Methods in Ecology and Evolution British Ecological Society

Methods in Ecology and Evolution doi: 10.1111/j.2041-210X.2011.00179.x

APPLICATION

abc: an R package for approximate Bayesian computation (ABC)

Katalin Csilléry[1]*, Olivier François[2] and Michael G. B. Blum[2]

[1]Irstea, UR EMGR, 2 rue de la Papeterie, F-38402 Saint Martin d'Hères, France; and [2]Computational and Mathematical Biology Team, Laboratoire Techniques de l'Ingénierie Médicale et de la Complexité, Université Joseph Fourier, Grenoble 1, Centre National de la Recherche Scientifique UMR5525, F-38706 La Tronche, France

Summary

1. Many recent statistical applications involve inference under complex models, where it is computationally prohibitive to calculate likelihoods but possible to simulate data. Approximate Bayesian computation (ABC) is devoted to these complex models because it bypasses the evaluation of the likelihood function by comparing observed and simulated data.

2. We introduce the R package 'abc' that implements several ABC algorithms for performing parameter estimation and model selection. In particular, the recently developed nonlinear heteroscedastic regression methods for ABC are implemented. The 'abc' package also includes a cross-validation tool for measuring the accuracy of ABC estimates and to calculate the misclassification probabilities when performing model selection. The main functions are accompanied by appropriate summary and plotting tools.

3. R is already widely used in bioinformatics and several fields of biology. The R package 'abc' will make the ABC algorithms available to a large number of R users. 'abc' is a freely available R package under the GPL license, and it can be downloaded at http://cran.r-project.org/web/packages/abc/index.html.

Key-words: coalescent, model-based inference, neural networks, population genetics

Introduction

In recent years, approximate Bayesian computation (ABC) has become a popular method for parameter inference and model selection under complex models, where the evaluation of the likelihood function is computationally prohibitive. ABC bypasses exact likelihood calculations via the use of summary statistics and simulations, which, in turn, allows the consideration of highly complex models. The name ABC was first coined by Beaumont *et al.* (2002) in population genetics, for inference under coalescent models, but its origin goes back to works by Tavaré *et al.* (1997); Pritchard *et al.* (1999). ABC is now increasingly applied especially in ecology or systems biology (for reviews of ABC methods and applications, see Beaumont 2010; Bertorelle *et al.* 2010; Csilléry *et al.* 2010). Software implementations of ABC dedicated to particular problems have already been developed in these fields (Anderson *et al.* 2005; Hickerson *et al.* 2007; Cornuet *et al.* 2008; Jobin & Mountain 2008; Tallmon *et al.* 2008; Lopes *et al.* 2009; Thornton 2009; Bray *et al.* 2010; Cornuet *et al.* 2010; Liepe *et al.* 2010; Wegmann *et al.* 2010; Huang *et al.* 2011).

The integration of ABC in a software package poses several challenges. First, data simulation, which is in the core of any ABC analysis, is specific to the model in question. Thus, many existing ABC software are specific to a particular class of models (Hickerson *et al.* 2007; Cornuet *et al.* 2008; Lopes *et al.* 2009) or even to the estimation of a particular parameter (Tallmon *et al.* 2008). Further, model comparison is an integral part of any Bayesian analysis; thus, it is essential to provide software, where users are able to fit different models to their data. Second, an ABC analysis often follows a trial–error approach, where users experiment with different models, ABC algorithms or summary statistics. Therefore, it is important that users can run different analyses using batch files, which contain each analysis as a sequence of commands. Third, ABC is subject to intensive research, and many new algorithms have been published in the past few years (Beaumont *et al.* 2002,

*Correspondence author. E-mail: kati.csillery@gmail.com
Correspondence site: http://www.respond2articles.com/MEE/

© 2012 The Authors. Methods in Ecology and Evolution © 2012 British Ecological Society

2009; Bortot *et al.* 2007; Sisson *et al.* 2007; Blum 2010). Thus, an ABC software should be flexible enough to accommodate the new developments of the field.

