MOLECULAR ECOLOGY
RESOURCES
Molecular Ecology Resources (2017) **17**, 110–119 doi: 10.1111/1755-0998.12627

SPECIAL ISSUE: POPULATION GENOMICS WITH R

The summary-likelihood method and its implementation in the `Infusion` package

FRANÇOIS ROUSSET,*† ALEXANDRE GOUY,*‡ CAMILLE MARTINEZ-ALMOYNA* and ALEXANDRE COURTIOL§¶

*CNRS, IRD, EPHE, CC065, Institut des Sciences de l'Évolution, University of Montpellier, Pl. E. Bataillon, 34095 Montpellier, France, †Institut de Biologie Computationnelle, University of Montpellier, CC05019, 860 rue St Priest, 34095 Montpellier, France, ‡Institute of Ecology and Evolution, University of Berne, Baltzerstrasse 6, CH-3012 Berne, Switzerland, §Leibniz Institute for Zoo and Wildlife Research, 10315 Berlin, Germany, ¶Berlin Center for Genomics in Biodiversity Research (BeGenDiv), 14195 Berlin, Germany

Abstract

In recent years, simulation methods such as approximate Bayesian computation have extensively been used to infer parameters of population genetic models where the likelihood is intractable. We describe an alternative approach, summary likelihood, that provides a likelihood-based analysis of the information retained in the summary statistics whose distribution is simulated. We provide an automated implementation as a standard R package, `Infusion`, and we test the method, in particular for a scenario of inference of population-size change from genetic data. We show that the method provides confidence intervals with controlled coverage independently of a prior distribution on parameters, in contrast to approximate Bayesian computation. We expect the method to be applicable for at least six-parameter models and discuss possible modifications for higher-dimensional inference problems.

Keywords: approximate Bayesian computation, demographic history, likelihood inference, simulation

Received 28 July 2016; revision received 3 October 2016; accepted 11 October 2016

Introduction

In recent years, simulation-based methods have been developed to estimate parameters of natural processes for situations in which the computation of the likelihood is intractable. By far, the most widely used approach in population genetics is approximate Bayesian computation (ABC) which, given a prior density of parameters of the process considered, uses simulation to construct an estimate of the posterior density of parameters from observed summary statistics (e.g. Beaumont *et al*. 2002; Marjoram & Tavaré 2006; Beaumont 2010). Point estimates and credible intervals for the parameters can be derived from the posterior distribution using standard techniques.

In this work, we develop an alternative to ABC methods and software, to deal with the same broad class of situations where likelihood is intractable, whether in population genetics or not. The method considered here is called summary likelihood, which makes clear that it is not full-data likelihood but is still a form of likelihood-based inference, which one can perform if the full data have been summarized and thrown away and only the summary statistics are available. It provides 'maximum summary-likelihood' estimates of parameters, and 'summary-likelihood'-based confidence intervals defined analogously to the estimates and confidence intervals based on full-data likelihood. Thus, its performance can be assessed in terms of the probability that the intervals include the parameter values (coverage).

It is expected from theory (e.g. Cox & Hinkley 1974; Casella & Berger 2002), but less well recognized in practice, that the credible intervals produced by ABC have no simple relationship with confidence intervals. Credible intervals may occasionally provide reasonable approximations for confidence intervals, but are not specifically adapted for that purpose. The different intervals (and more generally, regions for several parameters) are, or can be, defined in terms of their coverage under different sampling schemes. Credible intervals can be defined as intervals with known coverage (e.g. the conventional 95% coverage) over a prior distribution of parameters, for given data. Credible intervals defined in this way then also have known average coverage jointly under the given prior distribution for the parameters

Correspondence: François Rousset, Fax: (+33) 467143622; E-mail: francois.rousset@umontpellier.fr

and under repeated sampling of data. In contrast, confidence intervals should have known coverage for any given parameter value and thus for any prior distribution on parameters. The latter properties are usually not considered in ABC studies. One exception can be found in Peter *et al.* (2010, Appendix S3), who considered problems of inference of past population history, and found in simulations that credible intervals had higher coverage over parts of the parameter space, and lower coverage over other parts, so that only average coverage was controlled.

Summary likelihood follows essentially the same idea as discussed by Diggle & Gratton (1984) in the first dedicated discussion of generic approaches for simulation-based inference. Although similar ideas are recurrently considered (e.g. Rubio & Johansen 2013; Bertl *et al.* 2015), we are unaware of any generic implementation, which would further have demonstrated performance in terms of coverage. One of the factors that may have inhibited the spread of such methods may be the limited reliability and/or the computer requirements of available smoothing techniques, on which a generic and automated software implementation could be based.

