Supplement to "Malaria in Northwest India: Data analysis via partially observed stochastic differential equation models driven by Lévy noise"

Anindya Bhadra[1], Edward L. Ionides[1], Karina Laneri[2], Mercedes Pascual[2,3], Menno Bouma[4] and Ramesh C. Dhiman[5]

Departments of [1]Statistics, and [2]Ecology & Evolutionary Biology, University of Michigan, Ann Arbor, MI 48109, USA. [3]Investigator of the Howard Hughes Medical Institute. [4]Department of Infectious and Tropical Diseases, London School of Hygiene and Tropical Medicine, University of London, London, United Kingdom. [5]National Institute of Malaria Research, Sector 8, Dwarka, Delhi-110077, India.

April 27, 2011

Supplementary methods

Supplementary results

Supplementary methods

This section documents the exact implementation of iterated filtering used by Bhadra et al. (2011) as well as providing some additional heuristic insights into the methodology. The material in this supplement draws on previously published sources (Ionides et al., 2006; King et al., 2008; Bretó et al., 2009).

Algorithm S-1 presents the iterated filtering algorithm used for our calculations, via its implementation in the pomp software package (King et al., 2009). The unobserved Markov state process $\{X(t)\}$ in Algorithm S-1 takes values in $\mathbb{R}^{d_x}$. $\{X(t)\}$ can be defined in continuous time but we require only that we can simulate its increments over an arbitrary discrete time interval defined by a pair of times $s < t$. Specifically, we suppose there is a function f such that the model can be written in the form

$$X(t) = f\big(X(s), s, t, \theta, W\big). \tag{1}$$

Here, W is a random quantity which is drawn independently each time the function $f(.)$ is evaluated, and $\theta \in \mathbb{R}^{d_\theta}$ is a vector of parameters we are interested in estimating. Rather than specifying a transition density, we write the evolution of $\{X(t)\}$ in the form (1) to emphasize the plug-and-play property (He et al., 2010) that only the ability to simulate from $\{X(t)\}$ is required. The form (1) gives a representation of the process model as a 'black box' which takes as inputs $X(s)$, s, t and θ, and then makes use of a random number generator to produce W and hence to output a stochastic quantity $X(t)$. We treat the initial value of the state process at some time t_0 as an additional parameter $X(t_0)$ which requires estimation. For general non-stationary processes, such as our malaria model, there is no natural alternative; this contrasts with stationary time series models for which it may be natural to take $X(t_0)$ to be a draw from the stationary distribution. One must be careful to avoid notational confusion between the initial value $X(t_0)$ (where *initial* corresponds to $t = 0$) and the parameter estimates used to initialize the iterated filtering algorithm (corresponding to $m = 1$); to negotiate this we call the latter quantities *starting values* rather than initial values in Algorithm S-1. We need to evaluate $X(t)$ only at times $t_1 < \cdots < t_N$ corresponding to the N observations $y_1, \ldots, y_N$ taking values in $\mathbb{R}^{d_y}$. We suppose that conditional on $\{X(t_k) = x_k\}_{k=1}^{n}$, y_n is drawn from a density $g(y_n|x_n, t_n, \theta)$.

The iterated filtering algorithm in Algorithm S-1 consists of two nested loops. In the outer loop ($m = 1$ to M), the model parameters are stochastically perturbed with an intensity that decreases at each iteration. These perturbations occur in steps 2, 4 and 12, and are here taken to have a normal distribution. The general theory of iterated filtering permits this normal distribution to be replaced by any family with the specified mean and variance. The inner loop ($n = 1$ to N) implements a sequential Monte Carlo filter (Doucet et al., 2001; Arulampalam et al., 2002; Cappé et al., 2007). The two loops are linked by the parameter update rules in steps 16 and 17. Loosely speaking, these update rules involve averaging the estimates resulting from using the data to select the most appropriate parameter values among the choices arising from the stochastic perturbations. Theoretically, this procedure converges under appropriate conditions to the maximum likelihood estimates (Ionides et al., 2006, 2011). In practice, successful convergence is assessed by running the algorithm with various choices of the starting parameters.

Algorithm S-1. Iterated filtering.

Model: process model $f(\cdot)$, measurement model $g(\cdot|\cdot)$, data $y_1, \ldots, y_N$, times $t_0, \ldots, t_N$.

Algorithmic parameters:
number of particles J, fixed lag L, number of iterations M,
cooling factor $0 < a < 1$, initial variance multiplier $b > 0$,
variance-covariance matrices Σ_X and Σ_θ with dimensions $d_x \times d_x$ and $d_\theta \times d_\theta$.

