Sequential Monte Carlo without likelihoods

S. A. Sisson[†‡], Y. Fan[†], and Mark M. Tanaka[§]

[†]School of Mathematics and Statistics and [§]School of Biotechnology and Biomolecular Sciences, University of New South Wales, Sydney, NSW 2052, Australia

Edited by Michael S. Waterman, University of Southern California, Los Angeles, CA, and approved December 4, 2006 (received for review August 19, 2006)

Recent new methods in Bayesian simulation have provided ways of evaluating posterior distributions in the presence of analytically or computationally intractable likelihood functions. Despite representing a substantial methodological advance, existing methods based on rejection sampling or Markov chain Monte Carlo can be highly inefficient and accordingly require far more iterations than may be practical to implement. Here we propose a sequential Monte Carlo sampler that convincingly overcomes these inefficiencies. We demonstrate its implementation through an epidemiological study of the transmission rate of tuberculosis.

approximate Bayesian computation | Bayesian inference | importance sampling | intractable likelihoods | tuberculosis

Termed approximate Bayesian computation (ABC), recent new methods in Bayesian inference have provided ways of evaluating posterior distributions when the likelihood function is analytically or computationally intractable (1–4). ABC algorithms represent a substantial methodological advance because they now admit realistic inference on problems that were considered intractable only a few years ago. The rapidly increasing use of these methods has found application in a diverse range of fields, including molecular genetics (5), ecology (6), epidemiology (7), evolutionary biology (8, 9) and extreme value theory (1).

Given a likelihood function, $f(x_0|\theta)$, and a prior distribution $\pi(\theta)$ on the parameter space, Θ, interest is in the posterior distribution $f(\theta|x_0) \propto f(x_0|\theta)\pi(\theta)$, the probability distribution of the parameters having observed the data, x_0 (10, 11).

To avoid directly evaluating the likelihood, all ABC algorithms incorporate the following procedure to obtain a random sample from the posterior distribution. For a candidate parameter vector θ^* drawn from some density, a simulated data set x^* is generated from the likelihood function $f(x|\theta^*)$ conditioned on θ^*. This vector is then accepted if simulated and observed data are sufficiently "close." Here, closeness is achieved if a vector of summary statistics $S(\cdot)$ calculated for the simulated and observed data are within a fixed tolerance (ϵ) of each other according to a distance function ρ (e.g., Euclidean distance). In this manner, ABC methods sample from the joint distribution $f(\theta, x|\rho(S(x), S(x_0)) \leq \epsilon)$, where interest is usually in the marginal $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$. The algorithms work by accepting a value θ with an average probability of $\Pr(\rho(S(x), S(x_0)) \leq \epsilon|\theta)$. If the summary statistics $S(\cdot)$ are near-sufficient and ϵ is small then $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$ should be a reasonable approximation to $f(\theta|x_0)$.

Existing ABC methods for obtaining samples from the posterior distribution either involve rejection sampling (3, 4, 12) or Markov chain Monte Carlo (MCMC) simulation (1, 2). Both of these classes of methods can be inefficient. The ABC rejection sampler proceeds as follows

ABC-REJ Algorithm

REJ1. Generate a candidate value $\theta^* \sim \pi(\theta)$ from the prior.
REJ2. Generate a data set $x^* \sim f(x|\theta^*)$.
REJ3. Accept θ^* if $\rho(S(x^*), S(x_0)) \leq \epsilon$.
REJ4. If rejected, go to REJ1.

Each accepted vector represents an independent draw from $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$. Acceptance rates for algorithm ABC-REJ can be very low as candidate parameter vectors are generated from the prior $\pi(\theta)$, which may be diffuse with respect to the posterior. Accordingly, Marjoram *et al.* (2) proposed to embed the likelihood-free simulation method within the well known MCMC framework. This algorithm proceeds as follows

ABC-MCMC Algorithm

MC1. Initialize θ_1, $i = 1$.
MC2. Generate a candidate value $\theta^* \sim q(\theta|\theta_i)$, where q is some proposal density.
MC3. Generate a data set $x^* \sim f(x|\theta^*)$.
MC4. Set $\theta_{i+1} = \theta^*$ with probability

$$\alpha = \min\left\{1, \frac{\pi(\theta^*)q(\theta_i|\theta^*)}{\pi(\theta_i)q(\theta^*|\theta_i)}\,\mathbf{1}(\rho(S(x^*), S(x_0)) \leq \epsilon)\right\},$$

otherwise set $\theta_{i+1} = \theta_i$.
MC5. If $i < N$, increment $i = i + 1$ and go to MC2.

Here $\mathbf{1}(A) = 1$ if A is true, and 0 otherwise. The candidate vector is generated from an arbitrary proposal density $q(\theta|\cdot)$ and accepted with the usual Metropolis–Hastings acceptance probability. The (intractable) likelihood ratio is now coarsely approximated by 1 if simulated and observed data are sufficiently "close," and 0 otherwise. Algorithm ABC-MCMC generates a sequence of serially and highly correlated samples from $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$. Determination of the chain length, N, is therefore obtained through a careful assessment of convergence (13) and consideration of the chain's ability to explore parameter space (i.e., chain mixing).

When the prior and posterior are dissimilar, algorithm ABC-MCMC delivers substantial increases in acceptance rates over algorithm ABC-REJ [Marjoram *et al.* (2) report 0.2% acceptance rates over 0.0008% in a simple coalescent tree analysis], although at the price of generating dependent samples. However, because acceptance rates for ABC samplers are directly proportional to the likelihood; if the ABC-MCMC sampler enters an area of relatively low probability with a poor proposal mechanism, the efficiency of the algorithm is strongly reduced because it then becomes difficult to move anywhere with a reasonable chance of acceptance, and so the sampler "sticks" in that part of the state space for long periods of time. This is illustrated in the following toy example.

Toy Example

As an illustration, suppose that the posterior of interest is given by the mixture of two normal distributions

$$f(\theta \mid x_0) = \frac{1}{2}\,\phi\left(0, \frac{1}{100}\right) + \frac{1}{2}\,\phi(0, 1),$$

where $\phi(\mu, \sigma^2)$ is the density function of a $N(\mu, \sigma^2)$ random variable. Here, the second component implies large regions of

Author contributions: S.A.S. designed research; S.A.S. and Y.F. performed research; S.A.S. and M.M.T. contributed new reagents/analytic tools; Y.F. and M.M.T. analyzed data; and S.A.S., Y.F., and M.M.T. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS direct submission.