We have implemented summary likelihood in the R package `Infusion`, which is available on the Comprehensive R Archive Network, and which name stands for `Inf`erence `us`ing simulat`ion`. The current implementation handles models with less than nine parameters and is based on two main sets of techniques: the modelling of empirical distributions of summary statistics using mixtures of Gaussian distribution, as implemented in the `Rmixmod` package (Lebret *et al.* 2015), and the inference of likelihood surfaces from estimates of the likelihood of given parameter points, using methods implemented in the `spaMM` and `blackbox` packages (Rousset & Ferdy 2014; Rousset 2016). Previous attempts have considered kernel smoothing methods that can also be called by the `Infusion` procedures, but available implementations have apparent constraints (in particular, in terms of number of variables handled), which make them insufficient for implementation of summary likelihood.

In the following, we first introduce the summary-likelihood method and its implementation through a toy example based on the Gaussian distribution. Next, we will discuss how the package provides access to several methods for reducing the number of summary statistics, an important functionality not illustrated by this first example. To compare the method with ABC, we will reconsider the population-size change scenario of Peter *et al.* (2010), in which case we can demonstrate better coverage of the confidence intervals than achieved in that work. Finally, we will discuss possible modifications of the current implementation, for example to deal with higher-dimensional parameter space.

Methods

Toy example with most informative statistics

We consider here the estimation of the mean μ and variance σ^2 of a Gaussian distribution from the sample mean $\tilde{\mu}$ and bias-corrected sample variance $\tilde{\sigma}^2$ of 40 observations drawn from a Gaussian distribution. Simulation-based inference is obviously not necessary for such estimation, but it is easily visualized, and easily comparable to alternative methods. As the statistics $\tilde{\mu}$ and $\tilde{\sigma}^2$ each contain all information about each of the parameters of the Gaussian distribution, we expect the results to be practically equivalent to standard likelihood-based inference.

In this example, as in any application of the method, users must provide either tables of simulated summary statistics, or more conveniently a simulation function that can be called by the package's functions. Here, this function returns a vector of summary statistics $\tilde{\mu}$ and $\tilde{\sigma}^2$, for given parameter values. In R code, this function may be written

```
myrnorm <-function(mu, s2, sample.size) {
 s <- rnorm(n=sample.size, mean=mu, sd=sqrt(s2))
 return(c(mean=mean(s), var=var(s)))
}
```

For purposes of illustration, we produce a realization that will stand for the actual data to be analysed, for the parameter values $\mu = 4$ and $\sigma = 1$ to be estimated:

```
set.seed(123) ## initialize the random generator
Sobs <- myrnorm(mu=4, s2=1, sample.size=40)
```

The obtained `Sobs` has elements `mean` $= \tilde{\mu} = 4.045$ and `var` $= \tilde{\sigma}^2 = 0.806$. The maximum-likelihood estimate of the mean is $\tilde{\mu}$, and the maximum-likelihood estimate of the variance is $\hat{\sigma}^2 \equiv \tilde{\sigma}^2(n-1)/n = 0.786$. Further, the Student's t-based exact 95% confidence interval for the mean μ is [3.76, 4.33]. It uses the exact conditional distribution of t given the sample variance. Alternatively, the more generally available χ^2 approximation for the distribution of the (profile) log-likelihood ratio may be used to construct approximate likelihood-based confidence intervals (or more generally, confidence regions). The likelihood of the data can be written in terms of the sample mean and variance (see, e.g. Davison 2003, p. 66). For the mean, the resulting profile likelihood-based confidence interval is [3.77, 4.32]. Likewise, confidence intervals for the variance can be constructed using the exact distribution of the sample variance (yielding the interval [0.541, 1.33]) or the profile likelihood-based approximation (yielding the interval [0.522, 1.26]). The approximate

confidence intervals appear similar to the exact intervals, except for the upper bound of the variance.

Our goal in this example is to recover the likelihood-based confidence intervals by summary likelihood, ignoring the specific results that provide the above intervals by analytical means. For that purpose, we will explore the parameter space to evaluate the probability density of the statistics for different parameter ($\boldsymbol{\theta} = (\mu, \sigma^2)$) values. This exploration is iterative: a first step provides first estimates; then, new parameters points are chosen, for which new simulations are performed, and refined estimates are produced. At each step, a log-likelihood surface is inferred from the estimated log-likelihoods in different parameter points.

