Efficient acquisition rules for model-based approximate Bayesian computation

Marko Järvenpää[1], **Michael Gutmann**[2], **Arijus Pleska**[1], **Aki Vehtari**[1], **and Pekka Marttinen**[1]

[1]Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Helsinki, Finland
[2]School of Informatics, University of Edinburgh, UK

Abstract

Approximate Bayesian computation (ABC) is a method for Bayesian inference when the likelihood is unavailable but simulating from the model is possible. However, many ABC algorithms require a large number of simulations and running the simulation model can be costly. To reduce the computational cost, Bayesian optimisation (BO) and surrogate models such as Gaussian processes have been proposed. Bayesian optimisation enables one to intelligently decide where to evaluate the model next, but standard BO strategies are designed for optimisation and not specifically for ABC inference. Our paper addresses this gap in the literature. We propose to compute the uncertainty in the ABC posterior density, which is due to lack of simulations to estimate this quantity accurately, and define a loss function that measures this uncertainty. We then propose to select the next evaluation location to minimise the expected loss. Experiments show that the proposed method often produces the most accurate approximations as compared to common BO strategies. Note: this work is currently under review in a journal and a full-length version is available as a non-refereed pre-print (https://arxiv.org/abs/1704.00520)

1 Introduction

We consider the problem of Bayesian inference of some unknown parameter $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^p$ of a simulation model. Such models are typically not amenable to any analytical treatment and can be only simulated with any parameter $\boldsymbol{\theta} \in \Theta$ to produce data $\mathbf{x}_{\boldsymbol{\theta}} \in \mathcal{X}$. This renders standard Bayesian inference techniques inapplicable because the likelihood function cannot be directly evaluated. Approximate Bayesian computation (ABC) replaces likelihood evaluations with model simulations, see e.g. [14, 22, 12] for an overview. The main idea of the basic ABC algorithm is to draw a parameter value from the prior distribution, simulate a data set with the given parameter value, and accept the value as a draw from the (approximate) posterior if the discrepancy between the simulated and observed data is small enough. This algorithm produces samples from the approximate posterior

$$\pi_{\text{ABC}}(\boldsymbol{\theta} \,|\, \mathbf{x}_{obs}) \propto \pi(\boldsymbol{\theta}) \int \pi_{\varepsilon}(\mathbf{x}_{obs} \,|\, \mathbf{x}) \pi(\mathbf{x} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{x}, \tag{1}$$

where $\pi(\boldsymbol{\theta})$ is the prior probability density, $\pi_{\varepsilon}(\mathbf{x}_{obs} \,|\, \mathbf{x}) \propto \mathbb{1}_{\Delta(\mathbf{x}_{obs}, \mathbf{x}) \leq \varepsilon}$ and $\Delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is the discrepancy that tells how different the simulated and observed data sets are. The threshold ε controls the trade-off between the accuracy of the approximation and computational cost.

Algorithms based on Markov Chain and sequential Monte Carlo [15, 20, 1, 21, 14, 11] as well as different modelling strategies, e.g. [25, 2, 3, 4, 18], and especially Gaussian process (GP) modelling [24, 16, 10, 5, 9], have been proposed to improve the computational efficiency of ABC. In [5] Bayesian optimisation (BO) was used to speed up inference. However, the BO strategies they used to demonstrate their approach have not been designed for ABC originally.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

In the following, we formulate the ABC inference in a fully probabilistic and query-efficient framework. Specifically, we use Bayesian modelling to quantify the "computational uncertainty" in the ABC posterior curve (or some related quantity of interest) which is due to the lack of simulations to estimate it accurately. We then define a loss function $\mathcal{L}$ that could measure e.g. the overall uncertainty in the probability density π_{ABC} or the uncertainty in a particular point estimate of interest such as posterior mean. In this framework, our aim is to choose the next evaluation location $\boldsymbol{\theta}^*$ such that the expected loss, after simulating the model at this location, is minimised. That is, we minimise

