Sarah Filippi*, Chris P. Barnes, Julien Cornebise and Michael P.H. Stumpf*

On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo

Abstract: Approximate Bayesian computation (ABC) has gained popularity over the past few years for the analysis of complex models arising in population genetics, epidemiology and system biology. Sequential Monte Carlo (SMC) approaches have become work-horses in ABC. Here we discuss how to construct the perturbation kernels that are required in ABC SMC approaches, in order to construct a sequence of distributions that start out from a suitably defined prior and converge towards the unknown posterior. We derive optimality criteria for different kernels, which are based on the Kullback-Leibler divergence between a distribution and the distribution of the perturbed particles. We will show that for many complicated posterior distributions, locally adapted kernels tend to show the best performance. We find that the added moderate cost of adapting kernel functions is easily regained in terms of the higher acceptance rate. We demonstrate the computational efficiency gains in a range of toy examples which illustrate some of the challenges faced in real-world applications of ABC, before turning to two demanding parameter inference problems in molecular biology, which highlight the huge increases in efficiency that can be gained from choice of optimal kernels. We conclude with a general discussion of the rational choice of perturbation kernels in ABC SMC settings.

Keywords: dynamical systems; Bayesian parameter inference; sequential Monte Carlo; adaptive kernels; Likelihood-free; Kullback-Leibler.

***Corresponding authors: Sarah Filippi and Michael P.H. Stumpf,** Imperial College London, London, UK,
e-mail: s.filippi@imperial.ac.uk
Chris P. Barnes: Imperial College London, London, UK
Julien Cornebise: University College London, London, UK

1 Introduction

Statistical practice and theory tend to reflect scientific fashions (Stigler, 1986). Today mathematical models in biological sciences are becoming increasingly complex. This, together with the deluge of data typically produced in genetics and genomics, poses severe challenges to statistical inference (Efron, 2010). In particular, in many areas of computational biology evaluation of the likelihood (Cox, 2006)

$$L(\theta) = f(x|\theta),$$

where x are realizations of the data, and θ is the (potentially vector-valued) parameter characterising the data-generating process, is often turning out to be impractical. Approximate Bayesian Computation (ABC) methods (Beaumont et al., 2002; Marin et al., 2011) were first conceived to allow (Bayesian) statistical inference in situations where the evaluation of the likelihood is too complicated or numerically too demanding (Pritchard et al., 1999; Tanaka et al., 2006; Lopes and Beaumont, 2010). Rather than evaluating the likelihood directly, ABC-based approaches use systematic comparisons between real and simulated data in order to arrive at approximations of the true (but unobtainable) posterior distribution,

$$p(\theta|x) \propto f(x|\theta)\,\pi(\theta),$$

where $\pi(\theta)$ denotes the prior distribution of θ.

Simulating from $f(x|\theta)$ is generally straightforward, even if obtaining a reliable numerical or functional representation of the model is not possible. We then compare the simulated data, y, with the real data, x, and accept only those simulations where some distance measure between the two, $\Delta(x, y)$, falls below a specified threshold, ε. If the data are too intricate or complicated it is common to replace a comparison of the real and simulated data by a comparison of suitable summary statistics. This results in an often appreciable reduction of the dimension, but is fraught with problems if the summary statistics are not sufficient. Given that

sufficiency (Cox, 2006) is a rare quality indeed (Lehmann and Casella, 1998), and probably not given for any real-world problem of scientific interest, this problem is now attracting a lot of attention (Didelot et al., 2011; Fearnhead and Prangle, 2012; Robert et al., 2011; Barnes et al., 2012). Here, however, we shall focus on the data directly; we thus seek to determine approximate posteriors of the form,

$$p(\theta|x) \approx p_{\varepsilon}(\theta|x) \propto \int f(y|\theta)\, \mathbb{1}(\Delta(x,y) \leq \varepsilon)\, \pi(\theta) dy,$$

where y is the data simulated from the model $f(\cdot|\theta)$ for a given parameter, θ, drawn from the appropriate prior distribution, and x denotes the observed data.

The simple ABC scheme outlined above suffers from the same shortcomings as other rejection samplers: most of the samples are drawn from regions of parameter space which cannot give rise to simulation outputs that resemble the data. Therefore a number of computational schemes have been proposed that make ABC inference more efficient. These loosely come in three flavours: regression-adjusted ABC (Tallmon et al., 2004; Fagundes et al., 2007; Blum and François, 2010), Markov chain Monte Carlo ABC schemes (Marjoram and Molitor, 2003; Ratmann et al., 2007), and ABC implementing some variant of sequential importance sampling (SIS) or sequential Monte Carlo (SMC) (Sisson et al., 2007; Beaumont et al., 2009; Toni et al., 2009; Del Moral et al., 2011). Of these flavours the first and the last forms have received the greatest attention, and it is an ABC scheme based on sequential importance sampling that we will focus on as it offers greater flexibility and applicability, and appears to be enjoying greater popularity in applications.

We focus on the implementation of Toni et al. (2009) and Beaumont et al. (2009) (called ABC SMC in the following), which like other related SIS and SMC methods works by constructing a sequence of intermediate distributions that start out from a suitably specified prior distribution and increasingly resemble the (unknown) approximate posterior distribution. These methods aim to sample sequentially from a sequence of distributions, which increasingly resemble the target posterior; they are constructed by estimating intermediate distributions $p_{\varepsilon_t}(\theta|x)$ for a decreasing sequence of $\{\varepsilon_t\}_{1\leq t\leq T}$. Each intermediate distribution is described by a weighted sample of parameter vectors. Successive distributions are constructed by sampling parameters from the previous population, perturbing them through some kernel function, $\tilde{\theta} \sim K_t(\cdot|\theta)$, generating simulated data, $y \sim f(\cdot|\tilde{\theta})$, and, upon acceptance, calculating the corresponding new weights.

While this sequential ABC approach is computationally much more efficient than simple ABC rejection schemes, the overall computational burden does not only depend on the complexity of the model and the amount of data at hand, but also on details of the chosen sequential scheme. In particular the ε-schedule, $\{\varepsilon_1,\ldots,\varepsilon_T\}$, and the choice of perturbation kernels, $K_t(\cdot|\cdot)$ exert considerable influence on the algorithmic complexity. As in many Monte Carlo settings (Gilks et al., 1996; Robert and Casella, 2004) problems tend to arise as the dimension of the parameter space increases, and balancing convergence with an exhaustive exploration of the parameter space becomes harder. In this paper we focus on perturbation kernel selection and its effect on the algorithm's efficiency.

The construction of suitable kernel functions has been a longstanding problem in importance sampling (Oh and Berger, 1993; Givens and Raftery, 1996), sequential importance sampling and population Monte Carlo (Douc et al., 2007; Cappé et al., 2008; Cornuet et al., 2012), as well as in sequential Monte Carlo for state space models (Pitt and Shephard, 1999; Van Der Merwe et al., 2001; Cornebise et al., 2008). However, it is still far from being solved especially in an ABC context where formal and informal understanding of kernel choice remain areas of pressing concern.

Especially for models that are computationally expensive to simulate, such as dynamical systems (Gutenkunst et al., 2007; Secrier et al., 2009; Erguler and Stumpf, 2011), the choice of the kernel will have huge influence on the efficiency with which the parameter space is explored and posterior estimates obtained. Here we will discuss a range of kernel functions, characterize their performance, and put forward some analytic results as to their optimality. In the next section we discuss the ABC scheme in some detail before describing criteria for optimally choosing the perturbation kernels and outlining different classes of perturbation kernels. We then examine the performance of these kernels in a range of illustrative problems and compare their algorithmic complexity. We will then show that for two models in molecular biology with complex posterior parameter distributions the choice of suitable kernels can vastly improve the computational cost of ABC SMC inferences.

2 The ABC SMC algorithm

The general scheme of ABC inference is as follows:
- sample a parameter vector θ (also called a *particle*) from the prior distribution $\pi(\theta)$,
- simulate a dataset y according to the generative model $f(y|\theta)$,
- compare the simulated dataset with the experimental data x: if $\Delta(x, y) \leq \varepsilon$, accept the particle.

This scheme is repeated until N particles are accepted; these form a sample from the posterior distribution

$$p_\varepsilon(\theta|x) \propto \int \mathbb{1}(\Delta(x, y) \leq \varepsilon) f(y|\theta)\, \pi(\theta) dy,$$

which is an approximation of the posterior distribution $p(\theta|x)$.

Over the past few years many improvements of these algorithms have been proposed. In particular, Marjoram and Molitor (2003) introduced a method based on Markov chain Monte Carlo, which consists in constructing a Markov chain whose stationary distribution is $p_\varepsilon(\theta|x)$. This algorithm is guaranteed to converge, however, it is very difficult to assess when the Markov chain reaches the stationary regime; furthermore the chain may get trapped in local extrema.

SIS and SMC samplers have then been introduced in the ABC framework by several authors (Sisson et al., 2007; Beaumont et al., 2009; Toni et al., 2009; Del Moral et al., 2011). These methods aim to sample sequentially from the distributions $p_{\varepsilon_t}(\theta|x)$ for a decreasing sequence of $\{\varepsilon_t\}_{1\leq t\leq T}$. The scheme of the algorithm is as follows: first, the ABC algorithm described above is used to construct a sample from $p_{\varepsilon_1}(\theta|x)$ with a sufficiently large value of ε_1 such that many particles are accepted. The ABC algorithm is then used again with a smaller ε_2 as a threshold; but instead of sampling parameters from the prior, they are sampled from the set of accepted particles at the previous stage and perturbed according to a suitable *perturbation kernel*. This way a sample from $p_{\varepsilon_2}(\theta|x)$ is built, and so on until our target posterior has been reached.

In this article, we focus on the implementation described in Algorithm 1, which in the following will be referred to this as the ABC SMC algorithm according to Toni et al. (2009). The ABC population Monte Carlo algorithm proposed by Beaumont et al. (2009) is similar to the ABC SMC algorithm, except that a specific perturbation kernel is used. It is, however, worth distinguishing between these algorithms and the one of Del Moral et al. (2011) and Drovandi and Pettitt (2011) based on the SMC sampler of Del Moral et al. (2006). When using SMC samplers, both a forward and a backward kernel need to be defined, which reduces the algorithmic complexity from $O(N^2)$ to $O(N)$ where N is the number of particles. However, in many applications of interest, the most computationally expensive part of an ABC algorithm is the simulation of the data, which although of complexity $O(N)$ dominates the $O(N^2)$ term for any practically feasible value of N (Beaumont et al., 2002). In this paper, we will not discuss kernel choice in the context of these approaches, although here, too, the choice of kernel will impact the numerical efficiency.