Abbreviations: ABC, approximate Bayesian computation; MCMC, Markov chain Monte Carlo; PRC, partial rejection control; SMC, sequential Monte Carlo; ESS, effective sample size.

[‡]To whom correspondence should be addressed. E-mail: scott.sisson@unsw.edu.au.

www.pnas.org/cgi/doi/10.1073/pnas.0607208104

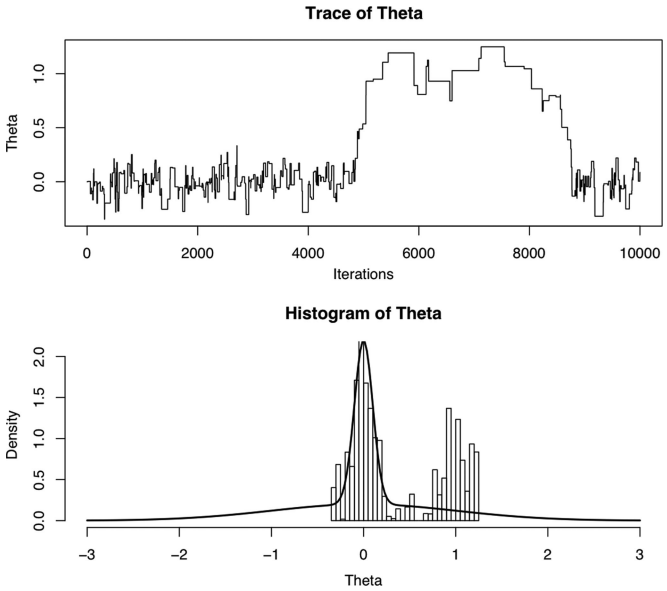

Fig. 1. Trace and histogram of θ. (*Upper*) Trace of 10,000 ABC-MCMC sampler iterations. (*Lower*) Target mixture distribution (solid line) and histogram of ABC-MCMC sampler output.

relatively low probability with respect to the lower variance first component (Fig. 1 *Lower*). In the ABC setting, this posterior may be realized by drawing $x = (x_1, \ldots, x_{100})$, $x_i \sim N(\theta, 1)$ and by specifying

$$\rho(S(x), S(x_0)) = \begin{cases} |\bar{x}| & \text{with probability } 1/2 \\ |x_1| & \text{with probability } 1/2 \end{cases},$$

where $\bar{x} = 1/100\ \Sigma\, x_i$ denotes the sample mean.

With a prior of $\pi(\theta) \sim U(-10, 10)$ and $\epsilon = 0.025$, the ABC-REJ algorithm required a mean of 400.806 data-generation steps (simulations from the likelihood) for each accepted realization based on 1,000 accepted realizations. With an acceptance rate of $<0.25\%$, this is highly inefficient.

In contrast, Fig. 1 shows the result of implementing the ABC-MCMC algorithm initialized at $\theta_0 = 0$, with $N = 10{,}000$ iterations and with proposals generated via the random walk $q(\theta|\theta_t) \sim N(\theta_t, 0.15^2)$. When the sampler is within the high-density region, transitions between different parameter values are frequent (acceptance rate $\approx 5\%$). However, when the sampler moves outside of this region, the frequency of transitions drops markedly, especially so for the extended period at $\approx$5,000–9,000 iterations. Of course, the samples should visit the distributional tails and will do so for a number of iterations proportional to the posterior probability. However, as is evident from the histogram of the sampler output, the number of further iterations required so that the realizations in the tail are in proportion to the target distribution will be far in excess of the initial 10,000.

In the search for improved ABC methods, poor Markov chain performance may be improved by embedding the target distribution as a marginal distribution within a larger family of related distributions among which it is far easier for the sampler to move around. This was essentially the approach adapted for ABC by Bortot *et al.* (1), although such algorithms are wasteful by construction in that samples from the auxiliary distributions are not used in the final analysis.

As an alternative, we propose to improve upon simple rejection sampling by adopting a sequential Monte Carlo (SMC)-based simulation approach. Here, a full population of parameters $\theta^{(1)}, \ldots, \theta^{(N)}$ (termed particles) is propagated from an initial, user-specified distribution, through a sequence of intermediary distributions, until it ultimately represents a sample from the target distribution. SMC methods can be considered an extension of importance sampling. We will demonstrate that the SMC approach can substantially outperform both MCMC and rejection sampling in the ABC framework.

Advantages of the SMC approach are

1. Like rejection sampling, the sampler will never become "stuck" in regions of low probability.
2. Unlike rejection sampling, severe inefficiencies generated by mismatch of (initial) sampling and target distributions are avoided.
3. Particles that represent the target posterior poorly are eliminated in favor of those particles that represent the posterior well.
4. The population-based nature of the sampler means that complex (e.g., multimodal) posteriors may be explored more efficiently.
5. Samples are obtained from a number of distributions with differing tolerances (ϵ). This permits an *a posteriori*, or dynamic, examination of the robustness of the posterior to this choice.
6. Unlike MCMC, SMC particles are uncorrelated and do not require the determination of a burn-in period or assessment of convergence.

Disadvantages of the SMC approach are considered in *Discussion.*

Here, we propose an ABC sampler based on SMC simulation. We initially outline a generic SMC algorithm before exploiting ideas based on PRC to derive a more efficient algorithm for the ABC setting. Finally, we demonstrate its utility with regard to the toy example considered above and in a reexamination of a previously implemented analysis of the transmission rate of tuberculosis.

Methods

SMC Without Likelihoods. We wish to sample N particles $\theta^{(1)}, \ldots, \theta^{(N)}$ from the posterior $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$, for observed data x_0, and for unknown parameter vector $\theta \in \Theta$. We assume that the summary statistics $S(\cdot)$, the tolerance ϵ, and the distance function ρ are known.