Here, we introduce a generalist R package 'abc', which aims to address the above challenges (R Development Core Team 2011). The price to pay for the generality and flexibility is that the simulation of data and the calculation of summary statistics are left to the users. However, simulation software might be called from an R session, which opens up the possibility for a highly interactive ABC analysis. For coalescent models, for instance, users can apply one of the many existing software for simulating genetic data such as 'ms' (Hudson 2002) or 'fastsimcoal' (Excoffier & Foll 2011). The calculation of summary statistics could be performed using either R or some specific software such as 'msABC' (Pavlidis *et al.* 2010), which runs 'ms' and calculates summary statistics from the output files. ABC methods have also been developed to handle full data (Sousa *et al.* 2009) – allele frequencies in population genetics – but the 'abc' package is dedicated to summary statistic approaches, which represent the bulk of the literature.

R provides many advantages in the context of ABC: (i) R already possesses the necessary tools to handle, analyse and visualise large data sets, (ii) sequences of R commands can be saved in a script file and (iii) R is a free and collaborative project; thus, new algorithms can be easily integrated to the package (e.g. via contributions from their authors).

Implementation

The main steps of an ABC analysis follow the general scheme of any Bayesian analysis: formulating a model, fitting the model to data (parameter estimation) and improving the model by checking its fit (posterior predictive checks) and comparing it to other models (Gelman *et al.* 2003; Csilléry *et al.* 2010). 'abc' provides functions for the inference and model comparison steps, and generic tools of R can be used for model checking.

To use the package, the following R objects should be prepared: a vector of the observed summary statistics, a matrix of the simulated summary statistics, where each row corresponds to a simulation and each column corresponds to a summary statistic, and finally, a matrix of the simulated parameter values, where each row corresponds to a simulation and each column corresponds to a parameter.

PARAMETER INFERENCE

For the sake of clarity, we recall the general scheme of parameter estimation with ABC. Suppose that we want to compute the posterior probability distribution of a univariate or multivariate parameter, θ. A parameter value θ_i is sampled from its prior distribution to simulate a data set y_i, for $i = 1,\ldots,n$ where n is the number of simulations. A set of summary statistics $S(y_i)$ is computed from the simulated data and compared to the summary statistics obtained from the actual data $S(y_0)$ using a distance measure d. We consider the Euclidean distance for d, and the 'abc' package standardises each summary statistic with a robust estimate of the standard deviation (the median absolute deviation). If $d(S(y_i),S(y_0))$ (i.e. the distance between $S(y_i)$ and $S(y_0)$) is less than a given threshold, the parameter value θ_i is accepted. To set a threshold for d, above which simulations are rejected, the user has to provide the tolerance rate, which is defined as the proportion of accepted simulations. The accepted θ_i's form a sample from an approximation of the posterior distribution. The estimation of the posterior distribution can be improved by the use of regression techniques, which we detail in the following paragraph.

The function `"abc"` implements three ABC algorithms for constructing the posterior distribution from the accepted θ_i's: a rejection method and two regression-based correction methods that use either local linear regression (Beaumont *et al.* 2002) or neural networks (Blum & François 2010). When the rejection method (`"rejection"`) is selected, the accepted θ_i's are considered as a sample from the posterior distribution (Pritchard *et al.* 1999). The two regression methods (`"loclinear"` and `"neuralnet"`) implement an additional step to correct for the imperfect match between the accepted, $S(y_i)$, and observed summary statistics, $S(y_0)$, using the following regression equation in the vicinity of $S(y_0)$