The behaviour of the algorithm depends on its settings: in particular the decreasing sequence of $\{\varepsilon_t\}_t$ and the perturbation kernels $\{K_t(\cdot|\cdot)\}_t$. The effect of the sequence of decreasing thresholds is easy to understand: if the difference between two successive tolerances ε_t and ε_{t+1} is small, the posterior distributions $p_{\varepsilon_t}(\theta|x)$ and $p_{\varepsilon_{t+1}}(\theta|x)$ are similar and a small number of simulations will be required to generate N draws from the next intermediate distribution, $p_{\varepsilon_{t+1}}(\theta|x)$, by sampling from the weighted population $\{\theta^{(i,t-1)}, \omega^{(i,t-1)}\}_{1\leq i\leq N}$. But a slowly decreasing sequence of thresholds $\{\varepsilon_t\}_{1\leq t\leq T}$ leads to a large number of iterations (large value of T) in order to obtain $\varepsilon_T = \varepsilon$. In practise, until recently, the sequence of tolerance thresholds were most often tuned by hand according to the model. An adaptive choice of the threshold schedule has been proposed by Del Moral et al. (2011) and Drovandi and Pettitt (2011). It consists of selecting the α-th quantile of the distances between the simulated data $\{y^{(i,t)}\}_{1\leq i\leq N}$ and the observed data, x. Selecting the threshold adaptively often significantly improves the efficiency of the ABC SMC algorithm. However, the efficiency strongly depends on the choice of α and, as it is argued in Silk et al. (2012), for some values of α the algorithm may not converge to the posterior distribution $p_\varepsilon(\cdot|x)$.

Similarly, the choice of the perturbation kernels $\{K_t(\cdot|\cdot)\}_{1\leq t\leq T}$ exerts considerable influence on the computational complexity of the algorithm. A local perturbation kernel hardly moves the particles and has the

Algorithm 1 ABC SMC algorithm.

input: a threshold ε
output: a weighted sample of particles from $p_{\varepsilon}(\theta|x)$
$t \leftarrow 0$
repeat
  $t \leftarrow t+1$
  determine the next threshold ε_t
  determine the parameters of the perturbation kernel $K_t(\cdot|\cdot)$
  $i \leftarrow 1$
  repeat
    if t=1 **then**
      sample $\tilde{\theta}$ from $\pi(\theta)$
    else
      sample θ from the previous population $\{\theta^{(i,t-1)}\}_{1\le i\le N}$ with weights $\{\omega^{(i,t-1)}\}_{1\le i\le N}$
      sample $\tilde{\theta}$ from $K_t(\cdot|\theta)$ and such that $\pi(\tilde{\theta})>0$
    end if
    sample y from $f(\cdot|\tilde{\theta})$
    if $\Delta(y,x)\le\varepsilon_t$ **then**
      $\theta^{(i,t)} \leftarrow \tilde{\theta}$
      $y^{(i,t)} \leftarrow y$
      $i \leftarrow i+1$
    end if
  until $i=N+1$
  calculate the weights: for all $1\le i\le N$
  if $t\neq 1$ **then**

$$\omega^{(i,t)} \leftarrow \frac{\pi(\theta^{(i,t)})}{\sum_{j=1}^{n}\omega^{(j,t-1)}K_t(\theta^{(i,t)}|\theta^{(j,t-1)})}$$

  else $\omega^{(i,1)} \leftarrow 1$
  end if
  normalise the weights
until $\varepsilon_t \le \varepsilon$
$T \leftarrow t$

advantage of producing new particles which are accepted with high probability if the successive values of ε are close enough; on the other hand, a widely spread out or permissive perturbation kernel enables the exploration of the parameter space more fully, but does so at the cost of achieving only a low acceptance rate.

3 Properties of optimal kernels

In sequential importance sampling, a perturbation kernel K_t should fulfill several requirements to be computationally efficient. In particular, the joint proposal distribution, corresponding to picking a particle at random and perturbing it to obtain a new particle, should "resemble" in some sense the target joint distribution, corresponding to independently sampling two particles. More precisely, the joint proposal distribution of two particles, where we first sample a particle $\theta^{(t-1)} \sim p_{\varepsilon_{t-1}}(\cdot|x)$ and then a perturbed particle $\theta^{(t)} \sim K_t(\cdot|\theta^{(t-1)})$, and accept the couple if and only if $\Delta(y,x)\le\varepsilon_t$ where $y \sim f(\cdot|\theta^{(t)})$, admits for density

$$q_{\varepsilon_{t-1},\varepsilon_t}(\theta^{(t-1)},\theta^{(t)}|x) = \frac{p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)\,K_t(\theta^{(t)}|\theta^{(t-1)})\int f(y|\theta^{(t)})\,\mathbb{1}\left(\Delta(x,y)\le\varepsilon_t\right)dy}{\alpha(K_t,\varepsilon_{t-1},\varepsilon_t,x)}. \tag{1}$$

The normalisation factor

$$\alpha(K_t,\varepsilon_{t-1},\varepsilon_t,x) = \iiint p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)K_t(\theta^{(t)}|\theta^{(t-1)})f(y|\theta^{(t)})\,\mathbb{1}\left(\Delta(x,y)\le\varepsilon_t\right)d\theta^{(t-1)}d\theta^{(t)}dy \tag{2}$$

is the *average acceptance probability*, that is, the proportion of proposed particles that are not rejected. This joint proposal distribution should "resemble" the target product distribution, that of sampling $\theta^{(t-1)}$ and $\theta^{(t)}$ independently from, respectively, $p_{\varepsilon_{t-1}}(\cdot|x)$ and $p_{\varepsilon_t}(\cdot|x)$, whose density is

$$q^{\star}_{\varepsilon_{t-1},\varepsilon_t}(\theta^{(t-1)},\theta^{(t)}|x)=p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)\,p_{\varepsilon_t}(\theta^{(t)}|x). \tag{3}$$

As argued by several authors, e.g. (Douc et al., 2007; Cappé et al., 2008; Cornebise et al., 2008; Beaumont et al., 2009), a mathematically convenient formal definition of this "resemblance" is the Kullback-Leibler (KL) divergence between the proposal distribution $q_{\varepsilon_{t-1},\varepsilon_t}(\theta^{(t-1)},\theta^{(t)})$ and the target distribution $q^{\star}_{\varepsilon_{t-1},\varepsilon_t}(\theta^{(t-1)},\theta^{(t)})$, i.e.

$$KL(q_{\varepsilon_{t-1},\varepsilon_t};q^{\star}_{\varepsilon_{t-1},\varepsilon_t})=\iint q^{\star}_{\varepsilon_{t-1},\varepsilon_t}\left(\theta^{(t-1)},\theta^{(t)}\right)\log\frac{q^{\star}_{\varepsilon_{t-1},\varepsilon_t}\left(\theta^{(t-1)},\theta^{(t)}\right)}{q_{\varepsilon_{t-1},\varepsilon_t}\left(\theta^{(t-1)},\theta^{(t)}\right)}d\theta^{(t-1)}d\theta^{(t)}. \tag{4}$$

In order to determine the variance of a component-wise Gaussian kernel, Beaumont et al. (2009) also minimise this quantity, albeit considering only the special case where $\varepsilon_{t-1}=\varepsilon_t=0$ for which the solution has a closed form. In particular, in this special case, $\alpha(K_t, 0, 0, x)=1$ for any K_t.

However, in addition to maximising $KL\left(q_{\varepsilon_{t-1},\varepsilon_t};q^{\star}_{\varepsilon_{t-1},\varepsilon_t}\right)$, an efficient ABC SMC proposal kernel K_t should have a high average acceptance rate $\alpha(K_t, \varepsilon_{t-1}, \varepsilon_t, x)$. When refining Beaumont et al. (2009) with $\varepsilon_t\neq\varepsilon_{t-1}$, we note that $KL(q_{\varepsilon_{t-1},\varepsilon_t};q^{\star}_{\varepsilon_{t-1},\varepsilon_t})$ can be separated into three terms:

$$KL(q_{\varepsilon_{t-1},\varepsilon_t};q^{\star}_{\varepsilon_{t-1},\varepsilon_t})=-Q(K_t,\varepsilon_{t-1},\varepsilon_t,x)+\log\alpha(K_t,\varepsilon_{t-1},\varepsilon_t,x)+C(\varepsilon_{t-1},\varepsilon_t,x), \tag{5}$$

where

$$Q(K_t,\varepsilon_{t-1},\varepsilon_t,x)=\iint p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)\,p_{\varepsilon_t}(\theta^{(t)}|x)\log K_t(\theta^{(t)}|\theta^{(t-1)})\,d\theta^{(t-1)}d\theta^{(t)} \tag{6}$$

can be maximised easily in some convenient cases (see Section 4), $\alpha(K_t, \varepsilon_{t-1}, \varepsilon_t, x)$ is the average acceptance probability already defined in (2), and $C(\varepsilon_{t-1}, \varepsilon_t, x)$ does not depend on the kernel K_t. Therefore the two following maximisation problems are equivalent:

$$\operatorname{argmax}_{K_t}Q(K_t,\varepsilon_{t-1},\varepsilon_t,x)=\operatorname{argmax}_{K_t}\left(-KL(q_{\varepsilon_{t-1},\varepsilon_t};q^{\star}_{\varepsilon_{t-1},\varepsilon_t})+\log\alpha(K_t,\varepsilon_{t-1},\varepsilon_t,x)\right). \tag{7}$$

As we will show in Section 4 this problem is easy to solve since the left-hand side often admits a closed-form solution. The most important remark is that the right-hand side is the solution of a *multi-objective optimisation problem*, solving a trade-off between jointly *minimising the Kullback-Leibler divergence* and *maximising the logarithm of the average acceptance probability*. Multi-objective optimisation using an additive combination of two distinct objective functions is common practice, see e.g. section 4.7.5 of Boyd and Vandenberghe (2004). Although the weights of such additive combination are here forced upon us, we note that the use of the logarithm of the acceptance probability strongly penalises very low probabilities, while making equally desirable moderate to large acceptance probabilities, a reasonable preference from the computational point of view.

The practical consequence of these theoretical arguments is to choose the proposal kernel

$$K_t=\operatorname{argmax}_{K_t}Q(K_t,\varepsilon_{t-1},\varepsilon_t,x)\,. \tag{8}$$

We have shown that this choice corresponds to a trade-off between two desirable properties of the kernel, namely the resemblance to the proposal distribution and the target in the sense of the KL divergence, and a high acceptance rate. Our criterion not only sheds new light on the justification of some existing, proven criteria, but, additionally, refines them. In the next section we will describe how to carry out this maximisation in practice from a set of particles for random walk kernels.