Let $\theta_1^{(1)}, \ldots, \theta_1^{(N)} \sim \mu_1(\theta)$ be an initial population from a known density, μ_1, from which direct sampling is possible, and by $f_T(\theta) = f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$ the target distribution (this notation will become clear shortly). Standard importance sampling would then indicate how well each particle $\theta_1^{(i)}$ adheres to $f_T(\theta)$ by specifying the importance weight, $W_T^{(i)} = f_T(\theta_1^{(i)})/\mu_1(\theta_1^{(i)})$, it should receive in the full population of N particles. The effectiveness of importance sampling is sensitive to the choice of sampling distribution, μ_1. The prior $\pi(\theta)$ is often used for this purpose. Importance sampling can be highly inefficient if μ_1 is diffuse relative to f_T and can fail completely in the case of sampling and target distribution mismatch.

The idea behind sequential sampling methods is to avoid the potential disparity between μ_1 and f_T by specifying a sequence of intermediary distributions $f_1, \ldots, f_{T-1}$, such that they evolve gradually from the initial distribution towards the target distribution. For example, one can choose a geometric path specification where $f_t(\theta) = f_T(\theta)^{\phi_t}\, \mu_1(\theta)^{1-\phi_t}$ with $0 < \phi_1 < \ldots < \phi_T = 1$ (14, 15). Hence, it is possible to move smoothly and effectively in sequence from μ_1 to f_T using repeated importance sampling, generating a series of particle populations $\{\theta_t^{(i)}\} = \{\theta_t^{(i)} : i = 1, \ldots, N\}$, for $t = 1, \ldots T$. That is, sequential methods proceed by moving and reweighting the particles according to how well they adhere to each successive distribution, f_t.

In the ABC setting, we may naturally define the sequence of distributions $f_1 \ldots, f_T$ as

$$f_t(\theta|\rho(S(x), S(x_0)) \leq \epsilon_t) = \frac{\pi(\theta)}{B_t} \sum_{b=1}^{B_t} \mathbf{1}(\rho(S(x_{(b)}),\ S(x_0)) \leq \epsilon_t). \quad \textbf{[1]}$$

STATISTICS

Here, $x_{(1)}, \ldots, x_{(B_t)}$ are B_t data sets generated under a fixed parameter vector, $x_{(b)} \sim f(x|\theta)$, and $\{\epsilon_t\}$ is a strictly decreasing sequence of tolerances. The nested family of distributions generated by varying ϵ (continuously) was considered by Bortot *et al.* (1) in their augmented state space ABC algorithm. By specifying $\epsilon_t < \epsilon_{t-1}$, we ensure that the likely range of parameter values in each progressive distribution is a subset of the one it precedes. This is a desirable property for our sampling distributions. By specifying $\epsilon_T = \epsilon$, we realize the final particle population $\{\theta_T^{(i)}\}$ as a weighted sample from the target distribution. Setting $B_t = B$ and $\epsilon_t = \epsilon$ for all t reduces to the likelihood specified by Marjoram *et al.* (2), and further $B = 1$ to the likelihood adopted in algorithm ABC-MCMC. The target distribution, f_T, is specified by $\epsilon_T = \epsilon$.

PRC. In ABC samplers, $B_t = 1$ is the most computationally efficient specification, in that some action occurs (e.g., a realization or move proposal is accepted) each time a nonzero likelihood is generated. However, because the particle weight under SMC methods is directly proportional to the likelihood, there is a large probability that the likelihood, and therefore the particle weight, will be zero, thereby rendering the particle useless. Fortunately, the idea of PRC (see chapters 2 and 3 of ref. 16) permits the repeated resampling (and moving) of particles from the previous population to replace those particles with zero weight. PRC continues until N particles with nonzero weight are obtained. See *Appendix* for further details.

At each step, SMC methods move each particle according to a Markov kernel K_t to improve particle dispersion. This induces a particle approximation to the importance sampling distribution $\mu_t(\theta_t) = \int_\Theta \mu_{t-1}(\theta_{t-1}) K_t(\theta_t|\theta_{t-1}) d\theta_{t-1}$, for populations $t = 2, \ldots, T$. Choices of K_t include a standard smoothing kernel (e.g., Gaussian) or a Metropolis–Hastings accept/reject step. The kernel accordingly enters the particle weight calculation.

A recent innovation in SMC methods has been the introduction of a backward Markov kernel L_{t-1} with density $L_{t-1}(\theta_{t-1}|\theta_t)$ into the weight calculation (17). The backward kernel relates to a time-reversed SMC sampler with the same target marginal distribution as the (forward-time) SMC sampler induced by K_t. Because only specification of K_t is required in order to implement an SMC sampler, the backward kernel is essentially arbitrary. The kernel L_{t-1} may therefore be optimized to minimize the variance of the weights induced by the importance sampling distribution μ_t (through K_t). This is difficult in general, however, so simplified forms are often adopted. See ref. 17 for further discussion.

SMC algorithms measure the degree of sample degeneracy within each population through the effective sample size (ESS). ESS calculates the equivalent number of random samples required to obtain an estimate, such that its Monte Carlo variation is equal to that of the N weighted particles. This may be estimated as $1 \leq [\Sigma_{i=1}^N (W_t^{(i)})^2]^{-1} \leq N$ for each t (16, 18). Sample degeneracy can occur through sampling and target distribution mismatch when a small number of particles have very large weights. Through a resampling step, particles with larger weights become better represented in the resampled population than those with smaller weights. Those particles with sufficiently small weights, which poorly approximate f_t, may be eliminated. The resampling threshold, E, is commonly taken to be $N/2$.

Combining PRC with SMC, we obtain the following (ABC-PRC) algorithm

ABC-PRC Algorithm

PRC1. Initialize $\epsilon_1, \ldots, \epsilon_T$, and specify initial sampling distribution μ_1.
Set population indicator $t = 1$.

PRC2. Set particle indicator $i = 1$.

PRC2.1. If $t = 1$, sample $\theta^{**} \sim \mu_1(\theta)$ independently from μ_1.
If $t > 1$, sample θ^* from the previous population $\{\theta_{t-1}^{(i)}\}$ with weights $\{W_{t-1}^{(i)}\}$, and perturb the particle to $\theta^{**} \sim K_t(\theta|\theta^*)$ according to a Markov transition kernel K_t.
Generate a data set $x^{**} \sim f(x|\theta^{**})$.
If $\rho(S(x^{**}), S(x_0)) \geq \epsilon_t$, then go to PRC2.1.