$$\theta_i = m(S(y_i)) + \epsilon_i, \qquad \text{eqn 1}$$

where m is the regression function and the ϵ_i's are centred random variables with equal variance. Simulations that closely match $S(y_0)$ are given more weight by assigning to each simulation $(\theta_i,S(y_i))$ the weight $K[d(S(y_i),S(y_0))]$, and the package implements different statistical kernels K. The local linear model (`"loclinear"`) assumes a linear function for m, while neural networks account for the non-linearity of m and allow users to reduce the dimension of the set of summary statistics. Once the regression is performed, a weighted sample from the posterior distribution is obtained by correcting the θ_i's as follows:

$$\theta_i^* = \hat{m}(S(y_0)) + \hat{\epsilon_i}, \qquad \text{eqn 2}$$

where $\hat{m}(\cdot)$ is the estimated conditional mean and the $\hat{\epsilon_i}$s are the empirical residuals of the regression (Beaumont *et al.* 2002). Additionally, a correction for heteroscedasticity is applied, by default, in `"abc"`,

$$\theta_i^* = \hat{m}(S(y_0)) + \frac{\hat{\sigma}(S(y_0))}{\hat{\sigma}(S(y_i))}\hat{\epsilon_i} \qquad \text{eqn 3}$$

where $\hat{\sigma}(\cdot)$ is the estimated conditional standard deviation (Blum & François 2010).

The function `"abc"` returns an object of class `"abc"` that can be printed, summarised and plotted using the S3 methods of the R generic functions, `"print"`, `"summary"`, `"hist"` and `"plot"`. The function `"print"` returns a description of the object. The function `"summary"` calculates summaries of the posterior distributions, such as the mode, mean, median and credible intervals, taking into account the posterior weights, when appropriate. The `"hist"` function displays the histogram of the weighted posterior sample. The `"plot"` function generates various plots that allow the evaluation of the quality of estimation when one of the regression methods is used. The following plots are generated: a density plot of the prior distribution, a density plot of the posterior distribution estimated with and without regression-based correction, a scatter plot of the Euclidean distances as a function of the parameter values and a normal Q–Q plot of the residuals from the regression. When the heteroscedastic regression model is used, a normal Q–Q plot of the standardised residuals is displayed (see Fig. 1 panel a).

Finally, we note that alternative algorithms exist that sample from an updated distribution that is closer in shape to the posterior than to the prior (Marjoram *et al.* 2003; Beaumont *et al.* 2009; Wegmann *et al.* 2010). However, we do not implement these methods in the 'abc' package because they require the repeated use of the simulation software.

POSTERIOR PREDICTIVE CHECKS

We strongly recommend that users perform posterior predictive checks after fitting their model to the data. There is no specific function in the package 'abc' for posterior predictive checks; nevertheless, the task can be easily carried out using R