$$\mathbb{E}_{\mathbf{x}^* \mid \boldsymbol{\theta}^*, D_{1:t}}(\mathcal{L}(\pi(\pi_{\text{ABC}} \mid \mathbf{x}^*, \boldsymbol{\theta}^*, D_{1:t}))) = \int \mathcal{L}(\pi(\pi_{\text{ABC}} \mid \mathbf{x}^*, \boldsymbol{\theta}^*, D_{1:t}))\pi(\mathbf{x}^* \mid \boldsymbol{\theta}^*, D_{1:t}) \, \mathrm{d}\mathbf{x}^*, \quad (2)$$

with respect to $\boldsymbol{\theta}^*$, where we need to average over the unknown simulator output $\mathbf{x}^*$ at parameter $\boldsymbol{\theta}^*$ using our model for the new simulator output $\pi(\mathbf{x}^* \mid \boldsymbol{\theta}^*, D_{1:t})$ and where $D_{1:t}$ is the training data of simulator-output, parameter pairs gathered this far. In this text we focus on sequential setting and myopic strategies (i.e. selecting one evaluation parameter at a time) although our approach can be generalised. Entropy search in global optimisation [7, 8] and probabilistic numerics literature [6] share similar ideas. However, different from these approaches, our interest is to design the evaluations to minimise the uncertainty in a quantity that itself describes the uncertainty of the parameters of a costly simulation model.

2 GP modelling for ABC and optimal acquisition

The framework briefly outlined above requires some modelling assumptions. While other choices are also possible, as in [5, 9], we model the discrepancy by a Gaussian distribution for each parameter value $\boldsymbol{\theta}$, i.e. $\Delta_{\boldsymbol{\theta}} \sim \mathcal{N}(f(\boldsymbol{\theta}), \sigma_n^2)$ for some unknown suitably smooth function $f : \Theta \to \mathbb{R}$ and variance $\sigma_n^2 \in \mathbb{R}_+$ both of which need to be estimated. We place a Gaussian process prior on f so that $f \sim \mathcal{GP}(\mu(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}'))$. We set $\mu(\boldsymbol{\theta}) = 0$ and use the squared exponential covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \exp(-\sum_{i=1}^p (\theta_i - \theta_i')^2/(2l_i^2))$. Conditioned on the obtained training data $D_{1:t} = \{(\Delta_i, \boldsymbol{\theta}_i)\}_{i=1}^t$ and the GP hyperparameters $\boldsymbol{\phi} = (\sigma_f^2, l_1, \ldots, l_p, \sigma_n^2)$, our knowledge of the function f evaluated at an arbitrary point $\boldsymbol{\theta} \in \Theta$ is $f(\boldsymbol{\theta}) \mid D_{1:t}, \boldsymbol{\theta}, \boldsymbol{\phi} \sim \mathcal{N}(m_{1:t}(\boldsymbol{\theta}), v_{1:t}^2(\boldsymbol{\theta}))$, where

$$m_{1:t}(\boldsymbol{\theta}) = k(\boldsymbol{\theta}, \boldsymbol{\theta}_{1:t}) K^{-1}(\boldsymbol{\theta}_{1:t}) \Delta_{1:t}, \quad v_{1:t}^2(\boldsymbol{\theta}) = k(\boldsymbol{\theta}, \boldsymbol{\theta}) - k(\boldsymbol{\theta}, \boldsymbol{\theta}_{1:t}) K^{-1}(\boldsymbol{\theta}_{1:t}) k(\boldsymbol{\theta}_{1:t}, \boldsymbol{\theta}) \quad (3)$$

and $K(\boldsymbol{\theta}_{1:t}) = k(\boldsymbol{\theta}_{1:t}, \boldsymbol{\theta}_{1:t}) + \sigma_n^2 \mathbf{I}$. Above we defined $k(\boldsymbol{\theta}, \boldsymbol{\theta}_{1:t}) = (k(\boldsymbol{\theta}, \boldsymbol{\theta}_1), \ldots, k(\boldsymbol{\theta}, \boldsymbol{\theta}_t))^T$ and similarly for $k(\boldsymbol{\theta}_{1:t}, \boldsymbol{\theta})$, $k(\boldsymbol{\theta}_{1:t}, \boldsymbol{\theta}_{1:t})_{ij} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ for $i, j = 1, \ldots, t$ and $\Delta_{1:t} = (\Delta_1, \ldots, \Delta_t)^T$.