Estimating a surface from points each estimated with some error is a classical problem in machine learning or 'computer experiments' (e.g. Bingham *et al.* 2014) and requires some form of smoothing. A standard approach to this problem (e.g. Sacks *et al.* 1989; Welch *et al.* 1992) is to use variants of Kriging, that is of prediction under a linear mixed model with autocorrelated random effects. This will be reliable only if the smoothing parameters (that is, the variance and autocorrelation parameters of the random effects) are all estimated. The first set of parameter points is therefore defined so as to facilitate estimation of smoothing parameters: we sample the parameters of the simulation function along an irregular grid (a regular grid is inappropriate to estimate autocorrelation parameters; Zimmerman 2006) including some replicated points (suitable for estimating the residual variance of the mixed model, which is essential for good smoothing). A convenience function `init_grid` is available to perform this initial sampling given an initial range of parameters:

```
library(Infusion)
parsp <- init_grid(lower=c(mu=2.8,s2=0.4,
sample.size=40),
       upper=c(mu=5.2,s2=2.4,sample.size=40))
```

We use the `add_simulation` function from the package to build a list of simulated distributions for this set of $\boldsymbol{\theta}$ values.

```
simuls <- add_simulation(Simulate="myrnorm",
par.grid=parsp)
```

For each parameter point $\boldsymbol{\theta}$, `add_simulation` has simulated an empirical distribution of summary statistics (by default, of 1000 realizations) by directly calling the `myrnorm` function given as argument `Simulate`. In more involved applications, the simulation code may not be callable from `R`, but this case is also handled by `add_simulation`, which can accept as input a new data frame of simulated summary statistics for given parameter values.

We then estimate the probability density of the observed summary statistics `Sobs` for each simulated distribution, using the `infer_logLs` function:

```
densv <- infer_logLs(simuls,stat.obs=Sobs)
```

`infer_logLs` performs a smoothing of the empirical distribution of summary statistics for given $\boldsymbol{\theta}$, using by default functions from the `Rmixmod` package to fit a mixture model of Gaussian distributions to each empirical distribution (Fig. 1a). `infer_logLs` uses Akaike's information criterion, justified as a measure of predictive accuracy (Akaike 1974), to select among mixture models with different numbers of Gaussian components. It then infers the log-likelihood of the given $\boldsymbol{\theta}$ as the log-density of the observed summary statistics in the fitted Gaussian mixture model (Fig. 1b).

We can then estimate a summary log-likelihood surface by smoothing all estimates of log-likelihoods of parameters obtained in this way. This is performed by calling,

```
slik <- infer_surface(densv)
```

where `slik` is an object of class `SLik` (pronounced 'sleek'). It contains information about practically all computations previously carried out, except the simulated distributions.

Parameter inference can then be performed as if the summary log-likelihood surface was a full-data log-likelihood surface. In particular, we can obtain from it the 'maximum summary-likelihood' (MSL) estimate as well as confidence intervals, using the `MSL` function:

```
slik2 <- MSL(slik) ## adds estimates and intervals
to the object
```

One can also visualize the results by, for example `plot(slik2, filled=TRUE)`, which in the present case shows both the bounds of confidence intervals and the two-dimensional confidence region (Fig. 1c). The first estimates are inaccurate, but can be improved iteratively (Fig. 1d) as described below.

Given the data, the parameter estimates and intervals have some random error as the probability densities on which they are based are themselves estimated with some error. A feature contributing to the performance is that linear mixed model theory provides estimates of prediction uncertainty of the summary log-likelihood surface, specifically a covariance matrix of the predictions of the log-likelihood in given points. From this covariance matrix, the `MSL` function computes the prediction variance of the log-likelihood ratio at the current confidence bounds. This can be used to determine whether more simulations should be run to reduce uncertainty.

Another feature contributing to the performance of the summary-likelihood method is the use of appropriate