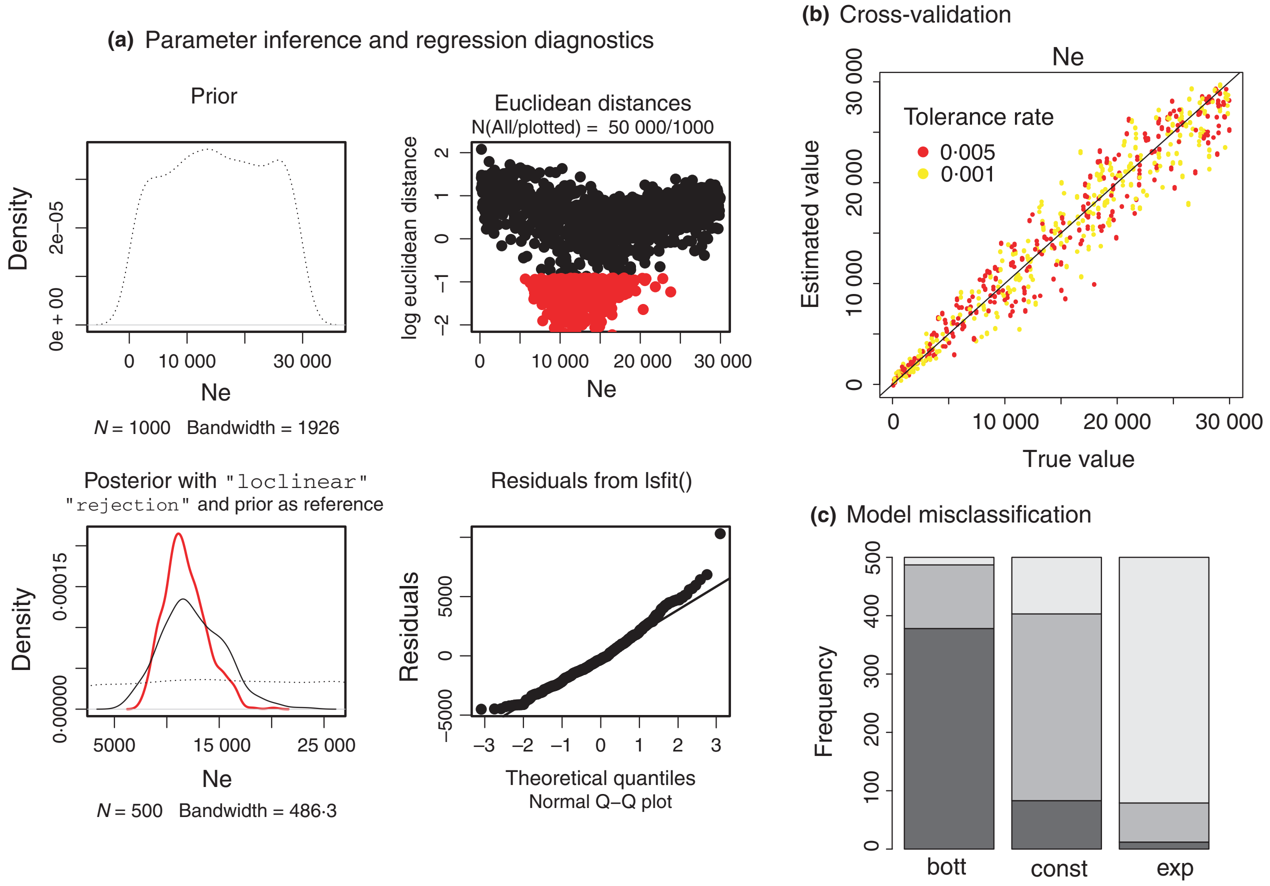

Fig. 1. Typical graphical outputs of the R 'abc' package (model selection and estimation of the effective population size Ne from population genetic data). (a) Parameter inference and regression diagnostics: plots show (clock-wise) the prior distribution, the distances between observed and simulated summary statistics as a function of the parameter values (where red points indicate the accepted values), normal Q–Q plot of the residuals of the regression, and the posterior distribution obtained with and without the regression correction method (and the prior distribution, for reference). (b) Cross-validation for parameter estimation: plot shows the estimated values as a function of true parameter values. Different colours correspond to different values of the tolerance rate. (c) Model misclassification: a graphical illustration of the confusion matrix for three models. The colours from dark to light grey correspond to models `bott`, `const`, `exp`, accordingly. If the simulations were perfectly classified, each bar would have a single colour of its own corresponding model. The following R code can be used to re-generate these plots.

```
> library(abc)
> data(human)
> cv.modsel<-cv4postpr(models, stat.3pops.sim, nval=50, tol=.01, method="mnlogistic")
> plot(cv.modsel)
> stat.italy.sim<-subset(stat.3pops.sim, subset=models=="bott")
> cv.res.reg<-cv4abc(data.frame(Na=par.italy.sim[,"Ne"]), stat.italy.sim,
 +nval=200, tols=c(.005,.001), method="loclinear")
> plot(cv.res.reg, caption="Ne")
> res<-abc(target=stat.voight["italian",], param=data.frame(Na=par.italy.sim[, "Ne"]),
 +sumstat=stat.italy.sim, tol=0.005, transf=c("log"), method="neuralnet")
> plot(res, param=par.italy.sim[, "Ne"])
```