Remark 1 (Resampling and finite population size). *For the sake of clarity, we have slightly simplified the equations above by omitting the resampling step of ABC-SMC. In reality it is not possible to sample exactly* $\theta^{(t-1)}$ *from* $p_{\varepsilon_{t-1}}(\cdot|x)$; *rather, as described in Algorithm 1,* $\theta^{(t-1)}$ *is sampled with replacement from the previous population of particles* $\{\theta^{(i,t-1)}\}$, *eventually sampling from the weighted empirical distribution of the previous population, noted* $p^N_{\varepsilon_{t-1}}(\cdot|x)$. *Systematically replacing* $p_{\varepsilon_{t-1}}(\cdot|x)$ *with* $p^N_{\varepsilon_{t-1}}(\cdot|x)$ *throughout equations* (1) *to* (8) *leads to the exact optimality criterion that we used, properly defined for a finite population size N.*

Remark 2 (Theoretical requirements for convergence). *Convergence of importance sampling-based algorithms such as SMC samplers and ABC-SMC places some precise conditions on the kernel: i) having a larger support than the target distribution, to guarantee asymptotic unbiasedness of the empirical mean by a law of large numbers; and ii) vanishing slowly enough in the tails of the target, equivalent to ensuring finite variance of the importance weights, to guarantee asymptotic normality and finite variance of the estimator by a central limit theorem.*

Fully investigating whether the optimal kernels satisfy such conditions is outside the scope of this article. However, the Gaussian kernels studied in the following sections have unbounded support, therefore satisfying a law of large numbers. The direction of the asymmetric KLD that we consider in equation (4) facilitates (but does not guarantee) finite variance. Indeed, this direction is opposite to that of the KLD used in Variational Bayes, but similar to that used in expectation propagation algorithms (Barthelmé and Chopin, 2011). As explained by, e.g. (Bishop, 2006, p. 468), this direction favours proposals with a good coverage of the target, therefore heuristically tending to satisfy the requirements for asymptotic normality.

4 Optimal choice of random walk kernels

Perturbing a particle $\theta^{(j,t-1)}$ consists of generating a new particle according to a probability distribution parametrised by $\theta^{(j,t-1)}$ and often centred on $\theta^{(j,t-1)}$. In the ABC SMC algorithm, in addition to sampling from the kernel, we must be able to compute the transition density $K_t(\theta^{(i,t)}|\theta^{(j,t-1)})$ for any particles $\theta^{(i,t)}$ and $\theta^{(j,t-1)}$. Then, instead of choosing the perturbation kernel K_t that maximises equation (6) over all possible kernels, the space of possible kernels is often restricted to a parametric family from which it is easy to sample and perform optimization. Different probability models may be used, with the most common being the uniform and the Gaussian distributions (Sisson et al., 2007; Toni et al., 2009; Liepe et al., 2010). In the following, we outline different classes of Gaussian perturbation kernels and compare their efficiency in terms of acceptance rate and computational cost.

4.1 Component-wise perturbation kernel

In most cases the particle is moved component-wise: for each component $1 \le j \le d$ of the parameter vector $\theta=(\theta_1,\ldots\theta_d)$, θ_j is perturbed independently according to a Gaussian distribution with mean θ_j and variance σ_j^2. The parameters $\{\sigma_j\}_{1\le j\le d}$ may be fixed in advance, but more frequently (Beaumont et al., 2009; McKinley et al., 2009; Toni and Stumpf, 2009; Jasra et al., 2010; Barnes et al., 2011; Didelot et al., 2011) adaptively chosen kernel widths, $\{\sigma_j^{(t)}\}_{1\le j\le d}$, are used which depend on the previous population — the scale or variance is then indexed by the population index, t. Considering a kernel of the form

$$K_t(\theta^{(t)}|\theta^{(t-1)}) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_j^{(t)}} \exp\left\{-\frac{(\theta_j^{(t)}-\theta_j^{(t-1)})^2}{2\sigma_j^{(t)\,2}}\right\} \tag{9}$$

and Maximising $Q(K_t, 0, 0, x)$ — or more precisely $Q(K_t, \varepsilon_{t-1}, \varepsilon_{t-1}, x)$ — Beaumont et al. (2009) showed that the optimal value of $\sigma_j^{(t)}$ is twice the empirical variance of the j-th component of the parameter vector in the previous population. Maximising $Q(K_t, \varepsilon_{t-1}, \varepsilon_t, x)$, however, leads to a slightly different choice,

$$\sigma_j^{(t)} = \left(\iint p_{\varepsilon_{t-1}}(\cdot|x)\, p_{\varepsilon_t}(\cdot|x)\, (\theta_j^{(t)} - \theta_j^{(t-1)})^2\, d\theta_j^{(t)}\, d\theta_j^{(t-1)} \right)^{1/2}. \tag{10}$$

This quantity can be approximated from the set $\{\theta^{(i,t-1)}, \omega^{(i,t-1)}, y^{(i,t-1)}\}_{1 \le i \le N}$ as,

$$\sigma_j^{(t)} \approx \left(\sum_{i=1}^{N} \sum_{k=1}^{N_0} \omega^{(i,t-1)} \tilde{\omega}^{(k,t-1)} (\tilde{\theta}_j^{(k,t-1)} - \theta_j^{(i,t-1)})^2 \right)^{1/2}, \tag{11}$$

where

$$\left\{ \left(\tilde{\theta}^{(k,t-1)}, \tilde{\omega}^{(k,t-1)} \right) \right\}_{1 \le k \le N_0} = \left\{ \left(\theta^{(i,t-1)}, \frac{\omega^{(i,t-1)}}{\bar{\omega}} \right) \text{s.t.} \quad \Delta(x, y^{(i,t-1)}) \le \varepsilon_t, \quad 1 \le i \le N \right\} \tag{12}$$

and $\bar{\omega}$ is a normalising constant such that $\sum_{k=1}^{N_0} \tilde{\omega}^{(k,t-1)} = 1$.

Another commonly used component-wise kernel is the *uniform kernel*, which consists of perturbing the j-th component of particle θ to any value in the interval $\left[\theta_j - \sigma_j^{(t)}; \theta_j - \sigma_j^{(t)}\right]$ with density $1/2\sigma_j^{(t)}$. A natural choice is to set the parameter $\sigma_j^{(t)}$ to the scale of the previous population, that is

$$\sigma_j^{(t)} \approx \frac{1}{2} \left(\max_{1 \le k \le N} \{\theta_j^{(k,t-1)}\} - \min_{1 \le k \le N} \{\theta_j^{(k,t-1)}\} \right).$$

Note that the main difference between the uniform and the component-wise normal kernels concerns their support: bounded vs. unbounded.

4.2 Multivariate normal perturbation kernels

Consider a population of two-dimensional parameters whose components are highly correlated. The perturbation of a particle according to the uniform kernel consists in sampling a parameter uniformly in a rectangle whose sides are parallel to the axes (see Figure 1 left). Similarly, the density levels of the component-wise normal kernel are ellipsoids whose principal axes are parallel to the parameter axes (see Figure 1 centre). For highly correlated parameters the use of component-wise perturbation kernels in the ABC SMC framework can lead to a small acceptance rate because they can inadequately reflect the structure of the true posterior.

Instead of using a component-wise kernel it may thus be more efficient to take into account the correlation between the different elements of the parameter vectors, in effect perturbing the particles according to a multivariate normal distribution with a covariance matrix $\Sigma^{(t)}$, which depends on the covariance of the previous population. Figure 1 (right) represents a *multivariate normal perturbation kernel* for a matrix $\Sigma^{(t)}$ proportional to the covariance of the previous population. We observe that fewer particle proposals are likely to be rejected with this perturbation kernel compared to the uniform or component-wise normal one.

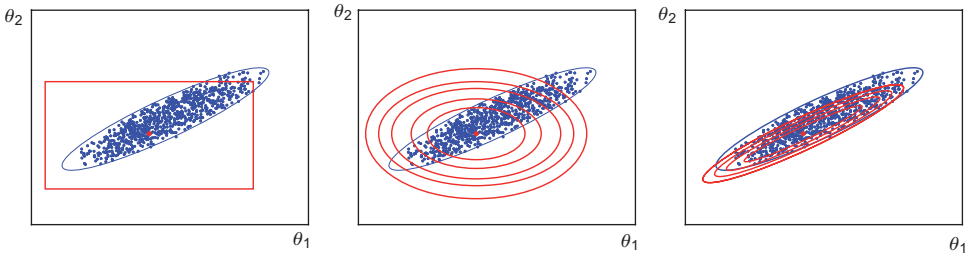

Figure 1 A population of particles and isodensity curves for a uniform kernel (left), a component-wise normal kernel (centre) and a multivariate normal perturbation kernel (right) around one particle (red dot).

The multivariate normal perturbation kernel relies on the covariance matrix $\Sigma^{(t)}$ which depends on the previous population. As before, it is possible to calculate the optimal covariance matrix $\Sigma^{(t)}$ using the Kullback-Leibler divergence minimisation approach [see also (Cappé et al., 2008)]. Maximising equation (6) for

$$K_t(\theta^{(t)}|\theta^{(t-1)}) = (2\pi)^{-d/2}\left(\det\Sigma^{(t)}\right)^{-1/2}\exp\left\{-\frac{1}{2}\left(\theta^{(t)}-\theta^{(t-1)}\right)^T\left(\Sigma^{(t)}\right)^{-1}\left(\theta^{(t)}-\theta^{(t-1)}\right)\right\}$$

with respect to $\Sigma^{(t)}$ leads to maximising the real-valued function

$$g(M) = \log\det(M) - \iint p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)\,p_{\varepsilon_t}(\theta^{(t)}|x)(\theta^{(t)}-\theta^{(t-1)})^T M(\theta^{(t)}-\theta^{(t-1)})\,d\theta^{(t)}d\theta^{(t-1)},$$

with respect to the symmetric $d\times d$ matrix M, and defining $\Sigma^{(t)}=M^{-1}$. We denote by v^T the transpose of vector v. We then obtain that the covariance matrix $\Sigma^{(t)}$ of the optimal kernel in the multivariate Gaussian family is

$$\Sigma^{(t)} = \iint p_{\varepsilon_{t-1}}(\theta^{(t-1)}|x)\,p_{\varepsilon_t}(\theta^{(t)}|x)(\theta^{(t)}-\theta^{(t-1)})(\theta^{(t)}-\theta^{(t-1)})^T\,d\theta^{(t)}d\theta^{(t-1)}\ .$$

Proceeding in a similar fashion to the component-wise case, if we assume that $\varepsilon_t=\varepsilon_{t-1}$, then the optimal covariance matrix can be approximated by twice the empirical covariance matrix of the population at t–1. In the general case, however, an optimal choice of the covariance matrix $\Sigma^{(t)}$ for the multivariate normal perturbation kernel is approximated by

$$\Sigma^{(t)} \approx \sum_{i=1}^{N}\sum_{k=1}^{N_0}\omega^{(i,t-1)}\tilde{\omega}^{(k)}(\tilde{\theta}^{(k)}-\theta^{(i,t-1)})(\tilde{\theta}^{(k)}-\theta^{(i,t-1)})^T, \tag{13}$$

where $\{\tilde{\theta}^{(k)}\}_{1\le k\le N_0}$ and $\{\tilde{\omega}^{(k)}\}_{1\le k\le N_0}$ are defined by equation (12). In the following, if nothing else is specified, the *multivariate normal kernel* refers to the kernel with this choice of covariance matrix. This result generalises the work of Beaumont et al. (2009).

4.3 Local perturbation kernels

In many applications the parameters of the system are highly correlated but in a non-linear way. The multivariate normal and the component-wise normal kernels discussed above may then behave similarly (see for example the toy model described in Section 5.3). Indeed, the covariance matrix based on all the previous particles yields only limited information about the local correlation among the individual components of the parameter vectors. In such cases it is interesting to consider the use of a local covariance matrix which now may differ between particles. In the following we discuss three local perturbation kernels for which each particle θ is perturbed according to a multivariate normal kernel whose covariance matrix $\Sigma_\theta^{(t)}$ is a function of θ.