Scale parameter: $s > 0$.

Starting values:
starting initial state $X^{(1)}(t_0) \in \mathbb{R}^{d_x}$, starting parameter $\theta^{(1)} = \big(\theta_1^{(1)}, \ldots, \theta_{d_\theta}^{(1)}\big) \in \mathbb{R}^{d_\theta}$.

Procedure:

1. for $m = 1$ to M
2. draw $X_I(t_0, j) \sim \text{normal}\big(X^{(m)}(t_0), (sa^{m-1})^2\, \Sigma_X\big), \quad j = 1, \ldots, J$
3. set $X_F(t_0, j) = X_I(t_0, j)$
4. draw $\theta(t_0, j) \sim \text{normal}\big(\theta^{(m)}, (sa^{m-1})^2\, b\, \Sigma_\theta\big)$
5. set $\bar{\theta}(t_0) = \theta^{(m)}$
6. for $n = 1$ to N
7. set $X_P(t_n, j) = f\big(X_F(t_{n-1}, j), t_{n-1}, t_n, \theta(t_{n-1}, j), W\big)$
8. set $w(n, j) = g\big(y_n | X_P(t_n, j), t_n, \theta(t_{n-1}, j)\big)$
9. draw $k_1, \ldots, k_J$ such that $\text{Prob}[k_j = i] = w(n, i) \big/ \sum_r w(n, r)$
10. set $X_F(t_n, j) = X_P(t_n, k_j)$
11. set $X_I(t_n, j) = X_I(t_{n-1}, k_j)$
12. draw $\theta(t_n, j) \sim \text{normal}\big(\theta(t_{n-1}, k_j), (sa^{m-1})^2 (t_n - t_{n-1}) \Sigma_\theta\big)$
13. set $\bar{\theta}_i(t_n)$ to be the sample mean of $\{\theta_i(t_{n-1}, k_j), j = 1, \ldots, J\}$
14. set $V_i(t_n)$ to be the sample variance of $\{\theta_i(t_n, j), j = 1, \ldots, J\}$
15. end for
16. $\theta_i^{(m+1)} = \theta_i^{(m)} + V_i(t_1) \sum_{n=1}^{N} V_i^{-1}(t_n) \left[\, \bar{\theta}_i(t_n) - \bar{\theta}_i(t_{n-1}) \,\right]$
17. set $X^{(m+1)}(t_0)$ to be the sample mean of $\{X_I(t_L, j), j = 1, \ldots, J\}$
18. end for

Return:
parameter estimate, $\hat{\theta} = \theta^{(M+1)}$,
estimated initial values, $\hat{X}(t_0) = X^{(M+1)}(t_0)$,
estimated maximized log likelihood, $\hat{\ell} = \sum_{n=1}^{N} \log\big(\sum_j w(n, j)/J\big)$.

Table S-1. Algorithmic parameters. Inputs for the implementation of iterated filtering in this paper. Here, I_{d_x} and I_{d_θ} are the $d_x \times d_x$ and $d_\theta \times d_\theta$ identity matrices. The scale parameter was taken to be $s = 0.05$ for the profile liklihood calculations and was varied as prescribed for Algorithm S-2.

Parameter	Description	Value
J	Number of particles used for filtering	15000
L	Fixed lag units for initial value parameters	60
M	Number of filtering iterations	50
a	Cooling factor	0.95
b	Initial variance multiplier	3
Σ_X	initial state perturbation covariance	I_{d_x}
Σ_θ	parameter perturbation covariance	I_{d_θ}

The algorithm in Algorithm S-1 requires specification of certain "algorithmic parameters" controlling the computational effort spent on the sequential Monte Carlo filtering, the relative intensity of the stochastic perturbations for each parameter, the correlation between these perturbations, and a fixed lag determining a time point beyond which it is assumed there is negligible additional information about the initial value $X(t_0)$. The values of the algorithmic parameters which were used for the data analysis presented in the paper are provided in Table S-1. We used standard values for these algorithmic parameters, which have already been used previously on different models and datasets. We did not attempt to fine-tune these parameters for this particular analysis. In particular, the covariance matrices Σ_θ and Σ_X were simply taken to be the identity matrix. In order to standardize the scale of these parameters, positive parameters were log transformed and $[0, 1]$ valued parameters were logit transformed (as documented in the third column of Table S-2). In principle, there could be scope for reducing the computational effort by investigating alternative choices of the values in Table S-1. However, for our analysis it was not necessary to do so, which suggests that iterated filtering is reasonably robust to the exact values of the algorithmic parameters. In addition, we note that the conclusions of the data analysis, e.g. the computed log likelihoods or the estimated profile confidence intervals, do not depend on the choice of these algorithmic parameters once convergence has been checked.