PRC2.2. Set

$$\theta_t^{(i)} = \theta^{**} \text{ and } W_t^{(i)} = \begin{cases} \pi(\theta_t^{(i)}) / \mu_1(\theta_t^{(i)}) & \text{if } t = 1 \\ \dfrac{\pi(\theta_t^{(i)}) L_{t-1}(\theta^*|\theta_t^{(i)})}{\pi(\theta^*) K_t(\theta_t^{(i)}|\theta^*)} & \text{if } t > 1 \end{cases},$$

where $\pi(\theta)$ denotes the prior distribution for θ, and L_{t-1} is a backward transition kernel.

If $i < N$, increment $i = i + 1$ and go to PRC2.1.

PRC3. Normalize the weights so that $\Sigma_{i=1}^N W_t^{(i)} = 1$.
If $ESS = [\Sigma_{i=1}^N (W_t^{(i)})^2]^{-1} < E$ then resample with replacement, the particles $\{\theta_t^{(i)}\}$ with weights $\{W_t^{(i)}\}$ to obtain a new population $\{\theta_t^{(i)}\}$, and set weights $\{W_t^{(i)} = 1/N\}$.

PRC4. If $t < T$, increment $t = t + 1$ and go to PRC2.

Samples $\{\theta_T^{(i)}\}$ are weighted samples from the posterior distribution $f(\theta|\rho(S(x), S(x_0)) \leq \epsilon)$. The validity of this algorithm is derived by construction from the validity of the combination of general SMC methods and the PRC process (see *Appendix*). Algorithm ABC-PRC corresponds to algorithm ABC-REJ for the special case when $T = 1$ and $\mu_1(\theta) = \pi(\theta)$.

For the remainder of this article, we consider $K_t(\theta_t|\theta_{t-1}) = L_{t-1}(\theta_{t-1}|\theta_t)$ as a Gaussian kernel with common variance (following ref. 19), which we have found to perform adequately. For discussions on closer to optimal choices of L_{t-1}, see ref. 17, and for applications of SMC and more technical proofs of the SMC algorithm's validity, see refs. 16, 17, and 20–24. Finally, we note that if $K_t(\theta_t|\theta_{t-1}) = L_{t-1}(\theta_{t-1}|\theta_t)$, $\mu_1(\theta) = \pi(\theta)$ and the prior $\pi(\theta) \propto 1$ over Θ, then all weights are equal throughout the sampling process and may safely be ignored [in addition to ignoring all population (PRC3) resampling steps].

Results

Toy Example (Revisited). We now implement algorithm ABC-PRC in the mixture of normals posterior considered earlier. We generate a sample of $N = 1{,}000$ particles by considering a sequence of three distributions f_1, f_2, and f_3 defined by Eq. **1** with $\epsilon_1 = 2$, $\epsilon_2 = 0.5$ and $\epsilon_3 = 0.025$, and with prior distribution $\pi(\theta) \sim U(-10, 10)$. We specify $\mu_1(\theta) = \pi(\theta)$ and $K_t(\theta_t|\theta_{t-1}) = L_{t-1}(\theta_{t-1}|\theta_t)$ as a Gaussian random walk so that all weights are equal.

The initial (μ_1) population and histograms of f_1 to f_3 are illustrated in Fig. 2. The movement in distribution towards the target distribution is a clear progression, with the final sample adhering remarkably well to the target distribution, especially in the tails where the ABC-MCMC algorithm performed particularly poorly (Fig. 1).

ABC algorithms may be intuitively compared through the number of likelihood "evaluations," that is, the number of data-generation steps. Table 1 enumerates the mean number of data-generation steps required to move a particle between two successive populations. As the tolerance reduces with each successive distribution, f_t, the number of data-generation steps we expect increases. This effect is partially offset by the degree of similarity between population distribution f_t and its induced sampling distribution μ_t. The total number of data-simulation steps in the ABC-PRC algorithm was 75,895. This is more than the illustrated 10,000 for the ABC-MCMC algorithm (Fig. 1), but this latter simulation requires a substantially longer run before we can be satisfied that a representative sample has been drawn. Accordingly, the ABC-PRC algorithm is far more efficient for this case.

In contrast, using the ABC-REJ algorithm results in utilizing >400 data-simulation steps per final particle (Table 1). Here, there is a clear advantage in adopting a series of intermediary distribu-