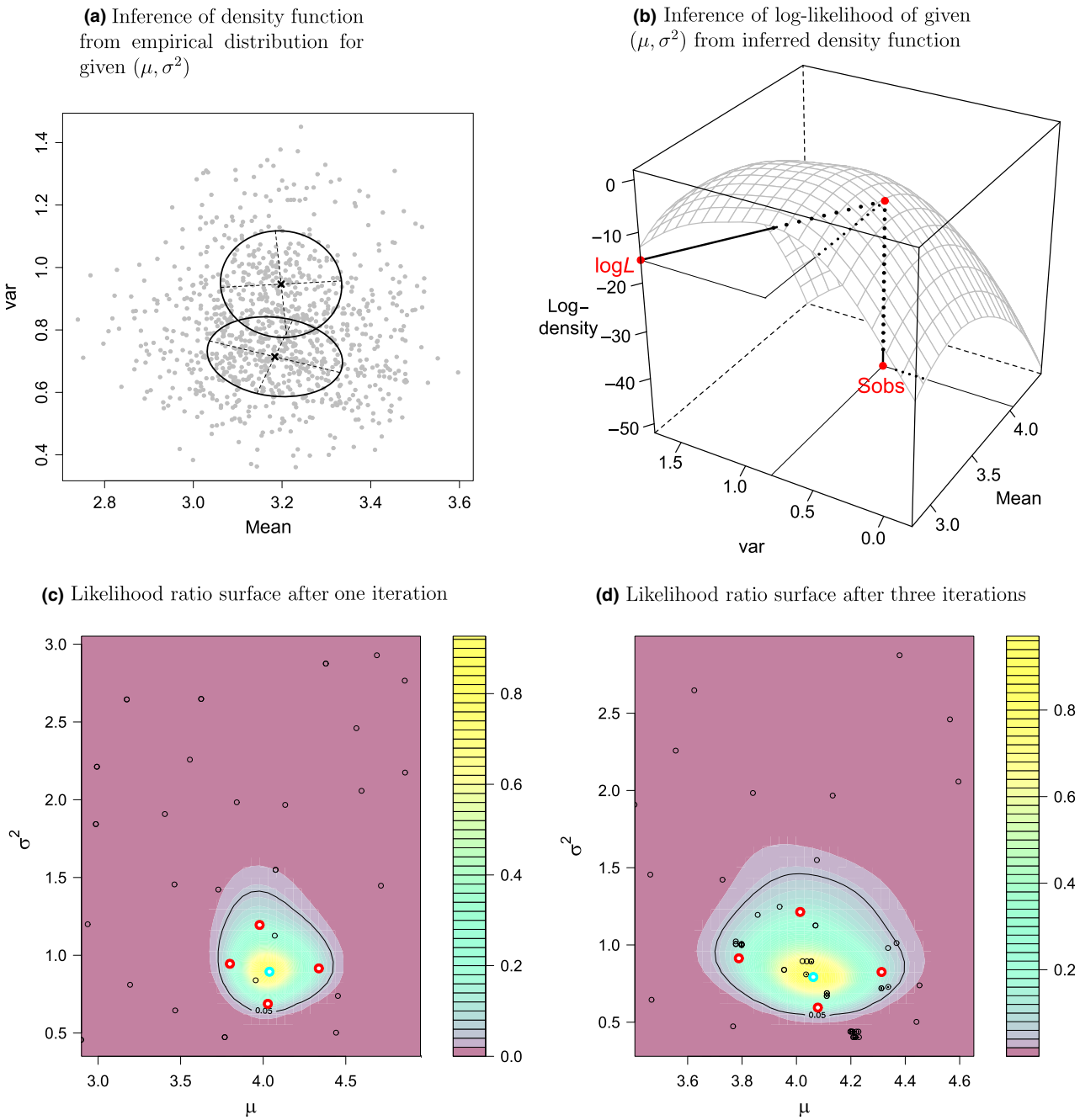

Fig. 1 Steps of the inference of the likelihood surface. The plots in the top row show the inference of the log-likelihood, here for parameters ($\mu = 3.19$, $\sigma^2 = 0.81$), and for data summarized by the statistics ($\tilde{\mu} = 4.045, \tilde{\sigma}^2 = 0.806$). In the present example, the empirical distribution is fitted by a mixture of two Gaussian distributions, with location and covariance matrices represented by ellipses depicted in plot (a). The likelihood of the parameters is then given by the inferred density of the observed summary statistics in this Gaussian mixture model (plot b). The plots in the bottom row show the profile likelihood ratio surfaces inferred from the inferred log-likelihoods of different parameter points. The scale is that of the likelihood ratio relative to the maximum. The blue and red circles mark, respectively, the estimated maximum-likelihood point and the confidence intervals points, that is, the out-most points on the contour defined by the profile likelihood threshold for the profile confidence intervals. There is a pair of CI points for each interval. Also shown is the inferred contour of the 95% confidence region for both parameters.

methods to estimate the smoothing parameters for inference of the log-likelihood surface. In `Infusion`, this estimation is performed automatically, by default using restricted likelihood (REML) estimation of random effect parameters. There is no need for the user to guess any smoothing parameter.

We then use 'expected improvement' methods (e.g. Bingham *et al.* 2014) to determine the new parameter points for which new empirical distributions of summary statistics will be simulated. The aim of expected improvement methods is to account for the fact that parameter regions that have yet been little sampled typically have high prediction variance, and may thus be worth sampling even if their predicted likelihood is relatively low in such regions. This again makes use of measures of prediction uncertainty, not specifically for the current estimates of summary log-likelihood in current estimates of the target points, but for any candidate point in parameter space. This approach is used here specifically to identify more accurately the maximum-likelihood estimates, but also the confidence limits. In the latter case, confidence limits (λ_-, λ_+) for any parameter λ are deduced from the profile log-likelihood ratio defined by maximization over other parameters. Then, expected improvement is used to select new values of these other parameters given $\lambda = \lambda_-$ or $\lambda = \lambda_+$.