4.3.1 The multivariate normal kernel with *M* nearest neighbours

The *multivariate normal kernel with M neighbours* follows this principle: for each particle $\theta\in\{\theta^{(l,t-1)}, 1\le l\le N\}$, the M-nearest neighbours of θ are selected, and the perturbed particle is sampled according to a multivariate normal distribution of mean θ and of covariance the empirical covariance $\Sigma_{\theta,M}^{(t)}$ based on the M nearest neighbours of θ.

The main drawback of this perturbation kernel is that the parameter M typically needs to be fixed in advance before any of the intricacies of the posterior are known. Using too small a value may lead to a lack of exploration of parameter space, while too large a value of M would offer little or no advantage compared to the standard multivariate normal kernel. Ideally, a mixture of multivariate normal kernels with different values of M could be used; however, in practice, this solution is computationally too expensive.

4.3.2 The multivariate normal kernel with optimal local covariance matrix

The theoretical calculation of the optimal covariance matrix above (see Section 4.2) may be adapted to identify an optimal local covariance matrix. Let us therefore consider a particle $\theta^{(t-1)}$ that has been sampled from the previous population $\{\theta^{(k,t-1)}\}_{1\le k\le N}$. To determine the covariance matrix $\Sigma^{(t)}_{\theta^{(t-1)}}$ of a multivariate normal perturbation kernel centred in $\theta^{(t-1)}$ we derive a criterion $Q(K_t, \varepsilon_t, x)$ similar to the criterion defined in equation (6) with the difference that the particle $\theta^{(t-1)}$ is now fixed,

$$Q(K_t,\varepsilon_t,x)=\int p_{\varepsilon_t}(\theta^{(t)}|x)\log K_t(\theta^{(t)}|\theta^{(t-1)})d\theta^{(t)}, \tag{14}$$

which is equal to

$$\int p_{\varepsilon_t}(\theta^{(t)}|x)\log\det\left(\Sigma^{(t)}_{\theta^{(t-1)}}\right)^{-1}-\frac{d}{2}\log(2\pi)-\frac{1}{2}(\theta^{(t)}-\theta^{(t-1)})^T\left(\Sigma^{(t)}_{\theta^{(t-1)}}\right)^{-1}(\theta^{(t)}-\theta^{(t-1)})d\theta^{(t)}, \tag{15}$$

where $K_t(\theta^{(t)}|\theta^{(t-1)})$ is a multivariate normal kernel centred on $\theta^{(t-1)}$ with covariance matrix $\Sigma^{(t)}_{\theta^{(t-1)}}$. Maximising the equation above with respect to $\Sigma^{(t)}_{\theta^{(t-1)}}$ the optimal local covariance matrix is

$$\Sigma^{(t)}_{\theta^{(t-1)}}=\int p_{\varepsilon_t}(\theta^{(t)}|x)(\theta^{(t)}-\theta^{(t-1)})(\theta^{(t)}-\theta^{(t-1)})^T d\theta^{(t)},$$

which can be approximated from the set $\{\theta^{(i,t-1)}, \omega^{(i,t-1)}, y^{(i,t-1)}\}_{1\le i\le N}$ as

$$\Sigma^{(t)}_{\theta^{(t-1)}}\approx\sum_{k=1}^{N_0}\tilde{\omega}^{(k)}(\tilde{\theta}^{(k)}-\theta^{(t-1)})(\tilde{\theta}^{(k)}-\theta^{(t-1)})^T,$$

where $\{\tilde{\theta}^{(k)}\}_{1\le k\le N_0}$ and $\{\tilde{\omega}^{(k)}\}_{1\le k\le N_0}$ are defined by equation (12). To better understand the covariance matrix $\Sigma^{(t)}_{\theta^{(t-1)}}$ we remark, by a classical bias-variance decomposition, that it is equal to the covariance of the particles from the previous population corresponding to distances smaller than ε_t (i.e. the weighted particle denoted $\{\tilde{\theta}^{(k)},\tilde{\omega}^{(k)}\}_{1\le k\le N_0}$ through this article) plus a bias term related to the discrepancy between the mean of this population and the particle of interest $\theta^{(t-1)}$,

$$\Sigma^{(t)}_{\theta^{(t-1)}}\approx\sum_{k=1}^{N_0}\tilde{\omega}^{(k)}(\tilde{\theta}^{(k)}-m)(\tilde{\theta}^{(k)}-m)^T+(m-\theta^{(t-1)})(m-\theta^{(t-1)})^T,$$

where $m=\sum_{k=1}^{N_0}\tilde{\omega}^{(k)}\tilde{\theta}^{(k)}$.

The multivariate normal perturbation kernel with a covariance matrix as defined above will be referred to as the *multivariate normal kernel with OLCM* (where OLCM stands for optimal local covariance matrix). We now have a different covariance matrix $\Sigma^{(t)}_{\theta^{(j,t-1)}}$ for each particle $\theta^{(j,t-1)}$ of the previous population.

4.3.3 Perturbation kernel based on the Fisher information for models defined by ordinary or stochastic differential equations

Often in molecular biology the time evolution of species abundance is modelled by a system of ordinary differential equations (ODE), stochastic differential equations (SDE) or a chemical master equation (CME). For these types of models it is possible to use information from the generative model — via the Fisher information matrix (FIM) (Rao, 1945; MacKay, 2003; Cox, 2006) — to compute a local covariance matrix for the perturbation kernel. The FIM defined as

$$I(\theta)=-E_X\left[\frac{\partial^2}{\partial\theta^2}\log f(X|\theta)\right]$$

measures the amount of information that the observable random variable X carries about the parameter θ. As previously mentioned, the ABC algorithm is mainly used when the likelihood function $f(\cdot|\theta)$ is not known, and so the Fisher information matrix cannot be computed exactly. Nevertheless, Komorowski et al. (2011) have developed a method that approximates the FIM for deterministic and stochastic dynamical systems represented by ODEs and by SDEs (using the linear noise approximation). This evaluation of the FIM can be applied as part of the ABC SMC procedure and so the perturbation kernels can adapt appropriately. Despite the fact that the following kernels cannot be computed in the general case, we consider them here because of their potential use for inference in dynamical systems.

In the Laplace expansion the eigenvectors and eigenvalues of the inverse of the FIM $I(\theta)$ map out ellipsoidal levels of equal density around the parameter θ. This immediately suggests the use of the matrix $I^{-1}(\theta)$ as the covariance matrix for a multivariate normal perturbation kernel. The directions of the eigenvectors of $I^{-1}(\theta)$ and the relative size of their eigenvalues are indeed both relevant for perturbing the parameter θ efficiently (see Figure 2). However, the determinant of $I(\theta)$ is a measure of the amount of information available around θ and may vary exponentially with θ (see Figure 2 right); its value is very small for some parameters θ and this leads to a perturbation kernel with too large a covariance. On the other hand, if the determinant of $I(\theta)$ is large, additional information may be gained by moving only in the direct vicinity of θ, and a perturbation kernel based on the inverse of the FIM explores only the immediate neighbourhood of the parameter.

We therefore propose scaling the matrix $I^{-1}(\theta)$ such that its determinant varies in a more controlled manner. We consider two versions of the *multivariate normal perturbation kernel based on the FIM*: the first one consists of normalising the matrix such that its determinant is equal to the determinant of the empirical covariance matrix of the previous population,

$$\Sigma_{\theta}^{(t)} = \left(\frac{\det(\Sigma^{(t)})}{\det(I^{-1}(\theta))} \right)^{1/d} I^{-1}(\theta),$$

where $\Sigma^{(t)}$ is the matrix defined in equation (13); the second approach consists of normalising the matrix such that its determinant is equal to the determinant of the empirical covariance matrix, $\Sigma_{\theta,M}^{(t)}$, based on the M nearest neighbours of the particle defined according to

$$\Sigma_{\theta,M}^{(t)} = \left(\frac{\det(\Sigma_{\theta,M}^{(t)})}{\det(I^{-1}(\theta))} \right)^{1/d} I^{-1}(\theta).$$

The parameter M may, for instance, be equal to 20% of the previous population.

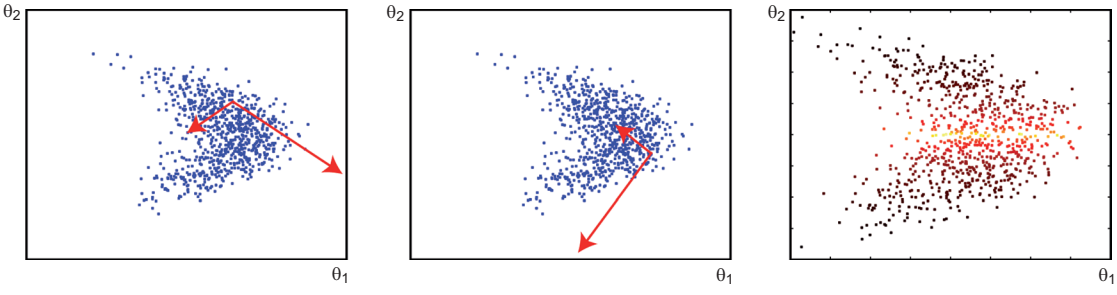

Figure 2 Local kernel based on the FIM $I(\theta)$. Left and centre: The eigenvectors of $I^{-1}(\theta)$ (red arrows) of size proportional to the eigenvalues for two different particles θ. Right: Logarithm of the determinant of $I^{-1}(\theta)$ for each particle θ. The maximum value of $\det(I^{-1}(\theta))$ over the population is equal to 85745 (yellow points) and the minimum one is equal to 0.14 (black points).

5 Numerical results

We first apply the ABC SMC algorithm with different kernels to three illustrative examples, which exhibit certain pathological features that highlight the differences between the perturbation kernels considered here. In order to analyse the impact of the kernel choice rather than any extraneous factors, we fix the threshold schedule $(\varepsilon_t)_t$ to (160, 120, 80, 60, 40, 30, 20, 15, 10, 8, 6, 4, 3, 2, 1) and use the same for every simulation. However, in practice we strongly advise using an adaptive choice of the ε threshold as well. All the simulations were done using the software *ABC-SysBio* (Liepe et al., 2010) version 1.03 using an Euclidian distance to compare simulated and observed data. Newer versions have implemented some of the adaptive kernels discussed here.