In principle, the uncertainty in GP hyperparameters $\boldsymbol{\phi}$ can be taken into account but in the following we assume that $\boldsymbol{\phi}$ is either known or MAP-estimate is used to determine its value. If we knew f, the (unnormalised) ABC posterior $\tilde{\pi}_{\text{ABC}}(\boldsymbol{\theta})$ and the acceptance probability $p_a(\boldsymbol{\theta})$ could be computed as $\tilde{\pi}_{\text{ABC}}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}) p_a(\boldsymbol{\theta})$, and $p_a(\boldsymbol{\theta}) = \Phi\left((\varepsilon - f(\boldsymbol{\theta}))/\sigma_n\right)$. We do not know f accurately due to limited training data but, using the GP assumptions, we can derive the probability law and closed-form equations of different statistics for $\tilde{\pi}_{\text{ABC}}$ for each $\boldsymbol{\theta}$. For instance, we obtain

$$\mathbb{E}(\tilde{\pi}_{\text{ABC}}(\boldsymbol{\theta}) \mid D_{1:t}) = \pi(\boldsymbol{\theta}) \, \Phi\left((\varepsilon - m_{1:t}(\boldsymbol{\theta}))/\sqrt{\sigma_n^2 + v_{1:t}^2(\boldsymbol{\theta})}\right). \quad (4)$$

We consider the following loss function for model-based ABC inference

$$\mathcal{L}(\pi(\pi_{\text{ABC}} \mid D_{1:t})) = \int_\Theta \mathbb{V}(\tilde{\pi}_{\text{ABC}}(\boldsymbol{\theta}) \mid D_{1:t}) \, \mathrm{d}\boldsymbol{\theta} = \int_\Theta \pi^2(\boldsymbol{\theta}) \mathbb{V}(p_a(\boldsymbol{\theta}) \mid D_{1:t}) \, \mathrm{d}\boldsymbol{\theta}, \quad (5)$$

where the variance is taken with respect to $\pi(f \mid D_{1:t})$. We call the function as the integrated variance loss function. It measures the uncertainty in the unnormalised ABC posterior density averaged over the parameter space Θ. Based on our GP assumptions, the expected integrated variance (abbreviated as "expintvar") after running the simulation model with parameter $\boldsymbol{\theta}^*$ can be shown to be

$$L_{1:t}(\boldsymbol{\theta}^*) = \mathbb{E}_{\Delta^* \mid \boldsymbol{\theta}^*, D_{1:t}} \int_\Theta \pi^2(\boldsymbol{\theta}) \mathbb{V}(p_a(\boldsymbol{\theta}) \mid \Delta^*, \boldsymbol{\theta}^*, D_{1:t}) \, \mathrm{d}\boldsymbol{\theta} \quad (6)$$

$$= 2 \int_\Theta \pi^2(\boldsymbol{\theta}) \left[T\left(\frac{\varepsilon - m_{1:t}(\boldsymbol{\theta})}{\sqrt{\sigma_n^2 + v_{1:t}^2(\boldsymbol{\theta})}}, \sqrt{\frac{\sigma_n^2 + v_{1:t}^2(\boldsymbol{\theta}) - \tau_{1:t}^2(\boldsymbol{\theta}, \boldsymbol{\theta}^*)}{\sigma_n^2 + v_{1:t}^2(\boldsymbol{\theta}) + \tau_{1:t}^2(\boldsymbol{\theta}, \boldsymbol{\theta}^*)}} \right) \right.$$
$$\left. - T\left(\frac{\varepsilon - m_{1:t}(\boldsymbol{\theta})}{\sqrt{\sigma_n^2 + v_{1:t}^2(\boldsymbol{\theta})}}, \frac{\sigma_n}{\sqrt{\sigma_n^2 + 2v_{1:t}^2(\boldsymbol{\theta})}} \right) \right] \mathrm{d}\boldsymbol{\theta}, \quad (7)$$

where the variance of $p_a(\boldsymbol{\theta})$ is taken with respect to $\pi(f \mid \Delta^*, \boldsymbol{\theta}^*, D_{1:t})$, the function $T(\cdot, \cdot)$ is the Owen's t-function that can be computed efficiently using the algorithm in [19] and that satisfies