and the simulation software. A fully executable example using R and 'ms' can be found in the package's vignette. Briefly, to perform model checking, one can obtain replicates from the posterior distribution of the parameters using the function `abc`. Then, one can simulate the summary statistics *a posteriori* using the simulation software. In ABC, posterior predictive checks might use the summary statistics twice: once for sampling from the posterior distribution and once for comparing the marginal posterior predictive distributions to the observed values of the summary statistics. To avoid this circularity, we might consider using different summary statistics for posterior predictive checks than for parameter estimation, for example using the expected deviance function.

CROSS-VALIDATION

The function `"cv4abc"` performs a leave-one-out cross-validation to evaluate the accuracy of parameter estimates and the robustness of the estimates to the tolerance rate. To perform cross-validation, the *i*th simulation is randomly selected as a validation simulation, its summary statistic(s) $S(y_i)$ are used as pseudo-observed summary statistics, and its parameters are estimated via `"abc"` using all simulations except the i^{th} simulation. Ideally, the process is repeated *n* times, where *n* is the number of simulations (so-called *n*-fold cross-validation). However, performing an *n*-fold cross-validation might take up too much time, so the cross-validation is often performed for a subset of typically 100 randomly selected simulations. The `"summary"` S3 method of `"cv4abc"` computes the prediction error as

$$E_{\text{pred}} = \frac{\sum_i (\tilde{\theta}_i - \theta_i)^2}{\text{Var}(\theta_i)}, \qquad \text{eqn 4}$$

where θ_i is the true parameter value of the *i*th simulated data set and $\tilde{\theta}_i$ is the estimated parameter value (the posterior median). The `"plot"` function displays the estimated parameter values as a function of the true values (see Fig. 1 panel b).

MODEL SELECTION

The function `"postpr"` implements model selection to approximate the posterior probability of a model M as $Pr(M|S(y_0))$. Three different methods are implemented. With the rejection method (`"rejection"`), the approximate posterior probability of a given model is proportional to the proportion of accepted simulations under this model. The two other methods are based on multinomial logistic regression (`"mnlogistic"`) or neural networks (`"neuralnet"`). In these two approaches, the model indicator is treated as the response variable of a polychotomous regression, where the summary statistics are the independent variables (Beaumont 2008). Using neural networks can be efficient when highly dimensional statistics are used. Any of these methods are valid when the different models to be compared are, *a priori*, equally likely, and the same number of simulations are performed under each model. The `"summary"` S3 method for `"postpr"` displays the approximate posterior model probabilities, and calculates the ratios of model probabilities, the approximate Bayes factor, for all possible pairs of models (François *et al.* 2008).

A further function, `"expected.deviance"`, is implemented to guide the model selection procedure. The function computes an approximate expected deviance from the posterior predictive distribution. Thus, to use the function, users have to re-use the simulation tool and to simulate data from the posterior parameter values. The method is particularly advantageous when it is used with one of the regression methods. Further details on the method can be found in François & Laval (2011), and fully worked out examples are provided in the package's manual pages.