5.1 Ellipsoid shape

We begin with a toy example where the prior distribution of the two dimensional parameter is a uniform distribution on the square $[-50, 50]\times[-50, 50]$ and the likelihood function is given by

$$x \sim \mathcal{N}\left((\theta_1 - 2\theta_2)^2 + (\theta_2 - 4)^2, 1\right).$$

It is assumed that $x=0$ is observed. The posterior density is then

$$p(\theta|x) \propto \phi(0; (\theta_1 - 2\theta_2)^2 + (\theta_2 - 4)^2, 1)\mathbb{1}_{[-50,50]\times[-50,50]}(\theta)$$

where $\phi(x; \mu, \sigma^2)$ is the one dimensional normal density with mean μ and variance σ^2, and is represented in Figure 3A. The ABC SMC algorithm is used to estimate $p_\varepsilon(\theta|x)$ with $N=800$ particles. We compare six different perturbation kernels: (1) the uniform kernel (Section 4.1); (2) the component-wise normal kernel (Section 4.1); (3) the component-wise normal kernel proposed by Beaumont et al. (2009); (4) the multivariate normal kernel with the covariance matrix computed from the whole previous population (Section 4.2); (5) the multivariate normal kernel whose covariance matrix is computed according to the M nearest neighbours of each particle (Section 4.3.1), with $M=50$; and (6) the multivariate normal kernel with OLCM (Section 4.3.2).

We reiterate that the variance of the component-wise normal kernel proposed by Beaumont et al. (2009) is slightly different than the one we propose here: they suggest to use twice the empirical variance of the previous population as a variance whereas we take into account the new threshold ε_t, as described in equation (11).

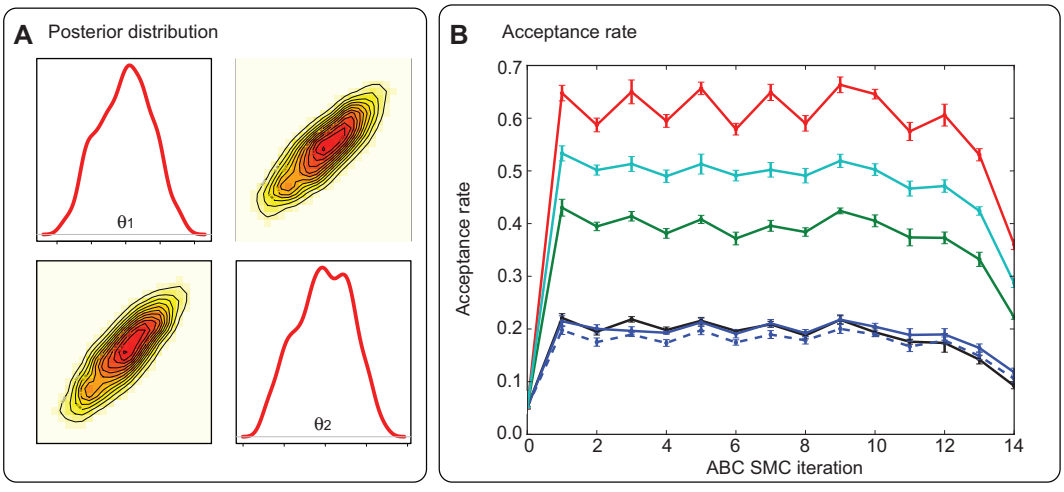

Figure 3 (A) Posterior distribution for an ellipsoid posterior. (B) Average of the acceptance rate over ten independent runs for six different kernels: the uniform kernel (green), the component-wise normal kernel (blue), the component-wise normal kernel proposed by Beaumont et al. (2009) (dashed blue), the multivariate normal kernel (black), the multivariate normal kernel with 50 neighbours (red), the multivariate normal kernel with OLCM (cyan).

Figure 3B shows that the acceptance rate differs significantly between kernels. The uniform kernel has an acceptance rate roughly equal to that of the component-wise normal kernel. Moreover, the two versions of the component-wise normal kernels have similar acceptance rates, with a slightly better performance for the one taking into account the difference between the successive threshold values. Given the shape of the posterior distribution, it is easy to understand that a multivariate normal kernel results in a larger acceptance rate than the other kernels. Since the two components of the parameters are strongly correlated, using an estimate of the covariance from the previous population instead of an estimation of only the component-wise variances makes a marked difference on the acceptance rates. Both the multivariate normal kernel based on the 50 nearest neighbours and the one based on the OLCM result in acceptance rates over two times higher than the component-wise kernels.

5.2 Ring shape

In the second toy example, the prior distribution of the two dimensional parameter is still a uniform distribution on the square [−50,50]×[−50,50] but the likelihood function is now given by

$$x \sim \mathcal{N}\left(\theta_1^2+\theta_2^2, 0.5\right).$$

Again we assume that x=0 is observed; the posterior density is then

$$p(\theta|x) \propto \phi(0;\theta_1^2+\theta_2^2,0.5)\,\mathbb{1}_{[-50,50]\times[-50,50]}(\theta).$$

As in the previous example we used the ABC SMC algorithm with N=800 particles and compare the same six perturbation kernels.

The posterior distribution, represented by Figure 4A, has a ring shape centred around 0. In this case, in contrast to the previous example, the multivariate normal perturbation kernel using an estimate of the covariance based on the previous population, as well as the OLCM version, have an acceptance rate similar to the component-wise normal perturbation kernel. In this example the correlation between the two parameters, θ_1 and θ_2, at the whole population level is weak. This kind of shape requires a more local perturbation kernel in order to obtain higher acceptance rates. This is the case for the perturbation kernel based on the covariance matrix computed from the 50 nearest neighbours.

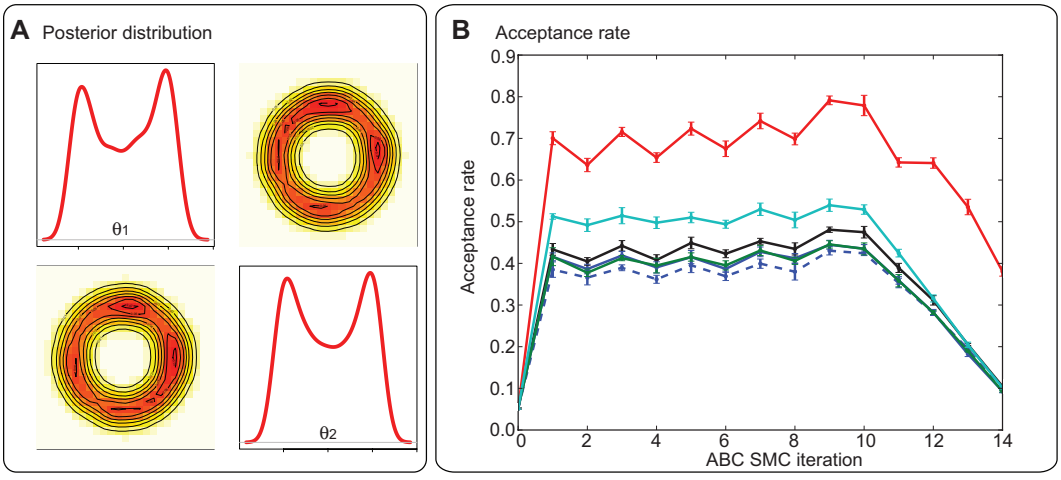

Figure 4 (A) Posterior distribution for a ring posterior. (B) Average of the acceptance rate over ten independent runs for six different kernels: the uniform kernel (green), the component-wise normal kernel (blue), the component-wise normal kernel proposed by Beaumont et al. (2009) (dashed blue), the multivariate normal kernel (black), the multivariate normal kernel with 50 neighbours (red), the multivariate normal kernel with OLCM (cyan).

5.3 Banana shape

The third example we consider is one of the canonical examples of a posterior distribution which poses a challenge to simple kernels: the so-called "banana-shape" distribution in two dimensions (Haario et al., 1999). The likelihood function is given by

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix} \theta_1 \\ \theta_1+\theta_2^2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}\right)$$

and we use a uniform prior distribution on the square $[-50,50]\times[-50,50]$. It is assumed that $x=(0,0)$ is observed. The posterior density is then

$$p(\theta|x) \propto \Phi\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} \theta_1 \\ \theta_1+\theta_2^2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}\right) \mathbb{1}_{[-50,50]\times[-50,50]}(\theta)$$

where $\Phi(x; \nu, \Sigma)$ is the multi-dimensional normal density of mean ν and covariance Σ. We use the same ABC SMC settings and again compare the 6 previous perturbation kernels, as well as two versions of the multivariate normal perturbation kernel where the covariance matrix is proportional to the inverse of the FIM. Here the FIM is exactly computable:

$$I(\theta) = \begin{pmatrix} 1.5 & \theta_2 \\ \theta_2 & 2\theta_2^2 \end{pmatrix}.$$

When $\theta_2 = 0$ we replace it by a very small value, 10^{-4}, such that $I(\theta)$ is no longer singular; safeguarding against singular FIMs is straightforward and unproblematic and a sensible precaution when running such algorithms without manual intervention.

The posterior distribution is represented in Figure 5A. As in the ring example, the multivariate normal perturbation kernel using the full estimated covariance of the whole previous population has an acceptance rate similar to the component-wise normal perturbation kernel with adaptive estimation of the variances (see Figure 5B). The multivariate normal kernel with OLCM obtains slightly better results. The two versions of the perturbation kernel based on the FIM have significantly different acceptance rates. The most efficient version in term of acceptance rate is the one using the M nearest neighbours, as might be expected. However this kernel, as in the case of the multivariate normal kernel based on the M nearest neighbours, can show undesirable dependence on the chosen value of M. Figure 5B represents the acceptance rate evolution for this last perturbation kernel with different values of M. The acceptance rate diminishes considerably as M increases and the kernel becomes less "locally aware".

6 Applications in molecular biology

6.1 The repressilator model

To analyse the differences between the efficiency of the perturbation kernel in biological applications we first focus on simulated datasets for the repressilator model, a popular model for gene regulatory systems (Elowitz and Leibler, 2000). It consists of three genes connected in a feedback loop, where each gene transcribes the repressor protein for the next gene in the loop (see Figure 6). This model also exemplifies the challenges that are frequently encountered in attempts to reverse engineer the structure and parameters of dynamical systems from data (Toni et al., 2009; Girolami and Calderhead, 2011).