$$T(h, a) = \frac{1}{2\pi} \int_0^a \frac{e^{-h^2(1+x^2)/2}}{1+x^2} \, \mathrm{d}x, \tag{8}$$

and $\tau^2_{1:t}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \mathrm{cov}^2_{1:t}(\boldsymbol{\theta}, \boldsymbol{\theta}^*)/(\sigma_n^2 + v^2_{1:t}(\boldsymbol{\theta}^*))$, where $\mathrm{cov}_{1:t}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = k(\boldsymbol{\theta}, \boldsymbol{\theta}^*) - k(\boldsymbol{\theta}, \boldsymbol{\theta}_{1:t}) K^{-1}(\boldsymbol{\theta}_{1:t}) k(\boldsymbol{\theta}_{1:t}, \boldsymbol{\theta}^*)$ is the posterior covariance between the evaluation point $\boldsymbol{\theta}$ and the candidate location for the next evaluation $\boldsymbol{\theta}^*$. Derivation of this result can be found in our full paper.

Computing the integral in Equation (7) can be done using grid integration in low dimensions or using importance sampling (IS). In the latter case the IS proposal π_q can be chosen to be current variance surface interpreted as a probability density function (pdf) which has similar formula as the integrand in Equation (7). Sampling from π_q is not straightforward but can be done using standard MCMC techniques. The new evaluation location is chosen to minimise the expected loss, that is

$$\boldsymbol{\theta}_{t+1} \in \{\boldsymbol{\theta} \in \Theta : \boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}^* \in \Theta} L_{1:t}(\boldsymbol{\theta}^*)\}, \tag{9}$$

where the right hand side is a set of parameters because the minimiser may not be unique. The resulting algorithm for estimating the ABC posterior is outlined as Algorithm 1.

Algorithm 1 GP-based ABC inference using the expected integrated variance acquisition function.

1: Generate initial training locations $\boldsymbol{\theta}_{1:t_0} \sim \pi(\cdot)$
2: **for** $t = 1 : t_0$ **do**
3: Simulate $\mathbf{x}_t \sim \pi(\cdot \mid \boldsymbol{\theta}_t)$ and compute $\Delta_t \leftarrow \Delta(\mathbf{x}_{obs}, \mathbf{x}_t)$
4: **end for**
5: **for** $t = t_0 : t_{\max} - 1$ **do**
6: Estimate GP hyperparameters $\boldsymbol{\phi}^{\mathrm{MAP}}_{1:t}$ using $D_{1:t}$
7: Precompute Cholesky factorisation for the GP prediction
8: Simulate evaluation points $\boldsymbol{\theta}^{(i)}$ and IS weights $\omega^{(i)}$ for $i = 1, \ldots, s$ by sampling from $\pi_q(\cdot)$
9: Precompute the second term in Equation (7)
10: Obtain $\boldsymbol{\theta}_{t+1}$ by solving the optimisation problem in Equation (9)
11: Simulate $\mathbf{x}_{t+1} \sim \pi(\cdot \mid \boldsymbol{\theta}_{t+1})$ and compute $\Delta_{t+1} \leftarrow \Delta(\mathbf{x}_{obs}, \mathbf{x}_{t+1})$
12: Update the training data $D_{1:t+1} \leftarrow D_{1:t} \cup \{(\Delta_{t+1}, \boldsymbol{\theta}_{t+1})\}$
13: **end for**
14: Estimate GP hyperparameters $\boldsymbol{\phi}^{\mathrm{MAP}}_{1:t_{\max}}$ using $D_{1:t_{\max}}$
15: Simulate samples $\boldsymbol{\vartheta}_{1:n}$ from the estimate of ABC posterior given by Equation (4)
16: **return** $\boldsymbol{\vartheta}_{1:n}$ as a sample from the approximate posterior density

Instead of strictly following the outlined framework, one could also evaluate the simulation model where the current uncertainty of $\tilde{\pi}_{\mathrm{ABC}}$ is maximised, or choose the location that is expected to produce largest reduction in the uncertainty of $\tilde{\pi}_{\mathrm{ABC}}$ in this particular location. We call these alternative approaches, albeit more heuristic, as maxvar and expdiffvar, respectively. One could also interpret the current variance of $\tilde{\pi}_{\mathrm{ABC}}$ as a pdf and sample the new evaluation point from it (called rand_maxvar).