COMPUTING MISCLASSIFICATION ERRORS

A cross-validation tool is available for model selection as well via the function `"cv4postpr"`. The objective is to evaluate whether model selection with ABC is able to distinguish between the proposed models by making use of the existing simulations. The summary statistics from one of the simulations are considered as pseudo-observed summary statistics and classified using all the remaining simulations. Then, if the summary statistics contain sufficient information to discriminate among models, one expects that a large posterior probability should be assigned to the model that generated the pseudo-observed summary statistics. Two versions of the cross-validation are implemented. The first version is a 'hard' model classification. We consider a given simulation as the pseudo-observed data and assign it to the model for which `"postpr"` gives the highest posterior model probability. This procedure is repeated for a given number of simulations for each model. The results are summarised in a so-called *confusion matrix* (Hastie *et al.* 2009). Each row of the confusion matrix represents the number of simulations under a given model, while each column represents the number of simulations assigned by `"postpr"`. If all simulations had been correctly classified, only the diagonal elements of the matrix would be non-zero. The second version is called 'soft' classification. Here, we do not assign a simulation to the model with the highest posterior probability but average the posterior probabilities over many simulations for a given model. This procedure is again summarised as a matrix, which is similar to the confusion matrix. However, the elements of the matrix do not give model counts, but the average posterior probabilities across simulations for a given model. The matrices can be visualised with a bar plot using the `"plot"` S3 method for `"cv4postpr"` (see Fig. 1c).

Conclusions

We provide an R package 'abc' to perform model selection and parameter estimation via ABC. Integrating 'abc' within the R statistical environment offers high-quality graphics and data visualisation tools. The R package implements recently developed non-linear methods for ABC and is going to evolve as new algorithms and methods accumulate. We further direct our users to the package's vignette that contains a detailed worked-through example of an ABC analysis for inferring ancestral human population size based on DNA sequence data.

Acknowledgements

We thank Mark Beaumont for kindly providing an R script that we used in the implementation of the functions abc and postpr. While working on this package, KC was funded by a post-doctoral fellowship from the Université Joseph Fourier (ABC MSTIC) at the Computational and Mathematical Biology Team (BCM, TIMC-IMAG) and then was hosted and financed by the Ecology and Evolution Laboratory (ENS, Paris, ANR-06-BDIV-003).

References

Anderson, C.N.K., Ramakrishnan, U., Chan, Y.L. & Hadly, E.A. (2005) Serial SimCoal: a population genetics model for data from multiple populations and points in time. *Bioinformatics*, **21**, 1733–1734.

Beaumont, M.A. (2008) Joint determination of topology, divergence time, and immigration in population trees. *Simulation, Genetics and Human Prehistory, McDonald Institute Monographs* (eds Matsumura, S. Forster P. & Renfrew, C.), pp. 134–1541. McDonald Institute Monographs, UK.

Beaumont, M.A. (2010) Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, **41**, 379–406.

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

Beaumont, M.A., Marin J.M., Cornuet J.M. & Robert C.P. (2009) Adaptivity for ABC algorithms: the ABC-PMC scheme. *Biometrika*, **96**, 983–990.

Bertorelle, G., Benazzo, A. & Mona, S. (2010) ABC as a flexible framework to estimate demography over space and time: some cons, many pros. *Molecular Ecology*, **19**, 2609–2625.

Blum, M.G.B. (2010) Approximate Bayesian computation: a nonparametric perspective. *Journal of the American Statistical Association*, **105**, 1178–1187.

Blum, M.G.B. & François, O. (2010) Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, **20**, 63–73.

Bortot, P., Coles, S.G. & Sisson, S.A. (2007) Inference for stereological extremes. *Journal of the American Statistical Association*, **102**, 84–92.

Bray, T.C., Sousa, V., Parreira, B., Bruford, M. & & Chikhi, L. (2010) 2BAD: an application to estimate the parental contributions during two independent admixture events. *Molecular Ecology Resources*, **10**, 538–541.

Cornuet, J.M., Santos, F., Beaumont, M.A., Robert, C.P., Marin, J.M., Balding, D.J., Guillemaud, T. & Estoup, A. (2008) Inferring population history with DIY ABC: a user-friendly approach to approximate Bayesian computation. *Bioinformatics*, **24**, 2713–2719.

Cornuet, J.M., Ravigne, V. & Estoup, A. (2010) Inference on population history and model checking using DNA sequence and microsatellite data with the software DIYABC (v1.0). *BMC Bioinformatics*, **11**, 401. ISSN 1471-2105.