In the current implementation, it is by default assumed that simulating the distribution of statistics is more costly than other computation steps; hence, only a few new parameter points are defined in each iteration. Here, there are five target values (the maximum and two interval bounds for each of two parameters), and on average, three points are defined for each of these targets, with more or fewer points depending on the relative prediction uncertainty of log-likelihood for the current estimates of the targets. In addition, distributions of summary statistics are computed for a few parameter points taken from the previous iterations. These replicates are computed to improve smoothing (as in the first iteration). In total, about twenty empirical distributions are added in each iteration for this two-parameter model.

An important feature of the iterative approach is that it is not very important to have accurate estimation of likelihood in each parameter point, because the accumulation of likelihood estimates near a target point over successive iterations will provide, by the infill asymptotic properties of Kriging (Stein 1999), an accurate estimation of log-likelihood at the target point.

According to such principles, we can therefore refine estimates iteratively, using the refine function

```
slik3 <- refine(slik2) ## performs new simulations
and updates the object
```

The results after two successive `refine` calls, which added simulations for 38 new likelihood values, are shown in Fig. 1d. The point estimates are $\hat{\mu} = 4.047$, $\hat{\sigma}^2 = 0.794$, with interval [3.775, 4.303] for μ and [0.548, 1.245] for σ^2, clearly approaching the analytical likelihood-based intervals.

Projecting summary statistics

The previous example did not address all difficulties of simulation-based inference, as we started from statistics known to contain all information about the parameters. In practice, one often has to deal with less appropriate statistics.

Further, these statistics are often in excess of the number of parameters. Although the above functions can deal with this case, it may be necessary to reduce the number of statistics. Such a reduction is useful because it will reduce the computation time of the smoothing of empirical distribution of summary statistics, which might otherwise become prohibitive as the number of statistics increases. The need to reduce the number of summary statistics is also discussed in the ABC literature, where neural networks (Blum & François 2010), boosting procedures based on weighting of different predictors (Aeschbacher *et al.* 2012), random forests (Pudlo *et al.* 2016) and simple linear regression (Fearnhead & Prangle 2012) have been used. Kriging can also be used for this step but may be much slower. We refer to all relevant methods as 'projection methods'.

The package provides a `project` function that acts as an interface for various projection methods. One must call once (before the `infer_logLs` call) the `project` function for a parameter given as its first argument, for example

```
mufit <- project("mu", stats=c("mean", "var"),
data=simuls)
```

This call constructs a projector function (here called `mufit`) that will compute from the given statistics (`stats` argument) a single summary statistic for the given parameter (here, the mean). This function is a predictor of the given parameter, constructed from the input data (simulations of original summary statistics for known parameter values). Similarly, we create a projector function `s2fit` that will compute a summary statistic for the variance:

```
s2fit <- project("s2", stats=c("mean", "var"),
data=simuls)
```

Next, one applies all defined projectors both on the observed summary statistics and on the initial simulations. For example, given two projectors `mufit` and `s2fit`, we construct projected values `projSobs` of the

summary statistics and `projSimuls` of the simulation table by running

```
projSobs <- project(Sobs, projectors=list
(MEAN=mufit, VAR=s2fit))
projSimuls <- project(simuls, projectors=list
(MEAN=mufit, VAR=s2fit))
```

This calls again `project`, but now with statistics rather than parameter names as first argument, thereby calling a distinct method of a generic `project` function. The `projectors` argument determines the names of the new summary statistics, which are here `MEAN` and `VAR`. The project function is not further explicitly called by the user in the sequel of the analysis, as the projector functions have been previously defined and are called automatically in subsequent calls to `refine`, provided `projSobs` and `projSimuls` are provided as arguments to the initial `infer_logLs` call. The documentation available on the package web page, and the R script for the population-size change model (see Supporting Information), provides complete examples using several projection methods.

Summary of main procedures in the `Infusion` *package*

The above example has introduced the main functions and the work flow of a typical analysis, successively using `init_grid` (optional, but safe), `add_simulation`, `project` (optional), `infer_logLs`, `infer_surface`, `MSL` (all called once) and `refine` (which can then be called repeatedly). Other convenient functions include a predict method that returns the log-likelihood for a given complete vector $\boldsymbol{\theta}$ of parameter values; a `profile` method that gives the value of the profile log-likelihood at the given set of parameter values (i.e. any subset of values of the parameter vector $\boldsymbol{\theta}$); a `confint` method that gives a confidence interval for a given parameter value (any element of $\boldsymbol{\theta}$); and functions `plot1Dprof` and `plot2Dprof`, which plot one- and two-dimensional likelihood ratio profiles. When called at each iteration, these functions allow the user to monitor the progress and in particular to check that a good initial parameter range has been chosen.

Assessing the validity of the confidence intervals

Here, we will discuss the performance of summary likelihood in an evolutionary demographic situation, the population-size change (PSC) model. We chose this scenario because it has already been considered in an ABC application, where some assessment of coverage was provided as function of parameter values (Peter *et al.* 2010).