3 Results

We compare our methods to expected improvement (EI) and lower confidence bound (LCB) criteria. We also draw points sequentially from the uniform distribution, abbreviated as "unif". MATLAB and GPstuff [23] are used for the experiments, however, the algorithms in this article are also available in the ELFI (engine for likelihood-free inference) Python software package [13]. We first consider synthetic 2D simulation models: 1) a unimodal density with two correlated variables, 2) a bimodal density, 3) a density where the first parameter is (almost) unidentifiable, and 4) a banana shaped density, all with a uniform prior. The integration and sampling required by expintvar and rand_maxvar strategies are performed in a 50^2 grid and the initial training set size is $t_0 = 10$. The threshold ε is fixed. TV denotes the total variation distance between the estimated and the true baseline ABC density. The results in Figure 1 show that the expintvar is the best method overall but also rand_maxvar produces good results. Of the common alternatives, LCB is clearly the best and produces results

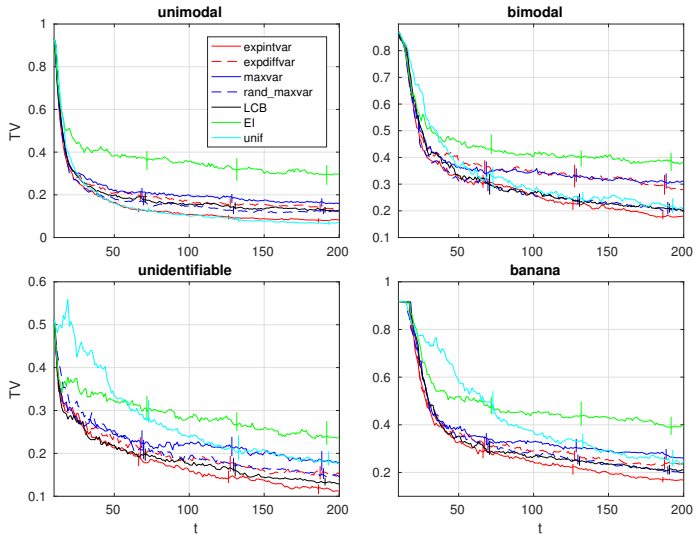

Figure 1: Median of the TV distance between the estimated and the true ABC posterior over 100 experiments. Vertical lines show the 95% confidence interval of the median.

with similar accuracy as rand_maxvar. The performance of the EI strategy is poor due to excessive exploitation.

As an another test case, we consider a simulation model that describes transmission dynamics of bacterial infections in day care centers. The model has three parameters: an internal infection parameter $\beta \in [0, 11]$, an external infection parameter $\Lambda \in [0, 2]$ and a co-infection parameter $\theta \in [0, 1]$ that we estimate. Details of the model and data are described in [17]. We use uniform prior and set the threshold ε adaptively to match the 0.01th quantile of realised discrepancies. Figure 2 shows the results. The performances of the proposed methods are similar and also LCB works well. However, LCB and maxvar tend to underestimate the credible interval while expintvar and rand_maxvar overestimate it. EI performs again the worst. However, the posterior approximations produced by the proposed methods are clearly more accurate than those in [5] who used similar experimental design to illustrate their approach.

We note that the computational overhead caused by our method is negligible compared to the runtime of many complex simulation models. We have also tested the acquisition rules in other test problems with e.g. higher dimension than here. Futhermore, we have observed that our acquisition rules work consistently under informative prior densities of the simulation model and do not necessarily require bounded parameter domain unlike standard BO strategies. These experiments and further analyses can be found in our full paper.