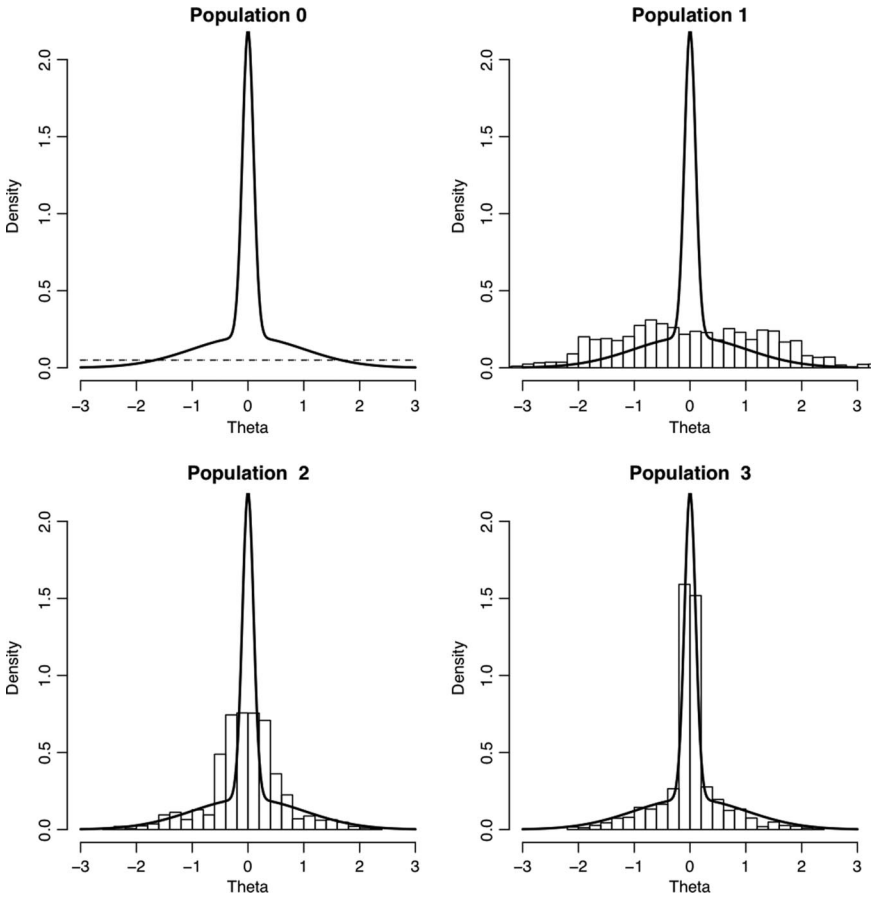

Fig. 2. (*Upper Left* to *Lower Right*) Particle distributions μ_1, f_1, f_2, and f_3 defined with $\epsilon_1 = 2$, $\epsilon_2 = 0.5$, and $\epsilon_3 = 0.025$ using ABC-PRC algorithm. Dashed line denotes $\pi(\theta)$. The mixture of normals target distribution is superimposed.

tions between μ_1 and the target distribution. Finally, as an indication of the maximum possible efficiency of ABC samplers for this example, performing rejection sampling with sampling distribution equal to the posterior distribution requires >21 data-generation steps per final particle. Note that each particle must still satisfy steps REJ2 and REJ3, so we do not automatically accept every particle proposed. This gives algorithm ABC-PRC 28% of maximum possible efficiency in this case, compared with only 5% for rejection sampling.

Analysis of Tuberculosis Transmission Rates. We now reimplement an analysis of tuberculosis transmission rates originally investigated using algorithm ABC-MCMC (7). The aim of this study was to estimate three compound parameters of biological interest, namely, the reproductive value, the doubling time, and the net transmission rate. The data used come from a study of tuberculosis isolates collected in San Francisco during the early 1990s by Small *et al.* (25). These consist of 473 isolates genotypes using the IS*6110* marker; the resulting DNA fingerprints can be grouped into 326 distinct genotypes as follows

$$30^1\ 23^1\ 15^1\ 10^1\ 8^1\ 5^2\ 4^4\ 3^{13}\ 2^{20}\ 1^{282},$$

where n^k indicates there were k clusters of size n. The ABC-MCMC method was used in conjunction with a stochastic model of transmission, which is an extension of a birth and death process to include mutation of the marker. Simulating samples from this model allows comparisons with the actual data through two summary statistics: g, the number of distinct genotypes in the sample, and H, the gene diversity. An informative prior was used for the mutation rate taken from published estimates of the rate for IS*6110*. Further details can be found in ref. 7.

Table 1. Mean number of data-generation steps per final particle for each population, based on 1,000 particles, under algorithms ABC-PRC and ABC-REJ

			ABC-REJ	
t	ϵ_t	ABC-PRC	Prior	Posterior
1	2.000	4.907	–	–
2	0.500	4.899	–	–
3	0.025	66.089	400.806	21.338
	Total	75.895	400.806	21.338

Final two columns use $U(-10, 10)$ prior and known posterior mixture of normals as sampling distributions.

Tanaka *et al.* (7) previously implemented the ABC-MCMC algorithm with tolerance $\epsilon = 0.0025$. Three Markov chains with an average acceptance rate of ≈0.3% were thinned and combined to form the final sample, utilizing >2.5 million data-generation steps. (Actually, more were used, as one chain became "stuck" in a distributional tail for most of its length, as illustrated in Fig. 1, and had to be rerun.)

We illustrate algorithm ABC-PRC with a sequence of 10 distributions, defined by $\epsilon_1 = 1$ and for $t = 2, \ldots, 9$, $\epsilon_t = \frac{1}{2}(3\epsilon_{t-1} - \epsilon_{10})$ is taken to be halfway between the previous tolerance and the target of $\epsilon_{10} = 0.0025$. Ten distributions were selected so that successive distributions were reasonably similar. We adopt $K_t(\theta_t|\theta_{t-1}) = L_{t-1}(\theta_{t-1}|\theta_t)$ as the ABC-MCMC Gaussian random walk proposal

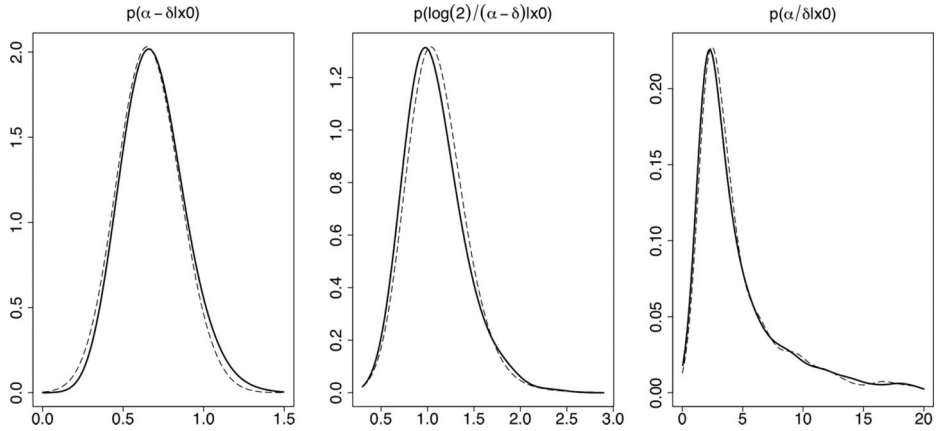

Fig. 3. (*Left* to *Right*) Posterior distribution of $f(\alpha - \delta|x_0)$ (net transmission rate) $f(\log(2)/(\alpha - \delta)|x_0)$ (doubling time) and $f(\alpha/\delta|x_0)$ (reproductive value) for both ABC-MCMC (solid) and ABC-PRC (dash) samplers.

of Tanaka *et al.* (7) with a slightly larger step for the mutation parameter.