Simulation model and summary statistics

The PSC model allows for exponential growth or decline at a constant rate starting from the current population with size N_0, and, going backwards in time, to vary exponentially over t generations to a population of size N_t. In this model, the population size at a given generation i is computed as follows: $N_i = N_0 a^{i/t}$, with $a = N_0/N_t$.

The empirical distributions of summary statistics were simulated for parameters drawn from log-uniform distributions in the following ranges for the three model parameters: $N_0 \in [100, 50\,000]$, $a \in [10^{-3}, 10^{3}]$ and $t \in [1, 10^{3}]$. We simulated 200 unlinked microsatellite loci for samples of size $n = 25$ diploids individuals. Microsatellite data were generated under a generalized stepwise model (GSM) with a mutation rate $\mu = 5.10^{-4}$. As in Peter *et al.* (2010), we estimated a and the scaled parameters $\theta = N_0\mu$ and $\tau = t/N_0$ of the PSC model.

We used the software `IBDsim` (Leblois *et al.* 2009) to make the coalescent-based simulations by calling its command-line version from R. We computed the following six summary statistics on the simulated data sets: the F_{IS}, Garza & Williamson's (2001) M, the number of alleles K, the heterozygosity H and the standard deviations of H and K over loci, as in the original study. Neural networks were used to generate three projected summary statistics from these six ones.

Coverage property assessment

As summary likelihood is an approximate method, we checked the convergence of the estimation of confidence intervals by computing the coverage probability along a range of parameters under the PSC model.

We simulated data sets for known parameters and then checked whether we were able to correctly estimate them. We did this for 10 sets of parameters, where we simulated 200 data sets under the PSC model for each set. We then estimated the 90% and 95% confidence intervals for each of the three parameters (θ, τ and a) with the summary-likelihood method. To do so, we inferred parameters using `Infusion`'s default procedures, based on an initial set of 110 empirical distributions (of which 10 are replicates). We refined the summary-likelihood surface by adding three iterations for each inference. We then calculated the proportion of the true parameters contained in these intervals to estimate the coverage probabilities for 90 and 95% confidence intervals.

Unlike what Peter *et al.* (2010) found using ABC, we observe that the coverage probability appears fairly constant over all the parameter ranges with our method, in particular when the true value lies close to the range limits (Fig. 2). Indeed, all our estimates fit with theoretical

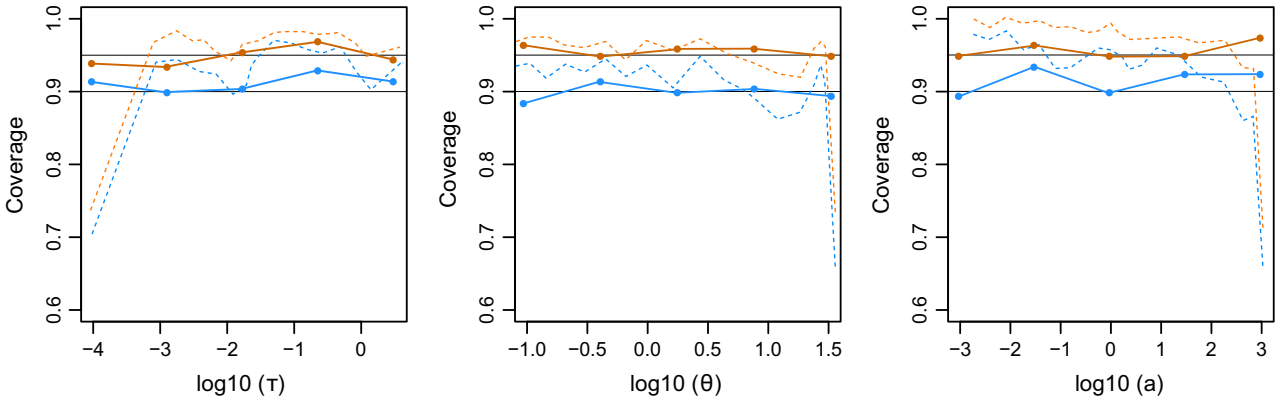

Fig. 2 Coverage properties under the PSC model. For each parameter τ, θ and a, coverage probabilities computed along their respective ranges for 95% (orange) and 90% confidence intervals (blue) are represented. The points joined by solid lines are the coverage probabilities of the intervals computed using summary likelihood. The coloured dashed lines represent the coverage probabilities of the intervals computed using ABC under the same model, obtained from Peter et al. (2010, Fig. S3). The black lines mark the theoretical expectations for 95% and 90% confidence intervals coverage.

expectations. The 2.5% and 97.5% quantiles of the distribution of the expected proportion of successes for 200 binomial trials with probability 0.95 are 0.92 and 0.98, and 0.855 and 0.940 for binomial trials with probability 0.90. The average coverages for the parameters τ, θ and a are, respectively, 0.949, 0.957, 0.958 for 95% confidence intervals, and 0.913, 0.900, 0.916 for 90% intervals. As in the original study, the number of replicates is limited, not only by the computation time of each replicate, but also by issues with the full automation of the work flow involving the external simulation program (here, IBDsim). This prevented a finer evaluation of coverages in both the original and the present study.