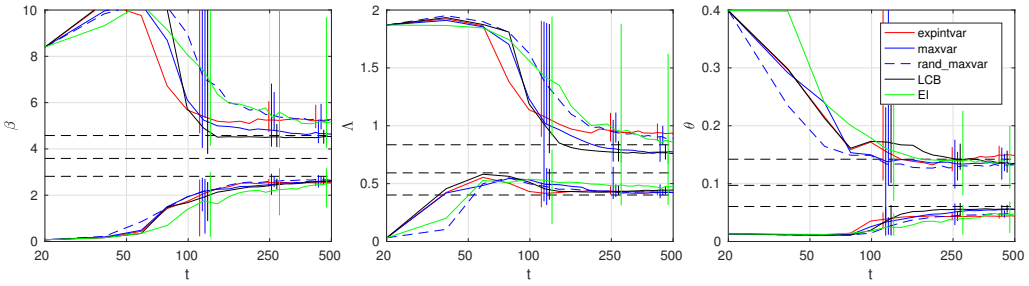

Figure 2: Comparison of the 95% credible interval estimates in the bacterial model. The black dashed lines show the ground truth by [17] and the vertical lines show the 75% interval of the realisations over 100 experiments. The x-axis shows the iterations t on the log-scale.

Acknowledgements

This work was funded by the Academy of Finland (grants no. 286607 and 294015 to PM). We acknowledge the computational resources provided by Aalto Science-IT project.

References

[1] M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009. 1

[2] M. G. B. Blum. Approximate Bayesian Computation: a nonparametric perspective. *Journal of American Statistical Association*, 105(491):1178–1187, 2010. 1

[3] M. G. B. Blum and O. François. Non-linear regression models for Approximate Bayesian Computation. *Statistics and Computing*, 20(1):63–73, 2010. 1

[4] Y. Fan, D. J. Nott, and S. A. Sisson. Approximate Bayesian computation via regression density estimation. *Stat*, 2(1):34–48, 2013. 1

[5] M. U. Gutmann and J. Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47, 2016. 1, 2, 4

[6] P. Hennig, M. A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015. 2

[7] P. Hennig and C. J. Schuler. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 13(1999):1809–1837, 2012. 2

[8] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. *Advances in Neural Information Processing Systems 28*, pages 1–9, 2014. 2

[9] M. Järvenpää, M. Gutmann, A. Vehtari, and P. Marttinen. Gaussian process modeling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria. Available at `https://arxiv.org/abs/1610.06462`, 2017. 1, 2

[10] K. Kandasamy, J. Schneider, and B. Póczos. Bayesian active learning for posterior estimation. In *International Joint Conference on Artificial Intelligence*, pages 3605–3611, 2015. 1

[11] M. Lenormand, F. Jabot, and G. Deffuant. Adaptive approximate Bayesian computation for complex models. *Computational Statistics*, 28(6):2777–2796, 2013. 1

[12] J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and Recent Developments in Approximate Bayesian Computation. *Systematic biology*, 66(1):e66–e82, 2017. 1

[13] J. Lintusaari, H. Vuollekoski, A. Kangasrääsiö, K. Skytén, M. Järvenpää, M. U. Gutmann, A. Vehtari, J. Corander, and S. Kaski. ELFI: Engine for Likelihood Free Inference. Available at `https://arxiv.org/abs/1708.00707`, 2017. 3

[14] J. M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012. 1

[15] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavare. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–8, 2003. 1

[16] E. Meeds and M. Welling. GPS-ABC: Gaussian Process Surrogate Approximate Bayesian Computation. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, 2014. 1

[17] E. Numminen, L. Cheng, M. Gyllenberg, and J. Corander. Estimating the transmission dynamics of streptococcus pneumoniae from strain prevalence data. *Biometrics*, 69(3):748–757, 2013. 4

[18] G. Papamakarios and I. Murray. Fast e-free inference of simulation models with Bayesian conditional density estimation. In *Advances in Neural Information Processing Systems 29*, 2016. 1

[19] M. Patefield and D. Tandy. Fast and accurate Calculation of Owen's T-Function. *Journal of Statistical Software*, 5(5):1–25, 2000. 3

[20] S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–5, 2007. 1

[21] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society, Interface*, 6(31):187–202, 2009. 1

[22] B. M. Turner and T. Van Zandt. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85, 2012. 1

[23] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14:1175–1179, 2013. 3

[24] R. D. Wilkinson. Accelerating ABC methods using Gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 2014. 1

[25] S. N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466:1102–1104, 2010. 1