Based on a population size of $N = 1{,}000$, Fig. 3 illustrates smoothed posterior distributions of the quantities of interest: (Fig. 3 *Left*) the net transmission rate ($\alpha - \delta$) is the rate of increase of the number of infectious cases in the population; the doubling time [$\log(2)/(\alpha - \delta)$] is the required duration for the number of infectious cases in the population to double; (Fig. 3 *Right*) the reproductive value (α/δ) is the expected number of new infectious cases produced by a single infectious case while the primary case is still infectious. As is evident, the results of the ABC-MCMC and ABC-PRC algorithms are essentially indistinguishable.

Relative algorithm efficiencies can again be measured by the mean number of data-generation steps per final particle. Table 2 lists the number of data-generation instances in ABC-PRC and ABC-REJ algorithms. For algorithm ABC-REJ, this amounts to a mean of 7,206.3 data-generation steps per particle. In contrast, algorithm ABC-PRC yields a mean of 1,421.3 data-generation steps per particle, >5 times more efficient.

Comparisons to the original ABC-MCMC analysis of Tanaka *et al.* (7) can also be made in terms of the number of data-generation steps required to generate one uncorrelated particle. Here, the Markov nature of the sampler and the very low acceptance rates induce a strongly dependent sequence. Thinning this sequence so that there were no significant (marginal) autocorrelations above lag 10 resulted in using 8,834 data-generations steps per realization. Repeating this so that there were no significant autocorrelations at any lag yielded 67 uncorrelated particles, corresponding to ≈27,313 data-generation steps per final realization. By this measure, algorithm ABC-PRC is ≈20 times more efficient than the MCMC implementation.

Table 2. Mean number of data-generation steps per final particle for each population, based on 1,000 particles, under algorithms ABC-PRC and ABC-REJ

t	ϵ	ABC-PRC	ABC-REJ
1	1.000	2.595	
2	0.5013	8.284	
3	0.2519	8.341	
4	0.1272	7.432	
5	0.0648	10.031	
6	0.0337	17.056	
7	0.0181	34.178	
8	0.0102	72.704	
9	0.0064	171.656	
10	0.0025	1,089.006	7,206.333
Total		1,421.283	7,206.333

Discussion

Likelihood-free samplers for Bayesian computation are growing in importance, particularly in population genetics and epidemiology, so it is crucial that efficient and accessible algorithms are made available to the practitioner. Existing MCMC algorithms exhibit an inherent inefficiency in their construction, whereas rejection methods are wasteful when sampling and target distributions are mismatched. Through an SMC approach, we may circumvent these problems and generate improved sampling, particularly in distributional tails, while achieving large computational savings.

Evaluations of certain user-specified aspects of algorithm ABC-PRC have not been presented, although these have been studied for SMC algorithms in general, and the necessity of their specification could be considered a disadvantage of the SMC approach. The incorporation of measures other than effective sample size to determine the optimal resampling time is given by Chen *et al.* (26) and forward and backward kernel choice by Del Moral *et al.* (17). Jasra *et al.* (27) give a study of various tolerance schedules and the number of distributions, $f_1, \ldots, f_T$. It seems credible that the tolerance schedule and distribution number could be determined dynamically based on one-step-ahead estimates of distributional change, $f_{t-1} \to f_t$, and the required computation (number of data-generation steps). This could be considered one method of selecting the final tolerance ϵ_T.

Appendix

We briefly justify the use of partial rejection control in deriving algorithm ABC-PRC. Following ref. 17, a generic sequential Monte Carlo algorithm is implemented as follows

SMC Algorithm

SMC1. Identify the sequence of distributions $f_1, \ldots, f_T$, where f_T corresponds to the target distribution, and initial sampling distribution μ_1.
Set population indicator $t = 1$.
SMC2. Set particle indicator $i = 1$.
SMC2.1. If $t = 1$, sample $\theta_t^{(i)} \sim \mu_1(\theta)$ independently from μ_1.
If $t > 1$, perturb each particle to $\theta_t^{(i)} \sim K_t(\theta|\theta_{t-1}^{(i)})$ according to a Markov transition kernel K_t.
SMC2.2. Evaluate weights $W_t^{(i)}$ for each particle according to

$$W_t^{(i)} \propto \begin{cases} f_t(\theta_t^{(i)})/\mu_1(\theta_t^{(i)}) & \text{if } t = 1 \\ W_{t-1}^{(i)} \dfrac{f_t(\theta_t^{(i)}) L_{t-1}(\theta_{t-1}^{(i)}|\theta_t^{(i)})}{f_{t-1}(\theta_{t-1}^{(i)}) K_t(\theta_t^{(i)}|\theta_{t-1}^{(i)})} & \text{if } t > 1, \end{cases}$$

where f_t denotes the intermediate distribution at step t and L_{t-1} is a backward transition kernel.
If $i < N$, increment $i = i + 1$ and go to SMC2.1.

SMC3. Normalize the weights so that $\Sigma_{i=1}^{N} W_t^{(i)} = 1$.
If $ESS = |\Sigma_{i=1}^{N} (W_t^{(i)})^2]^{-1} < E$, then resample with replacement, the particles $\{\theta_t^{(i)}\}$ with weights $\{W_t^{(i)}\}$ to obtain a new population $\{\theta_t^{(i)}\}$, and set weights $\{W_t^{(i)} = 1/N\}$.

SMC4. If $t < T$, increment $t = t + 1$ and go to SMC2.

Algorithm SMC can be justified intuitively by considering the final weight of particle $\theta_T^{(i)}$, assuming no weight normalization for clarity

$$W_T^{(i)} = \frac{f_1(\theta_1^{(i)})}{\mu_1(\theta_1^{(i)})} \prod_{t=2}^{T} \frac{f_t(\theta_t^{(i)}) L_{t-1}(\theta_{t-1}^{(i)}|\theta_t^{(i)})}{f_{t-1}(\theta_{t-1}^{(i)}) K_t(\theta_t^{(i)}|\theta_{t-1}^{(i)})}$$

$$= \frac{f_T(\theta_T^{(i)})}{\mu_1(\theta_1^{(i)})} \prod_{t=2}^{T} \frac{L_{t-1}(\theta_{t-1}^{(i)}|\theta_t^{(i)})}{K_t(\theta_t^{(i)}|\theta_{t-1}^{(i)})}.$$

The ratio $f_T(\theta_T^{(i)})/\mu_1(\theta_1^{(i)})$ is immediately identifiable as the standard importance sampling weight with μ_1 as the sampling distribution. The product of kernel ratios term evaluates the ratio of probabilities of moving from $\theta_T^{(i)} \to \theta_1^{(i)}$ (numerator) and from $\theta_1^{(i)} \to \theta_T^{(i)}$ (denominator).