Algorithm S-2. Random starts and sequential refinement. For the results in Figure 3 we used $R = 30$, $s_1 = 0.05$ and $s_R = 0.03$, with $\mathcal{A}$ being the hyperrectangle specified in Table S-2.

Model and algorithmic parameters for iterated filtering:
arguments passed unchanged to the iterated filtering procedure in Algorithm S-1.

Search region: $\mathcal{A} \subset \mathbb{R}^{d_x+d_\theta}$.

Refinement parameters: number of refinements R, initial scale s_1, final scale s_R.

Procedure:

1. draw $\big(X^*(t_0), \theta^*\big) \sim \text{uniform}(\mathcal{A})$
2. set $\ell^* = -\infty$ (i.e., the update in step 6 is necessarily accepted when $r = 1$)
3. for $r = 1$ to R
4. apply iterated filtering (Algorithm S-1) with the following inputs
   model: fixed
   algorithmic parameters: fixed [here, given in Table S-1]
   starting values: $X^{(1)}(t_0) = X^*(t_0),\ \theta^{(1)} = \theta^*$
   scale parameter: $s = s_1 + (s_R - s_1)(r-1)/(R-1)$
5. set $\big(\hat\theta, \hat X(t_0), \hat\ell\big)$ to be the output from iterated filtering (Algorithm S-1)
6. if $\hat\ell > \ell^*$ then set $\ell^* = \hat\ell,\ \theta^* = \hat\theta$ and $X^*(t_0) = \hat X(t_0)$
7. end for

Return:
parameter estimate, $\hat\theta = \theta^*$,
estimated initial values, $\hat X(t_0) = X^*(t_0)$,
estimated maximized log likelihood, $\hat\ell = \ell^*$.

Full details for the investigation of global optimization reported in Figure 3 are given in Algorithm S-2 and Table S-2. Algorithm S-2 implements a simple strategy of random starting points followed by a sequence of increasingly local stochastic searches. Each search is implemented as an iterated filtering procedure, which itself involves the use of increasingly local perturbations in parameter space. Thus, Algorithm S-2 is analogous to simulated quenching approaches which have been found effective modifications to simulated annealing algorithms (Ingber, 1993).

Our computations were implemented in the `pomp` package (King et al., 2009) for R (R Development Core Team, 2006) on a cluster of linux machines with Intel Xeon processors. Each filtering operation with $J = 15000$ particles took around 3 minutes. Each iterated filtering procedure with $M = 50$ took around 140 minutes. Each random start with sequential refinement took around 70 hours.

Table S-2. Ranges for the random starts and parameter transformations for the VS^{2}EI^2 model with rainfall. The first two columns give the lower and upper bound of the hyperrectangle defining the region $\mathcal{A}$ in which global optimization was investigated by multiple independent replications of Algorithm S-2. The random perturbations in steps 2, 4 and 12 of Algorithm S-1 were carried out on a transformed scale given by the third column. In addition, the initial values $[S_1]_0$, $[S_2]_0$, $[E]_0$, $[I_1]_0$ and $[I_2]_0$ were parameterized as unnormalized fractions of the total population. For example, we set $S_1(t_0) = P(t_0)[S_1]_0/\big([S_1]_0 + [S_2]_0 + [E]_0 + [I_1]_0 + [I_2]_0\big)$. The redundancy of this parameterization causes no difficulties, and the point estimates were subsequently self-normalized to add to unity.

parameter	**lower bound**	**upper bound**	**transformation (if applicable)**
$\mu_{S_2S_1}$	0.000	100.0	log
μ_{EI_1}	0.000	100.0	log
$\mu_{I_1I_2}$	0.000	100.0	log
$\mu_{I_2S_2}$	0.000	100.0	log
$\mu_{I_1S_1}$	0.000	100.0	log
β_1	-10.00	10.00	–
β_2	-10.00	10.00	–
β_3	-10.00	10.00	–
β_4	-10.00	10.00	–
β_5	-10.00	10.00	–
β_6	-10.00	10.00	–
τ	0.000	0.500	log
σ	0.000	0.500	log
ρ	0.000	1.000	logit
$q \times 10^4$	0.000	100.0	log
ψ	0.000	1.000	log
β	-10.00	10.00	–
$[S_1]_0$	0.000	1.000	logit
$[S_2]_0$	0.000	1.000	logit
$[E]_0$	0.000	1.000	logit
$[I_1]_0$	0.000	1.000	logit
$[I_2]_0$	0.000	1.000	logit
$[\lambda_1]_0 \times 10$	0.000	10.00	log
$[\lambda_2]_0 \times 10$	0.000	10.00	log
c	0.000	1.000	logit

Supplementary results

Figure S-1. Investigation of the distributed delay shape parameter. Profile likelihood plots for the reporting rate (ρ) for various choices of k show that $k = 2$ gives rise to the highest likelihood, and subsequently we considered only this choice. The profile also reveals that, at least for this parameter, the choice of k does not have a dramatic effect on the confidence interval for the parameter. In particular, the evidence that values of ρ larger than around 3% are inconsistent with the data is robust to the exact choice of k.