The `Infusion` package has also been tested in the conditions of the toy example. A total of 1000 Gaussian data sets were simulated for $\boldsymbol{\theta}_0 = (\mu = 4, \sigma^2 = 1)$, and analysed as shown in Methods, with a total of 11 iterations (see R script in Supporting Information). 96% of the 95% confidence regions for (μ, σ^2) contained the true value. The full distribution of the summary-likelihood ratio P-values for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ was indistinguishable from a uniform distribution (Kolmogorov-Smirnov test, $P = 0.4701$), indicating an appropriate coverage irrespective of the threshold used to define confidence regions. When the amount of simulation is reduced down to four iterations and only 100 realizations are drawn for each simulated empirical distribution, the distribution of P-values is barely affected (Fig. 3), although the variance of estimation of the likelihood ratio is increased.

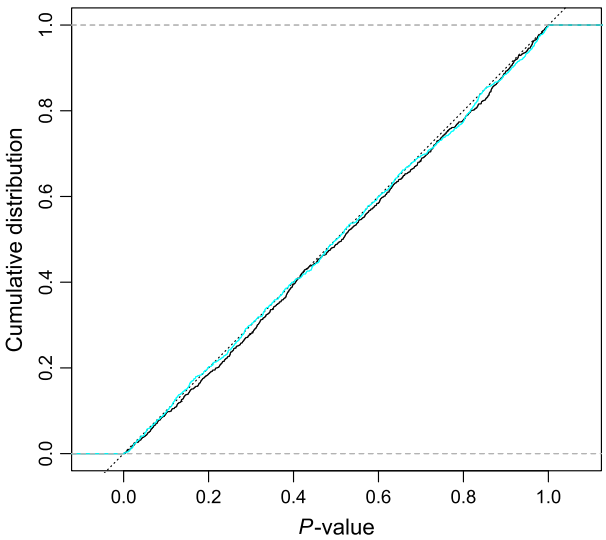

Fig. 3 Distribution of P-values in the Gaussian model. The empirical distribution of P-values for 1000 simulated data sets is shown for either 11 iterations with 1000 realizations for each simulated empirical distribution of statistics (black curve), or four iterations with only 100 realizations (cyan curve).

Discussion

In this work, we have developed methods and implemented them in the package `Infusion`, to construct estimates of likelihood surfaces for summary statistics, from which likelihood ratio confidence intervals can be constructed. We show that this provides intervals with better controlled coverage than previous comparable methods. The methodology is implemented in the standard R package `Infusion`. Users have to provide initial ranges for the parameters, a simulation function or simulated distributions of summary statistics, and must choose some projection method(s) if too many summary statistics are given in input, but the procedures implemented in `Infusion` are otherwise fully automated.

Simulated distributions may be provided by any program, not depending on R.

We have emphasized one measure of performance: the control of coverage of intervals for any parameter value, according to which confidence intervals are defined. This rests on the premise that such a criterion is useful for evaluating statistical methods of inference (e.g. Cox 2006, Appendix B). ABC methods may not have been conceived to achieve such control. Yet, this per se gives no reason for not applying the criterion to ABC. The validity of a criterion derives only from its ability to measure the consequences of applying an inference method, not from the way the method was conceived.

Given that the practical performance of any generic simulation-based inference method may degrade with increasing number of parameters, we have first considered low-dimensional problems. We have not yet tested our methods for large numbers of parameters, where we expect the likelihood surface inference step (the smoothing of summary-likelihood estimates) to be a substantial computational bottleneck. The projection step should be comparatively less problematic in high-dimensional cases: machine-learning methods such as neural networks and random forests have been designed to handle quickly scores of input variables. The procedures used in the smoothing step and for the sampling of new parameter points have been applied, in the context of full-data coalescent-based likelihood surface inference, to five-parameter (Rousset F, Beeravolu CR, Leblois R submitted) or even six-parameter (unpublished) models. These procedures are here used in a more economical way, requesting fewer parameter points, so they should be practical for up to at least six parameters. A harder constraint may be set by the `geometry` package, which is used for the sampling of new parameter points from a previously inferred summary-likelihood surface, but which is not expected to handle practically more than eight variables (Habel *et al.* 2015), in which case the sampling procedure should be redefined.