The evolution of the concentration of the three proteins and mRNAs over time is described by a system of six ordinary differential equations (ODE) parametrised by a four dimensional parameter vector, $\theta=(\alpha_0, n, \beta, \alpha)$,

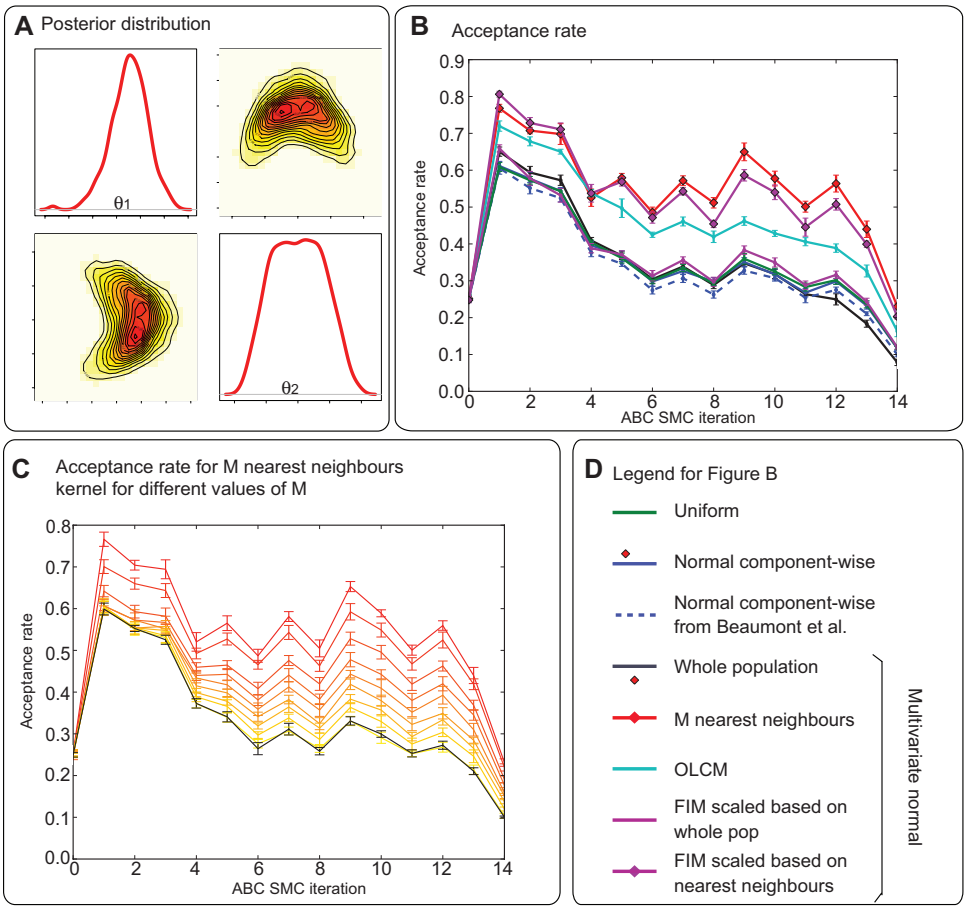

Figure 5 (A) Posterior distribution for a banana shape posterior. (B) Average of the acceptance rate over ten independent runs for eight different kernels. (C) Acceptance rate for multivariate normal kernels based on the M neighbours for $M \in \{50, 100, 200, 300, 400, 500, 600, 700,800\}$ (from red to yellow) and the multivariate kernel with an estimated covariance based on the whole population (black). (D) Legend for Figure B.

$$\frac{dm_1}{dt} = -m_1 + \frac{\alpha}{1+p_3^n} + \alpha_0$$
$$\frac{dp_1}{dt} = -\beta(p_1 - m_1)$$
$$\frac{dm_2}{dt} = -m_2 + \frac{\alpha}{1+p_1^n} + \alpha_0$$
$$\frac{dp_2}{dt} = -\beta(p_2 - m_2)$$
$$\frac{dm_3}{dt} = -m_3 + \frac{\alpha}{1+p_2^n} + \alpha_0$$
$$\frac{dp_3}{dt} = -\beta(p_3 - m_3)\,.$$

We denote by m_i and p_i the concentration of the mRNA and protein products of gene i respectively. The parameters in the model are the Hill coefficient n, repression strength α, basal expression rate α_0 and the ratio of the protein decay rate to the mRNA decay rate β. These are assumed to be the same for all three genes.

We assume that only the mRNA (m_1, m_2, m_3) measurements are available, and set the initial species concentrations of ($m1$, $p1$, $m2$, $p2$, $m3$, $p3$) to (0, 2, 0, 1, 0, 3); data are generated by simulating the model with

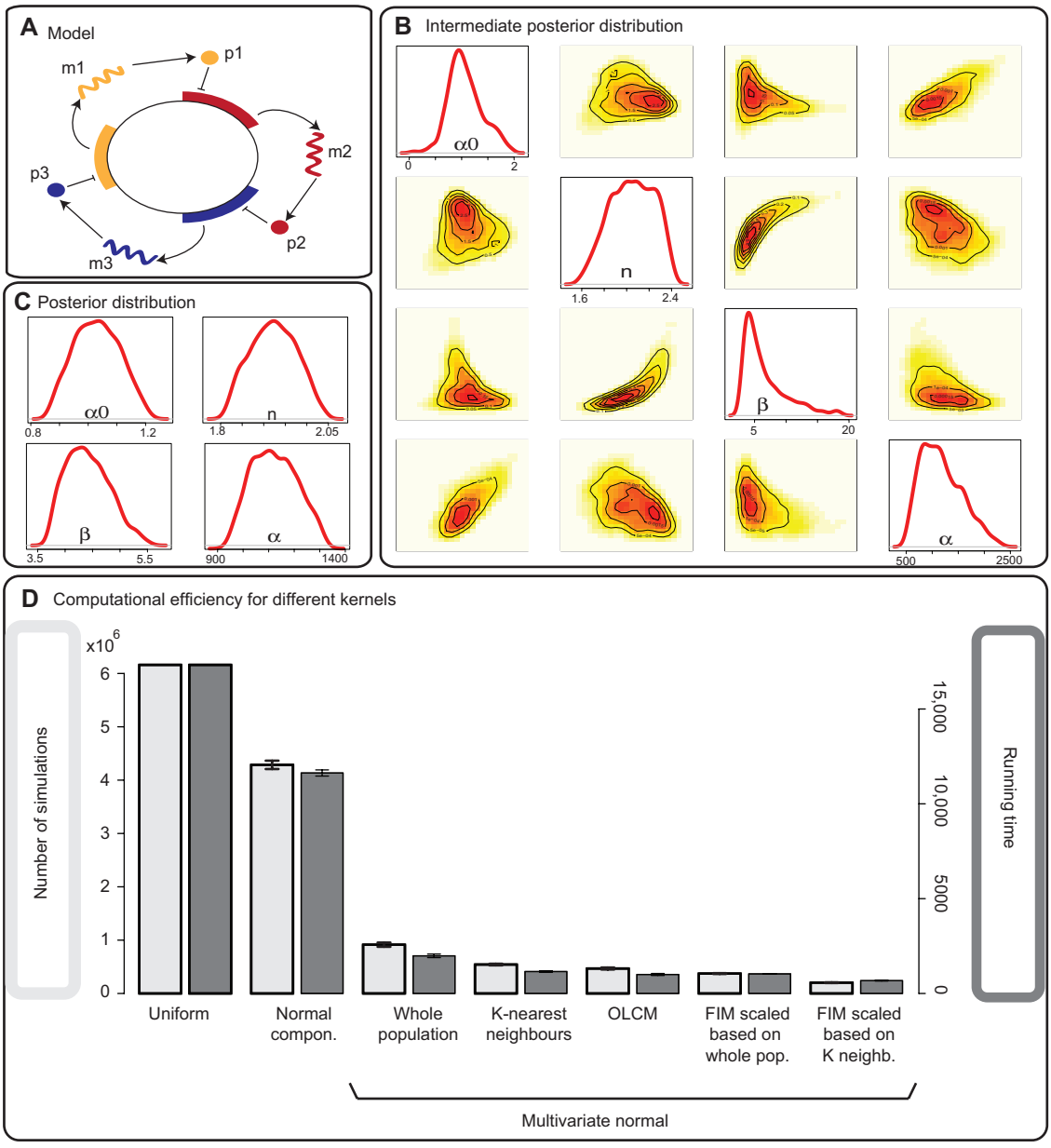

Figure 6 (A) The Repressilator model: three genes are connected in a feedback loop; Each gene *i*, transcribes the repressor protein p_i for the next gene in the loop. (B) Posterior distribution for $\varepsilon = 50$. (C) Marginal posterior distribution for $\varepsilon = 35$. (D) Number of simulations required to obtain the posterior distribution and running time of the algorithm (in min) for each kernel. The error bars on the barplots show the variance of the number of simulation and running time over 10 (resp.3) independent simulations for the multivariate normal kernels (resp. for the normal componentwise kernel). The mRNA measurements are the following: $(m_1(k))_k$=(0, 2.04, 32.29, 4.13, 2.15, 5.09, 1.07, 3.67, 39.01, 73.83, 8.54, 17.62, 11.96), $(m_2(k))_k$ =(0, 28.99, 32.29, 10.61, 55.27, 9.49, 68.56, 10.62, −1.95, 3.53, 63.87, 39.68, −0.6) and $(m_3(k))_k$ =(0, 20.96, 7.49, 44.25, 7.12, 60.52, 8.10, 63.76, 22.9, 6.27, 10.59, 6.50, 70.56).

$(\alpha_0, n, \beta, \alpha) = (1, 2, 5, 1000)$, and measuring the concentration of mRNA at time-points (0.0, 0.6, 4.2, 6.2, 8.6, 13.4, 16, 21.4, 27.6, 34.4, 39.8, 40.6, 45.2), subject to some added zero-mean Gaussian noise with variance 5. The same problems obviously prevail for different datasets.

The ABC SMC algorithm is used to estimate $p(\theta|\{m_1(k), m_2(k), m_3(k)\}_k)$ with N=1000 particles and a decreasing sequence of thresholds equal to (160, 150, 140, 130, 120, 100, 80, 50, 40, 37, 35). The marginal posterior distribution is represented in Figure 6C and agrees very well with what is known from previous studies

(Toni et al., 2009). An intermediate posterior distribution is represented in Figure 6B and shows a highly non-linear correlation between some parameters, in particular parameters n and β.

In Figure 6D we compare the cumulative number of sampled data over the algorithm as well as the running time of the algorithm for different perturbation kernels. Using the uniform kernel, up to 6×10^6 simulations are required to obtain an approximation of the posterior distribution whereas $<1\times10^6$ simulations are required if a multivariate normal kernel is used with, in particular, only 1.6×10^5 simulations if the second version of the multivariate normal kernel based on the FIM is used. Moreover, we observe that the running time of the algorithm for each kernel is proportional to the number of simulations; so the main computational cost of the algorithm is due to the simulation of the data. Therefore, the time spent on defining the kernels, including determining the nearest neighbours of each particle, or evaluating the FIM for each parameter is small compared to the time saved by proposing new particles more efficiently. So even in this simple model a significant improvement is possible through the appropriate choice of perturbation kernel.

6.2 The Hes1 model

We now compare the efficiency of the perturbation kernels on an experimental dataset. We consider a dynamical system describing the expression level of the transcription factor Hes1, which plays an important role in cell differentiation and the segmentation of vertebrate embryos. Oscillations of Hes1 expression level has been observed by Hirata et al. (2002). The Hes1 oscillator can be modelled by a system of three ordinary differential equations describing the evolution of the Hes1 mRNA, m, the Hes1 cytosolic protein, $p1$, and the Hes1 nuclear protein, $p2$ as follows [see Figure 7A and Silk et al. (2011)]:

$$\begin{aligned}
\frac{dm}{dt} &= -k_{deg}m + \frac{1}{1+(p_2/P_0)^h} \\
\frac{dp_1}{dt} &= -k_{deg}p_1 + \nu m - k_1 p_1 \\
\frac{dp_2}{dt} &= -k_{deg}p_2 + k_1 p_1 .
\end{aligned}$$

The model is parametrised by the protein degradation rate k_{deg} which is assumed to be the same for both cytoplasmic and nuclear proteins, the rate of transport k_1 of Hes1 protein into the nucleus, the amount P_0 of Hes1 protein in the nucleus, when the rate of transcription of Hes1 mRNA is at half its maximal value, the rate ν of translation of Hes1 mRNA and the Hill coefficient h.