Recognizing that many particles of small weight will have minimal impact on the final population, f_T, (e.g., they may be lost in the resampling step SMC3), partial rejection control aims to remove them at an earlier stage according to the following scheme (see chapters 2 and 3 of ref. 16).

Given a particle population $\{\theta_t^{(i)}\}$ with weights $\{W_t^{(i)}\}$, a small threshold, c, is specified such that all particles with weights greater than c remain unchanged. For those particles with weights smaller than c, there is a chance [with probability $\min(1, W_t^{(i)}/c)$] that these particles also remain unchanged, otherwise they are replaced by a particle from the previous population $\{\theta_{t-1}^{(i)}\}$ chosen according to weights $\{W_{t-1}^{(i)}\}$. This particle is then propagated from distribution f_{t-1} to f_t (via K_t) as before, where its weight is then compared to the threshold, c, once more. This process is repeated until all particles have passed the threshold. PRC is performed within SMC algorithms before the population resampling step (SMC3). See ref. 16 for a justification of this approach.

For a particle population $\{\theta_t^{(i)}\}$ with weights $\{W_t^{(i)}\}$ and $t > 1$, this process is given in algorithmic form by

PRC Algorithm

A1. Set threshold value $c > 0$ and particle indicator $i = 1$.

A2. With probability $\min\{1, W_t^{(i)}/c\}$, set weight $W_t^{(i)} = \max\{W_t^{(i)}, c\}$ and go to A4.
Otherwise, go to A3.

A3. Sample a new particle, θ^*, from $\{\theta_{t-1}^{(i)}\}$ with probability proportional to $\{W_{t-1}^{(i)}\}$.
Perturb the particle to $\theta_t^{(i)} \sim K_t(\theta|\theta^*)$ according to a Markov transition kernel K_t.
Set

$$W_t^{(i)} = \bar{W}_{t-1} \frac{f_t(\theta_t^{(i)}) L_{t-1}(\theta^*|\theta_t^{(i)})}{f_{t-1}(\theta^*) K_t(\theta_t^{(i)}|\theta^*)},$$

where $\bar{W}_{t-1} = 1/N \; \Sigma_{j=1}^{N} W_{t-1}^{(j)}$ and L_{t-1} is a backward transition kernel.
Go to A2.

A4. If $i < N$, increment $i = i + 1$ and go to A2.

A5. Normalize weights according to $\Sigma_{i=1}^{N} W_t^{(i)} = 1$.

PRC may benefit the SMC algorithm in the ABC setting as follows: The minimum computational specification for the sequence $\{B_t\}$ in the posterior (Eq. **1**) is $B_t = 1$ for all t. In this setting, large numbers of particles will have identically zero weight, as $\rho(S(x), S(x_0)) > \epsilon_t$ occurs with high probability. Suppose we then implement the PRC algorithm for some $c > 0$ such that only identically zero weights are smaller than c. This will remove those particles for which $\rho(S(x), S(x_0)) > \epsilon_t$ and replace them with particles for which $\rho(S(x), S(x_0)) \leq \epsilon_t$, which then belong to f_t.

This process is equivalent to deriving the *entire* population $\{\theta_t^{(i)}\}$ one particle at a time, by taking random draws from the previous population, perturbing the particle, and accepting the particle if $\rho(S(x), S(x_0)) \leq \epsilon_t$. That is, incorporating the data-generation process, we can replace step SMC2 in algorithm SMC above with step PRC2 in algorithm ABC-PRC. Accordingly, we are able to maximize algorithm efficiency in that we obtain a new particle on each occasion for which $\rho(S(x), S(x_0)) \leq \epsilon_t$, rather than wasting extra data-generation steps in overevaluating likelihood values.

We thank two anonymous referees whose suggestions have strongly improved the final form of this article. This work was supported by the Australian Research Council through the Discovery Grant scheme (DP0664970 and DP0556732) and by the Faculty of Science, University of New South Wales.

1. Bortot P, Coles SG, Sisson SA (2007) *J Am Stat Assoc*, in press.
2. Marjoram P, Molitor J, Plagnol V, Tavaré S (2003) *Proc Natl Acad Sci USA* 100:15324–15328.
3. Beaumont MA, Zhang W, Balding DJ (2002) *Genetics* 162:2025–2035.
4. Pritchard JK, Seielstad MT, Perez-Lezaun A, Feldman MW (1999) *Mol Biol Evol* 16:1791–1798.
5. Marjoram P, Tavaré S (2006) *Nat Rev Genet* 7:759–770.
6. Butler A, Glasbey C, Allcroft A, Wanless S (2006) *A Latent Gaussian Model for Compositional Data with Structural Zeroes* (Biomathematics and Statistics Scotland, Edinburgh), technical report.
7. Tanaka MM, Francis AR, Luciani F, Sisson SA (2006) *Genetics* 173:1511–1520.
8. Leman SC, Chen Y, Stajich JE, Noor MAF, Uyenoyama MK (2006) *Genetics* 171:1419–1436.
9. Thornton K, Andolfatto P (2006) *Genetics* 172:1607–1619.
10. Robert C, Casella G (2004) *Monte Carlo Statistical Methods* (Springer, New York), 2nd Ed.
11. Gilks WR, Richardson S, Spiegelhalter DJ, eds (1995) *Markov Chain Monte Carlo in Practice* (Chapman and Hall, London).
12. Tavaré S, Balding DJ, Griffiths RC, Donnelly P (1997) *Genetics* 145:505–518.
13. Cowles MK, Carlin BP (1996) *J Am Stat Assoc* 91:883–904.
14. Gelman A, Meng XL (1998) *Stat Sci* 13:163–185.
15. Neal R (2001) *Stat Comput* 11:125–139.
16. Liu JS (2001) *Monte Carlo Strategies in Scientific Computing* (Springer, New York).
17. Del Moral P, Doucet A, Jasra A (2006) *J R Stat Soc B* 68:411–436.
18. Liu J, Chen R (1998) *J Am Stat Assoc* 93:1032–1044.
19. Givens GH, Raftery AE (1996) *J Am Stat Assoc* 91:132–141.
20. Doucet A, de Freitas N, Gordon N, eds (2001) *Sequential Monte Carlo Methods in Practice* (Springer, New York).
21. Del Moral P (2004) *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications* (Springer, New York).
22. Kunsch HR (2005) *Ann Stat* 33:1983–2021.
23. Chopin N (2004) *Ann Stat* 32:2385–2411.
24. Peters GW (2005) MA thesis (Univ of Cambridge, Cambridge, UK).
25. Small PM, Hopewell PC, Singh SP, Paz A, Parsonnet J, Ruston DC, Schecter GF, Daley CL, Schoolnik GK (1994) *N Engl J Med* 330:1703–1709.
26. Chen Y, Xie J, Liu JS (2005) *J R Stat Soc B* 67:199–217.
27. Jasra A, Stephens DA, Holmes CC (2006) *On Population-Based Simulation for Static Inference* (Univ of Cambridge, Cambridge, UK), technical report.

