.A., Dobson, S.L., Lammie, P.J., Won, K.Y., 2014. Molecular xenomonitoring using mosquitoes to map lymphatic filariasis after mass drug administration in American Samoa. *PLoS Neglected Trop. Dis.* 8 (8), e3087.
- Shaffer, T.L., 2004. A unified approach to analyzing nest success. *Auk* 121 (2), 526–540.
- Stan Development Team, 2020a. RStan: the R Interface to Stan.
- Stan Development Team, 2020b. Stan Modeling Language Users Guide and Reference Manual, 2.25.
- Subramanian, S., Jambulingam, P., Krishnamoorthy, K., Sivagnaname, N., Sadanandane, C., Vasuki, V., Palaniswamy, C., Vijayakumar, B., Srividya, A., Raju, H.K.K., 2020. Molecular xenomonitoring as a post-MDA surveillance tool for global programme to eliminate lymphatic filariasis: field validation in an evaluation unit in India. *PLoS Neglected Trop. Dis.* 14 (1), e0007862.

- Sunjaya, A.F., Sunjaya, A.P., 2020. Pooled testing for expanding COVID-19 mass surveillance. *Disaster Med. Public Health Prep.* 14 (3), e42–e43.
- Tu, X.M., Litvak, E., Pagano, M., 1995. On the informativeness and accuracy of pooled testing in estimating prevalence of a rare disease: application to HIV screening. *Biometrika* 82 (2), 287–297.
- Van Domelen, D.R., 2020. Pooling: Fit Poolwise Regression Models.
- Walter, S.D., Hildreth, S.W., Beaty, B.J., 1980. Estimation of infection rates in populations of organisms using pools of variable size. *Am. J. Epidemiol.* 112 (1), 124–128.
- World Health Organisation, 2019. Global programme to eliminate lymphatic filariasis: progress report. 2018. *Wkly Epidemiol Rec* 41, 457–472.
- Zhang, B., Bilder, C., Biggerstaff, B., Schaarschmidt, F., Hitt, B., 2018. binGroup: Evaluation and Experimental Design for Binomial Group Testing.