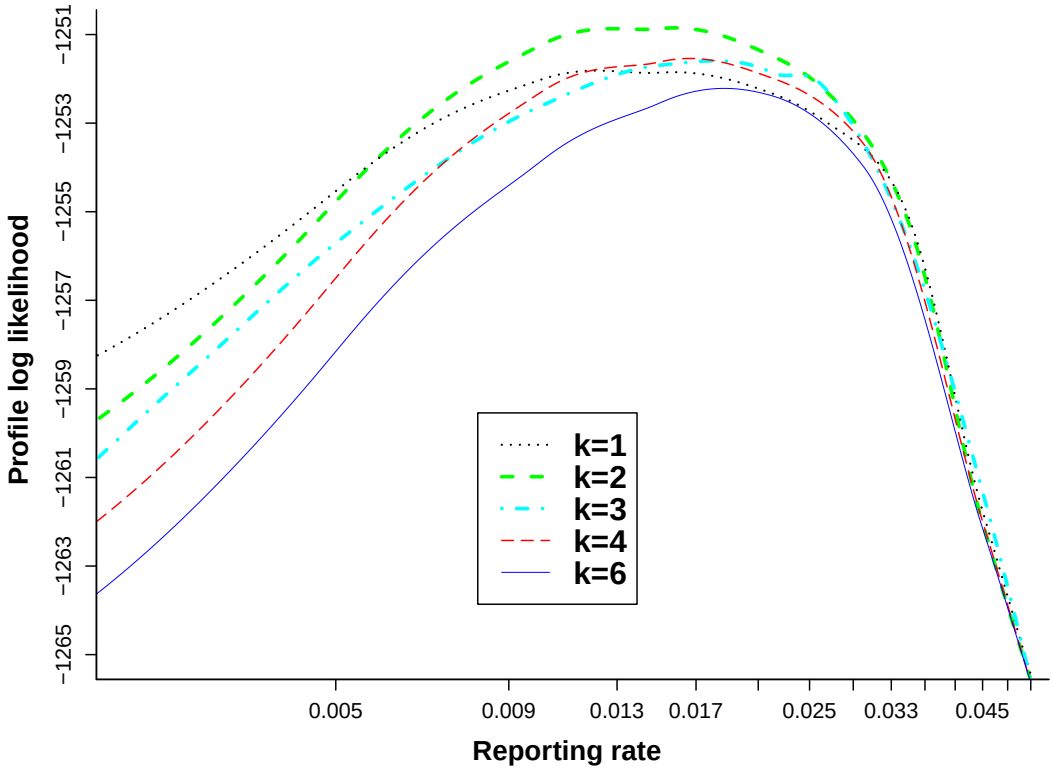
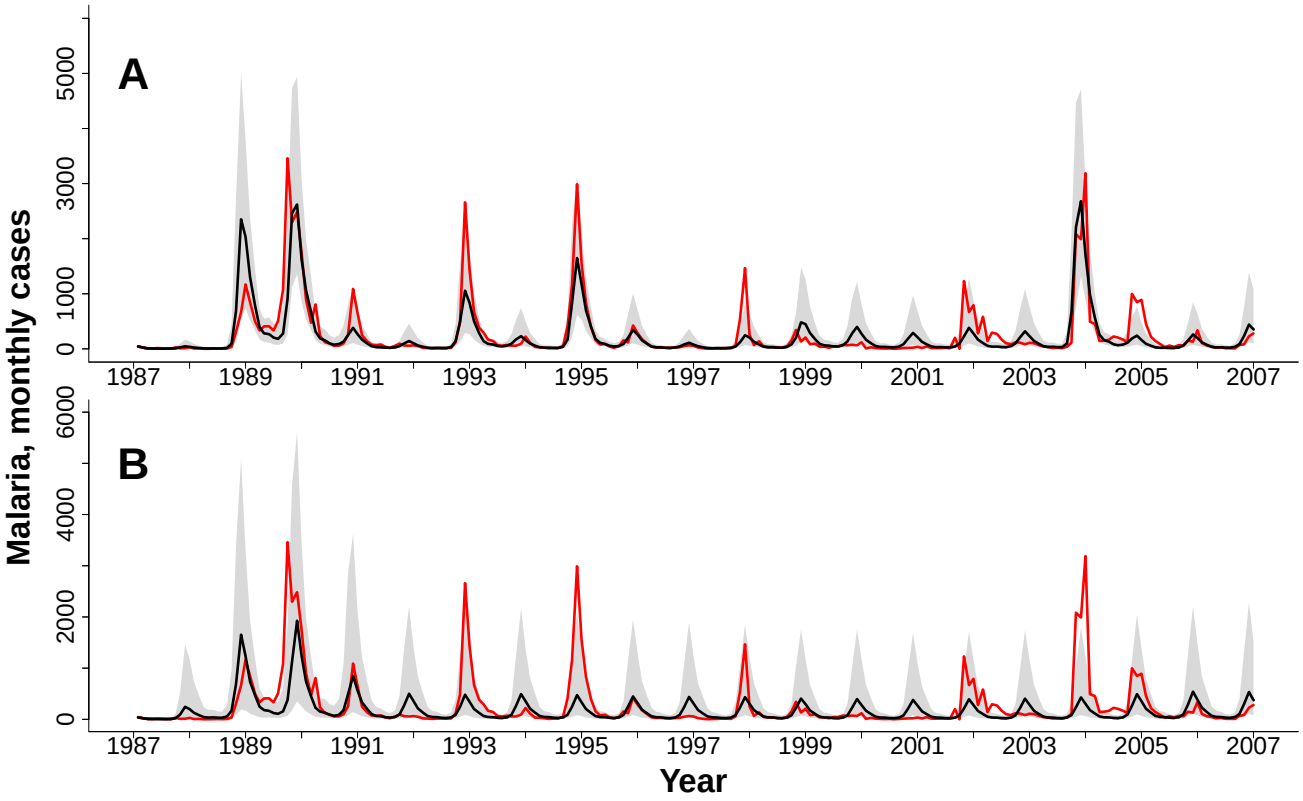