To infer the parameters of this model we use the real-time PCR measurements published by Silk et al. (2011): the concentration of mRNA is measured every 30 min over 3 h and is equal to $(m_k)_k$ = (2, 1.20, 5.90, 4.58, 2.64, 5.38, 6.42, 5.60, 4.48) (see Figure 7). We aim at inferring the four parameters $\{P_0, \nu, k_1, h\}$; the degradation rate k_{deg} is equal to the experimentally determined value of 0.03. According to the data and some preliminary results, we fix the initial concentration of mRNA to be equal to 2, the initial concentration of p_1 to 5 and the one of p_2 to 3. The ABC SMC algorithm is used to estimate the posterior probability distribution with N = 1000 particles and a decreasing sequence of thresholds equal to (20, 13, 10, 6, 5, 4, 3, 2.8, 2.7, 2.6, 2.5). The posterior distribution is represented in Figure 7 and is identical for every perturbation kernel. To compare the performance of the perturbation kernels we show in Figure 7C the number of simulations during the algorithm as well as its running time for each kernel independently. We observe that the multivariate normal kernels outperforms the uniform and normal component-wise kernels with a 4-fold increase of the speed for the kernel based on the M-nearest neighbours with M=50. Contrary to the repressilator model studied in the previous section, in the Hes1 dynamical system, the kernels based on the FIM do not perform well, with a running time comparable to the normal component-wise kernel. This may be explained by the fact that we use real data which may not be entirely explained by the model. In such

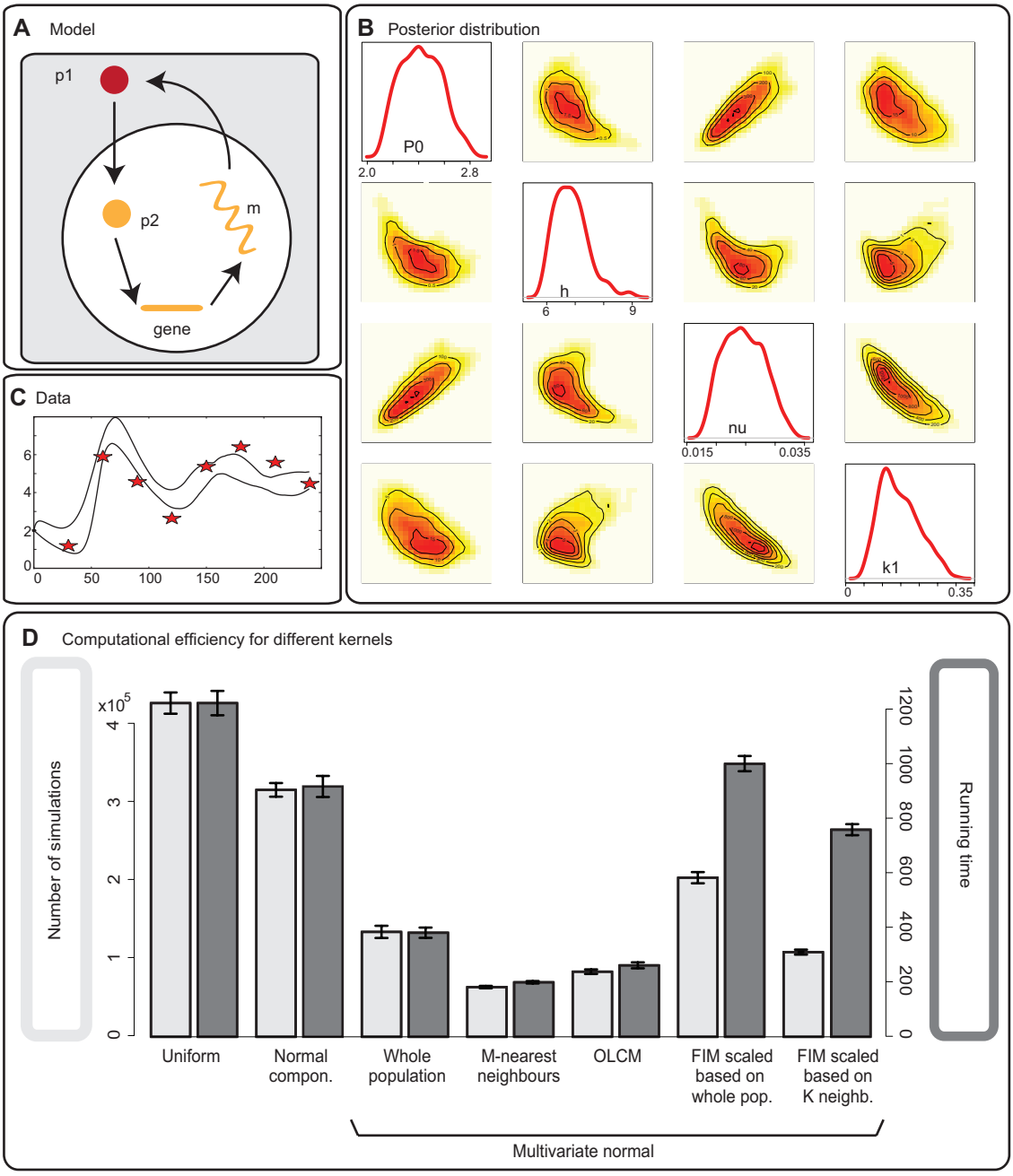

Figure 7 (A) Hes1 transcriptional regulation model: the Hes1 protein migrates from the cytoplasm to the nucleus where it regulates the transcription of the mRNA which then translates the cytoplasmic protein. (B) Posterior distribution. (C) Time-course of mRNA concentration: The stars represent the data published in Silk et al. (2011), the lines represent the minimum and maximum values for the evolution of the species for 1000 particles sampled from the posterior distribution. (D) Number of simulations required to obtain the posterior distribution, and running time of the algorithm (in min) for each kernel. The error bars on the barplots show the variance of the number of simulations and running time over ten independent simulations for each perturbation kernel.

a situation the likelihood surface may not be smooth enough for the FIM to offer much help. By contrast, however, the performance of the OLCM perturbation kernel is similar to that of the 50-nearest neighbours kernel, with the advantage of being free of any parameter choice. For such a model we would clearly recommend the use of the OLCM perturbation kernel.

7 Conclusion

In contrast to MCMC, where the pivotal role of perturbation kernels for convergence and mixing has been well documented (Gilks et al., 1996; Robert and Casella, 2004), for ABC SMC approaches there has been comparatively little work. In particular in the ABC context, which often relies on computationally costly simulation routines, poor choice of the perturbation kernel will result in potentially prohibitive computational overheads. We have addressed this lack of suitable kernels here in a rigorous but non-exhaustive fashion by focusing on kernels that are based around uniform or normal/multivariate normal parametric families. Importantly, in all the examples we were able to ensure that the different kernels had arrived at essentially identical posterior distributions, and for fixed ε_t schedule we can use the acceptance rate as an objective criterion for the numerical efficiency of different kernels.

For all these models it is relatively straightforward to construct optimality criteria by reference to the KL divergence following Beaumont et al. (2009). In higher-dimensional parameter spaces it is important to take into account the potentially correlated nature of parameters, and, not surprisingly we find that component-wise perturbation of particles tends to perform poorly compared to the other approaches considered here. In more complicated cases, e.g. decidedly non-Gaussian posteriors, multimodal posteriors, or posteriors with ridges, we find that a straightforward multivariate normal kernel is in turn inferior to kernels that are conditioned on the local environment of a particle.

In most applications of interest, the computational cost of simulating the data exceeds the algorithmic complexity $O(N^2)$ of the ABC SMC scheme. We therefore argue that the choice of a kernel with a high acceptance rate enables users to optimize the computational cost. However, when two kernels have the same acceptance rate — which may happen for some shapes of the posterior — it is more appropriate to select the one which is cheaper in terms of algorithmic complexity. The following table summarises the computational cost of implementing the proposed perturbation kernels from a previous population of N particles with dimension d (the number of individual parameters). In the case of the multivariate normal kernel based on the FIM, we denote by C the computational cost of simulating an observation, e.g. by solving the set of ODEs or SDEs which define the generative model.

Component-wise normal	$O(dN^2)$
Multivariate normal based on the whole previous population	$O(d^2N^2)$
Multivariate normal based on the M nearest neighbours	$O((d+M)N^2+d^2M^2N)$
Multivariate normal with OLCM	$O(d^2N^2)$
Multivariate normal based on the FIM (normalised with entire population)	$O(dCN+d^2N^2)$

As a general rule of thumb we would recommend the use of multivariate kernels with OLCM, which tend to have the highest acceptance rate in our examples and are relatively easy to implement at acceptable computational cost. For some probability models, in particular those describing dynamical systems, the FIM has attracted a lot of attention recently (Amari and Nagaoka, 2007; Arwini and Dodson, 2008; Secrier et al., 2009; Erguler and Stumpf, 2011; Girolami and Calderhead, 2011; Komorowski et al., 2011), and it appears likely that we will be able to exploit these notions, and those of information geometry more generally, fruitfully in ABC SMC. Therefore, where applicable, we tested the use of the FIM to drive the perturbation kernel. Theoretically, and as can be seen in the Repressilator model, this has the advantage of exploring so-called neutral spaces more efficiently while maintaining a high acceptance rate. However, the results on the Hes1 model suggest that the perturbation kernels based on the FIM are not robust enough and can lead to high computational cost in particular for relatively complex biological models with real data which induce a non-smooth likelihood surface.

The cost of local measures based on M nearest neighbours may seem too high to contemplate their use. However, given the increase in acceptance rate that we have observed, and the generally high computational cost of simulating complex data, they can prove to be fruitful. The user should be aware that the efficiency of such a measure strongly depends on the chosen value of M (the smaller the value of M the higher the

acceptance rate) and that a too small value of M can lead to a too local perturbation kernel with a risk of not converging to the posterior distribution as ε goes to 0. This sensitivity to a user-defined parameter makes the M nearest neighbours kernel less user-friendly than OLCM.

The kernels discussed here may seem restrictive, especially to those from a background in evolutionary computation. We may, for example, wish to consider other perturbations to generate new candidate particles, such as recombination (Baragona et al., 2010), as is frequently done in global optimisation. In principle it is possible to include this in ABC SMC approaches, as long as the weights for new particles can be calculated (which turns out to be relatively straightforward for recombination and different cross-over schemes). It has to be kept in mind, however, that these perturbations work best in cases where the parameter space is so under-sampled that random combinations of individual parameters are sufficiently likely to end up in a region with a more favourable cost-function than a local, e.g. gradient-based proposal would. While such strategies have been applied in many optimization settings, their use in Bayesian inference is rare, since generally here the optimal (by whichever criterion) parameter value is of less interest than the distribution as a whole. For maximum a posteriori inference such methods may be fruitfully applied, but here we do not see an obvious advantage (as is also borne out by simulation studies, data not shown).