Corrections

STATISTICS

Correction for "Sequential Monte Carlo without likelihoods," by S. A. Sisson, Y. Fan, and Mark M. Tanaka, which appeared in issue 6, February 6, 2007, of *Proc Natl Acad Sci USA* (104:1760–1765; first published January 30, 2007; 10.1073/pnas.0607208104).

The authors note the following: It has been brought to our attention that the algorithm introduced in our paper (ABC-PRC) can produce a biased posterior sample, most noticeably through underestimation in distributional tails. This result occurs as the likelihood ratio in the sequential Monte Carlo incremental weights is approximated by using two unbiased Monte Carlo estimates. One way to avoid a biased posterior sample in the ABC-PRC algorithm is to directly evaluate the importance sampling distribution using a near-optimal backwards kernel. Hence the need for the Monte Carlo estimate in the denominator of the likelihood ratio is circumvented, and an unbiased sampler is obtained. As such, a corrected ABC-PRC algorithm would be:

ABC-PRC Algorithm (corrected):

PRC1 Initialize $\epsilon_1, \ldots, \epsilon_T$, and specify initial sampling distribution μ_1.
Set population indicator $t = 1$.

PRC2 Set particle indicator $i = 1$.

PRC2.1 If $t = 1$ sample $\theta^{**} \sim \mu_1(\theta)$ independently from μ_1.
If $t > 1$ sample θ^* from the previous population $\{\theta_{t-1}^{(i)}\}$ with weights $\{W_{t-1}^{(i)}\}$, and perturb the particle to $\theta^{**} \sim K_t(\theta \mid \theta^*)$ according to a transition kernel K_t.

Generate a data set $x^{**} \sim f(x \mid \theta^{**})$.
If $\rho(S(x^{**}), S(x_0)) \geq \epsilon_t$ then go to PRC2.1.

PRC2.2 Set

$$\theta_t^{(i)} = \theta^{**} \qquad \text{and} \qquad W_t^{(i)} = \begin{cases} \pi\left(\theta_t^{(i)}\right) / \mu_1\left(\theta_t^{(i)}\right) & \text{if} \quad t = 1 \\ \pi\left(\theta_t^{(i)}\right) / \sum_{j=1}^{N} W_{t-1}\left(\theta_{t-1}^{(j)}\right) K_t\left(\theta_t^{(i)} | \theta_{t-1}^{(j)}\right) & \text{if} \quad t > 1 \end{cases}$$

where $\pi(\theta)$ denotes the prior distribution for θ.
If $i < N$, increment $i = i + 1$ and go to PRC2.1.

PRC3 Normalize the weights so that $\sum_{i=1}^{N} W_t^{(i)} = 1$.
If $ESS = \left[\sum_{i=1}^{N} (W_t^{(i)})^2\right]^{-1} < E$ then resample with replacement, the particles $\{\theta_t^{(i)}\}$ with weights $\{W_t^{(i)}\}$ to obtain a new population $\{\theta_t^{(i)}\}$, and set weights $\{W_t^{(i)} = 1/N\}$.

PRC4 If $t < T$, increment $t = t + 1$ and go to PRC2.

ACKNOWLEDGMENT. The authors thank C. Robert and G. W. Peters for constructive discussion.

www.pnas.org/cgi/doi/10.1073/pnas.0908847106

COMMENTARY

Correction for "A curvy, stretchy future for electronics," by John A. Rogers and Yonggang Huang, which appeared in issue 27, July 7, 2009, of *Proc Natl Acad Sci USA* (106:10875–10876; first published June 30, 2009; 10.1073/pnas.0905723106).

The authors note that on page 10875, right column, the sentence beginning on line 22, "For example, ultrathin sheets of silicon are flexible, for the same reason that any material in thin film form is flexible: bending strains are **inversely** proportional to thickness" should instead appear as "For example, ultrathin sheets of silicon are flexible, for the same reason that any material in thin film form is flexible: bending strains are **directly** proportional to thickness."

www.pnas.org/cgi/doi/10.1073/pnas.0908993106

ENVIRONMENTAL SCIENCES, CHEMISTRY

Correction for "Chlorine activation indoors and outdoors via surface-mediated reactions of nitrogen oxides with hydrogen chloride," by Jonathan D. Raff, Bosiljka Njegic, Wayne L. Chang, Mark S. Gordon, Donald Dabdub, R. Benny Gerber, and Barbara J. Finlayson-Pitts, which appeared in issue 33, August 18, 2009, of *Proc Natl Acad Sci USA* (106:13647–13654; first published July 20, 2009; 10.1073/pnas.0904195106).

The authors note that on page 13648, Equation 5 appeared incorrectly. Further, in Equation 6 on page 13649, the formula for nitryl chloride should have read: $ClNO_2$. These errors do not affect the conclusions of the article. The corrected equations appear below.

$$NO^+NO_3^- + HCl \rightarrow ClNO + HNO_3 \qquad [5]$$

$$NO_2^+NO_3^- + HCl \rightarrow ClNO_2 + HNO_3 \qquad [6]$$

www.pnas.org/cgi/doi/10.1073/pnas.0909721106