Figure S-2. Comparison of data with simulations from fitted models. Black lines show the median of ten thousand simulations; the shadowed regions corresponds to the 10% and 90% simulation percentiles. Red lines show the reported cases. (A) VS^2EI^2 model with rainfall; (B) VS^2EI^2 model without rainfall, i.e., fitted with the constraint $\beta = 0$. Note that these curves do not represent the fit of the model one time-step ahead but the numerical simulation from estimated initial conditions at the end of 1986 for the complete twenty year period, using observed rainfall values.



Supplementary References

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002), "A tutorial on particle filters for online nonlinear, non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, 50, 174 – 188.

Bhadra, A., Ionides, E. L., Laneri, K., Pascual, M., Bouma, M., and Dhiman, R. (2011), "Malaria in Northwest India: Data analysis via partially observed stochastic differential equation models driven by Lévy noise," *Journal of the American Statistical Association*, doi:10.1198/jasa.2011.ap10323.

Bretó, C., He, D., Ionides, E. L., and King, A. A. (2009), "Time series analysis via mechanistic models," *Annals of Applied Statistics*, 3, 319–348.

Cappé, O., Godsill, S., and Moulines, E. (2007), "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, 95, 899–924.

Doucet, A., de Freitas, N., and Gordon, N. J. (eds.) (2001), *Sequential Monte Carlo Methods in Practice*, New York: Springer.

He, D., Ionides, E. L., and King, A. A. (2010), "Plug-and-play inference for disease dynamics: Measles in large and small towns as a case study," *Journal of the Royal Society Interface*, 7, 271–283.

Ingber, L. (1993), "Simulated annealing: Practice versus theory," *Mathematical and Computer Modelling*, 18, 29–57.

Ionides, E. L., Bhadra, A., Atchadé, Y., and King, A. A. (2011), "Iterated Filtering," *Annals of Statistics*, Pre-published online.

Ionides, E. L., Bretó, C., and King, A. A. (2006), "Inference for nonlinear dynamical systems," *Proceedings of the National Academy of Sciences of the USA*, 103, 18438–18443.

King, A. A., Ionides, E. L., Bretó, C., Ellner, S., and Kendall, B. (2009), *pomp: Statistical inference for partially observed Markov processes*, R package: www.r-project.org.

King, A. A., Ionides, E. L., Pascual, M., and Bouma, M. J. (2008), "Inapparent infections and cholera dynamics," *Nature*, 454, 877–880.

R Development Core Team (2006), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.