Csilléry, K., Blum, M.G.B., Gaggiotti, O.E. & François, O. (2010) Approximate Bayesian computation in practice. *Trends in Ecology & Evolution*, **25**, 410–418.

Excoffier, L. & Foll, M. (2011) Fastsimcoal: a continuous-time coalescent simulator of genomic diversity under arbitrarily complex evolutionary scenarios. *Bioinformatics*, **27**, 1332.

François, O. & Laval, G. (2011) Deviance information criteria for model selection in approximate Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, **10**, 33.

François, O., Blum, M.G.B., Jakobsson, M. & Rosenberg, N.A. (2008) Demographic history of European populations of *Arabidopsis thaliana. PLoS Genetics*, **4**, 1e1000075.

Gelman, A., Carlin, J.B., Stern, H.S. & Rubin, D.B. (2003) *Bayesian Data Analysis, Second Edition (Texts in Statistical Science)*, 2nd edn. Chapman & Hall/CRC, Boca Raton.

Hastie, T., Tibshirani, R. & Friedman, J.H. (2009) *The Elements of Statistical Learning*, 2nd edn. Springer, Berlin.

Hickerson, M.J., Stahl, E. & Takebayashi, N. (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation. *BMC Bioinformatics*, **8**, 268. ISSN 1471-2105.

Huang, W., Takebayashi, N., Qi, Y. & Hickerson, M.J. (2011) MTML-msBayes: approximate Bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC Bioinformatics*, **12**, 1. ISSN 1471-2105.

Hudson, R.R. (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, **18**, 337–338.

Jobin, M.J. & Mountain, J.L. (2008) REJECTOR: software for population history inference from genetic data via a rejection algorithm. *Bioinformatics*, **24**, 2936–2937.

Liepe, J., Barnes, C., Cule, E., Erguler, K., Kirk, P., Toni, T. & Stumpf M.P. (2010) ABC-SysBio—approximate Bayesian computation in Python with GPU support. *Bioinformatics*, **26**, 1797–1799.

Lopes, J.S., Balding, D. & Beaumont, M.A. (2009) PopABC: a program to infer historical demographic parameters. *Bioinformatics*, **25**, 2747–2749.

Marjoram, P., Molitor, J., Plagnol, V. & Tavaré, S. (2003) Markov Chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, **100**, 15324–15328.

Pavlidis, P., Laurent, S. & Stephan, W. (2010) msABC: a modification of Hudson's ms to facilitate multi-locus ABC analysis. *Molecular Ecology Resources*, **10**, 723–727.

Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A. & Feldman, M.W. (1999) Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, **16**, 1791–1798.

R Development Core Team (2011) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Sisson, S.A., Fan, Y. & Tanaka, M.M. (2007) Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, **104**, 1760–1765. Errata (2009), 106, 16889.

Sousa, V.C., Fritz, M., Beaumont, M.A. & Chikhi, L. (2009) Approximate Bayesian computation without summary statistics: the case of admixture. *Genetics*, **181**, 1507.

Tallmon, D.A., Koyuk, A., Luikart, G. & Beaumont, M.A. (2008) COMPUTER PROGRAMS: onesamp: a program to estimate effective population size using approximate Bayesian computation. *Molecular Ecology Resources*, **8**, 299–301. ISSN 1755-0998.

Tavaré, S., Balding, D.J., Griffiths, R.C. & Donnelly, P. (1997) Inferring coalescence times from DNA sequence data. *Genetics*, **145**, 505–518.

Thornton, K.R. (2009) Automating approximate Bayesian computation by local linear regression. *BMC Genetics*, **10**, 35. ISSN 1471–2156.

Wegmann, D., Leuenberger, C., Neuenschwander, S. & Excoffier, L. (2010) ABCtoolbox: a versatile toolkit for approximate Bayesian computations. *BMC Bioinformatics*, **11**, 116. ISSN 1471–2105.

Received 4 September 2011; accepted 22 November 2011
Handling Editor: Robert Freckleton