Kernel choice is one of the obvious means of speeding up ABC SMC inference. Setting the ε schedule optimally is another. The latter is straightforwardly automated by basing the next ε_{t+1} on the acceptance rate obtained during the generation of the intermediate distribution, $p_{\varepsilon_t}(\theta|x)$, But again there is a trade-off to be made between convergence and exhaustive exploration of the parameter space. In particular too gentle a decrease in ε_t may result in loss of particle diversity (Silk et al., 2012). Here we believe that further investigation of FIMs may hold important clues as to how the ε_t are best chosen. This would, for example, resonate with the perspective on ε proposed by Ratmann et al. (2009).

Aknowledgments: SF is funded through an MRC Computational Biology Research Fellowship; JC is supported by BBSRC grant BB/G006997/1; CB and MPHS gratefully acknowledge financial support from the BBSRC (BB/G007934/1); MPHS is a Royal Society Wolfson Research Merit award holder.

References

Amari, S. and H. Nagaoka (2007): Methods of Information Geometry, vol. 191. American Mathematical Society, USA.

Arwini, K. A. and C. T. J. Dodson (2008): Information geometry. Berlin: Springer.

Baragona, R., F. Battaglia and I. Poli (2010): Evolutionary Statistical Procedures. Heidelberg: Springer Verlag.

Barnes, C. P., S. F. Filippi, M. P. H. Stumpf and T. Thorne (2012): "Considerate approaches to constructing summary statistics for abc model selection," Stat. Comput., 12, 1–17.

Barnes, C. P., D. Silk, X. Sheng and M. P. H. Stumpf (2011): "Bayesian design of synthetic biological systems," Proc. Natl. Acad. Sci. USA, 108(37), 15190–15195.

Barthelmé, S. and N. Chopin (2011): ABC-EP: Expectation-propagation for likelihood Bayesian Computation. ICML (Proceedings of the 28th International Conference on Machine Learning).

Beaumont, M. A., W. Zhang and D. J. Balding (2002): "Approximate Bayesian computation in population genetics," Genetics, 162(4), 2025–2035.

Beaumont, M. A., J. M. Cornuet, J. M. Marin and C. P. Robert (2009): "Adaptive approximate Bayesian computation," Biometrika, 96(4), 983–990.

Bishop, C. M. (2006): Pattern Recognition and Machine Learning. Springer New York.

Blum, M.G. B. and O. François (2010): "Non-linear regression models for Approximate Bayesian Computation," Stat. Comput., 20(1), 63–73.

Boyd, S. P. and L. Vandenberghe. (2004): Convex Optimization. Cambridge University Press.

Cappé,O., R. Douc, A. Guillin, J.-M. Marin and C. P. Robert (2008): "Adaptive importance sampling in general mixture classes," Stat. Comput., 18(4), 447–459.

Cornebise, J., É. Moulines and J. Olsson (2008): "Adaptive methods for sequential importance sampling with application to state space models," Stat. Comput., 18(4), 461–480.

Cornuet, J. M., J. M. Marin, A. Mira and C. P. Robert (2012): "Adaptive multiple importance sampling," Scand. J. Stat., 39(4), 798–812.

Cox, D. R. (2006): Principles of Statistical Inference. Cambridge University Press.
Del Moral, P., A. Doucet and A. Jasra (2006): "Sequential Monte Carlo samplers," J. Roy. Stat. Soc. B, 68(3), 411–436.
Del Moral, P., A. Doucet and A. Jasra (2011): "An adaptive sequential monte carlo method for approximate bayesian computation," Stat. Comput., 22, 1–12.
Didelot, X., R. G. Everitt, A. M. Johansen and D. J. Lawson (2011): "Likelihood-free estimation of model evidence," Bayesian Analysis, 6(1), 49–76.
Douc, R., A. Guillin, J.-M. Marin and C. P. Robert (2007): "Convergence of adaptive mixtures of importance sampling schemes," Ann. Stat., 35(1), 420–448.
Drovandi, C. C. and A. N. Pettitt (2011): "Estimation of parameters for macroparasite population evolution using approximate bayesian computation," Biometrics, 67(1), 225–233.
Efron, B. (2010): Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction. Cambridge University Press.
Elowitz, M. B. and S. Leibler (2000): "A synthetic oscillatory network of transcriptional regulators," Nature, 403(6767), 335–338.
Erguler, K. and M. P. H. Stumpf (2011): "Practical limits for reverse engineering of dynamical systems: A statistical analysis of sensitivity and parameter inferability in systems biology models," Mol. BioSystems, 7(5), 1593–1602.
Fagundes, N. J. R., N. Ray, M. Beaumont, S. Neuenschwander, F. M. Salzano, S. L. Bonatto and L. Excoffier (2007): "Statistical evaluation of alternative models of human evolution," Proc. Natl. Acad. Sci. USA, 104(45), 17614.
Fearnhead, P. and D. Prangle (2012): "Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation," J. R. Stat. Soc. B., 74(3), 419–474.
Gilks, W. R., S. Richardson and D. J. Spiegelhalter (1996): Markov Chain Monte Carlo in Practice. Chapman & Hall/CRC.
Girolami, M. and B. Calderhead (2011): "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," J. Roy. Stat. Soc. B, 73, 123–214.
Givens, G. H. and A. E. Raftery (1996): "Local adaptive importance sampling for multivariate densities with strong nonlinear relationships," J. Am. Stat. Assoc., 132–141.
Gutenkunst, R. N., J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers and J. P. Sethna (2007): "Universally sloppy parameter sensitivities in systems biology models," PLoS Comput. Bio., 3(10), e189.
Haario, H., E. Saksman and J. Tamminen (1999): "Adaptive proposal distribution for random walk metropolis algorithm," Comput. Stat., 14(3), 375–396.
Hirata, H., S. Yoshiura, T. Ohtsuka, Y. Bessho, T. Harada, K. Yoshikawa and R. Kageyama (2002): "Oscillatory expression of the bhlh factor hes1 regulated by a negative feedback loop," Science's STKE 298(5594), 840.
Jasra, A., S. S. Singh, J. S. Martin and E. McCoy (2010): "Filtering via approximate Bayesian computation," Stat. Comput., 22.
Komorowski, M., M. J. Costa, D. A. Rand and M. P. H. Stumpf (2011): "Sensitivity, robustness and identifiability in stochastic chemical kinetics models," Proc. Natl. Acad. Sci. USA, 108(21), 8645–8650.
Lehmann, E. L. and G. Casella (1998): Theory of Point Estimation, Vol. 31. New York: Springer Verlag.
Liepe, J., C. Barnes, E. Cule, K. Erguler, P. Kirk, T. Toni and M. P. H. Stumpf (2010): "ABC-SysBio-Approximate Bayesian computation in Python with GPU support," Bioinformatics, 26(14), 1797–1799.
Lopes, J. S. and M. A. Beaumont (2010): "ABC: a useful Bayesian tool for the analysis of population data," Infect. Gene. Evol., 10(6), 825–832.
MacKay, D. J. C. (2003): Information Theory, Inference and Learning Algorithms. Cambrige University Press.
Marin, J. M., P. Pudlo, C. P. Robert and R. J. Ryder (2011): "Approximate Bayesian computational methods," Stat. Comput., 1–14.
Marjoram, P. and J. Molitor (2003): "Markov chain Monte Carlo without likelihoods," Proc. Natl. Acad. Sci. USA, 100(26), 15324.
McKinley, T., A. R. Cook and R. Deardon (2009): "Inference in epidemic models without likelihoods," Int. J. Biostatistics, 5(1).
Oh, M. S. and J. O. Berger (1993): "Integration of multimodal functions by Monte Carlo importance sampling," J. Am. Stat. Assoc., 88(422), 450–456.
Pitt, M. K. and N. Shephard (1999): "Filtering via simulation: auxiliary particle filters," J. Am. Stat. Assoc., 590–599.
Pritchard, J. K., M. T. Seielstad, A. Perez-Lezaun and M. W. Feldman (1999): "Population growth of human Y chromosomes: a study of Y chromosome microsatellites," Mol. Biol. Evol. 16(12), 1791–1798.
Rao, C. R. (1945): "Information and accuracy attainable in the estimation of statistical parameters," Bull. Calcutta Math. Soc., 37, 81–91.
Ratmann, O., C. Andrieu, C. Wiuf and S. Richardson (2009): "Model criticism based on likelihood-free inference, with an application to protein network evolution," Proc. Natl. Acad. Sci., USA, 106(26), 10576–10581.
Ratmann, O., O. Jørgensen, T. Hinkley, M. Stumpf, S. Richardson and C. Wiuf (2007): Using likelihood-free inference to compare evolutionary dynamics of the protein networks of H. pylori and P. falciparum. PLoS Comput. Bio., 3(11), e230.
Robert, C. P. and G. Casella (2004): Monte Carlo Statistical Methods. New York: Springer Verlag.
Robert, C. P., J. M. Cornuet, J. M. Marin and N. S. Pillai (2011): "Lack of confidence in approximate Bayesian computation model choice," Natl. Acad. Sci., 108, 15112–15117.
Secrier, M., T. Toni and M. P. H. Stumpf (2009): "The ABC of reverse engineering biological signalling systems," Mol. BioSystems, 5(12), 1925–1935.
Silk, D, S. Filippi and M. P. H. Stumpf (2012): "Optimizing threshold – schedules for approximate Bayesian computation sequential Monte Carlo samplers: applications to molecular systems," *arXiv*, 1210.3296, 10

Silk, D., P. D. W. Kirk, C. P. Barnes, T. Toni, A. Rose, S. Moon, M. J. Dallman and M. P. H. Stumpf (2011): "Designing attractive models via automated identification of chaotic and oscillatory dynamical regimes," Nat. Commun., 2, 489.

Sisson, S. A., Y. Fan and M. M. Tanaka (2007): "Sequential Monte Carlo without likelihoods," Proc. Natl. Acad. Sci. USA, 104(6), 1760–1765.

Stigler, S. M. (1986): The History of Statistics: The Measurement of Uncertainty Before 1900. Belknap Press.

Tallmon, D. A., G. Luikart and M. A. Beaumont (2004): "Comparative evaluation of a new effective population size estimator based on approximate bayesian computation," Genetics, 167(2), 977–988.

Tanaka, M. M., A. R. Francis, F. Luciani and S. A. Sisson (2006): "Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data," Genetics, 173(3), 1511–1520.

Toni, T., D. Welch, N. Strelkowa, A. Ipsen and M. P. H. Stumpf (2009): "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," J. Royal Soc. Int./the Royal Soc., 6(31), 187–202.

Toni, T. and M. P. H. Stumpf (2009): "Simulation-based model selection for dynamical systems in systems and population biology," Bioinformatics 26(1), 104–110.

Van, D. M. R., A. Doucet, N. De Freitas and E. Wan (2001): "The unscented particle filter," Adv. Neural Inform. Proc. Sys., 584–590.