With a large enough amount of simulation, the estimated summary likelihood of a given parameter $\boldsymbol{\theta}$ should become identical to the true summary-likelihood value, and the performance of inferences based on estimated likelihood ratios should become equivalent to that of exact likelihood ratio-based inference of the summary statistics. There are many ways in which the implementation can be modified and possibly improved to reduce the amount of computation required to achieve a given precision in estimation of likelihoods, or to fit models with more parameters.

Small-sample corrections to likelihood ratio statistics may also be needed. Indeed, likelihood ratio-based intervals may not be accurate. The textbook argument for using likelihood-based confidence intervals is based on the asymptotic chi-square distribution of the likelihood ratio for large samples, under the assumption that the log-likelihood can be represented as a sum of n independent and typically identically distributed random variables, generally corresponding to the log-likelihood of observations from n individuals (e.g. Severini 2000). But, in genetic applications, in particular, a sample of n genes is typically not considered as resulting from n independent draws. Instead, the n genes are related through their common ancestry, and the realized ancestral genealogy can be viewed as a single draw of a latent variable. The impact of this dependence is clear for example on the full-likelihood inference of the mutation rate under the infinite allele model (IAM), where the variance of the maximum-likelihood estimator decreases asymptotically as $1/\log(n)$ (Tavaré 1994, p. 41), rather than as $1/n$ as is usual for independent draws. Yet, even in this case, full-data likelihood-based intervals achieve practically perfect coverage from small one-locus samples (Rousset *et al.* submitted). We therefore anticipate that summary-likelihood ratios will also be appropriate for most genetic applications, although in some cases, they may still need some form of small-sample correction. For example, a Bartlett correction (e.g. Severini 2000) could be implemented, requiring little or no additional simulation effort.

In later developments, the following changes will likely be considered. First, the simulation input could be more similar to that of ABC, that is, consisting in realizations of summary statistics, one for each of different parameters points $\boldsymbol{\theta}$ sampled from an instrumental prior distribution. Users might then be concerned with the choice of the prior. However, given an appropriate processing of the simulation results, this choice should not affect the actual coverage of the confidence intervals more than the initial sampling of parameter points in the present implementation, and a uniform prior may then be appropriate. This form of input may be useful from three perspectives. First, it would make it easier for one to switch from ABC to summary likelihood. Second, simulating one realization of the summary statistics for each of different parameters points could also, at least in some cases, allow inference from fewer simulations than required by the current implementation. Third, it could allow the recycling of techniques previously developed for inference of posterior densities, for the inference of likelihood surfaces. This could be useful in particular to deal with higher-dimensional parameter spaces.

Second, several projections methods have been considered in the literature and are accessible through the package functions, but these are little general guidance on which to choose. This problem is shared with ABC.

Preliminary investigations (not shown) suggest that neural networks give reasonable results, but also that different projection methods may be appropriate for different types of simulation input. In particular, random forests may not give good results for training data taken from our initial simulation table with many replicates of a few parameter values. Further work is clearly needed.

It is not easy to compare computation times with those of ABC, as the two methods should converge to different results for increasing computation times. Such comparisons would make sense only for similar accuracy of the results (here, for similar control of coverage of confidence intervals). This is not easy to define, since for given computation time, markedly different degrees of accuracy may be reached in different regions of the parameter space. Thus, we make no claim that our method is uniformly faster than ABC to reach a certain accuracy, and very crude results may even be easier to reach with current ABC software. However, we obtained reasonable accuracy with only four iterations in the population-size change model, and also with four iterations in the Gaussian model even though the number of realizations for each simulated empirical distribution was reduced to 100. In such low-dimensional problems, the surface inference is performed in a few seconds at most, and thus, the main computational bottleneck may be the simulation step. In our analysis of the population-size change model, fewer simulations were required than in the comparable ABC analysis by Peter *et al*. (2010). A limited amount of simulation thus appears sufficient at least to identify a parameter region of interest, on which more refined simulations could be conducted.

In conclusion, we have implemented summary likelihood, a method of inference based on the estimation of likelihood of parameters from the simulated distribution of summary statistics. We have shown that it is applicable and performs according to theoretical expectations, providing intervals with better controlled coverage than a previous ABC method applied to the same problem. Further work may reduce the amount of simulation needed to reach the same precision of inference.

Acknowledgements

We thank editors Armando Geraldes and Emmanuel Paradis for inviting and handling this contribution, the reviewers for useful comments and Jean-Michel Marin for many discussions about machine-learning methods. A. Gouy was funded by the French ANR program (project 661 "SilentAdapt" BIOADAPT 2013-2015). Simulations were performed on the cluster of the Institut des Sciences de l'Évolution.

References

Aeschbacher S, Beaumont MA, Futschik A (2012) A novel approach for choosing summary statistics in approximate Bayesian computation. *Genetics*, **192**, 1027–1047.

Akaike H (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716–723.

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

Beaumont MA, Zhang W, Balding DJ (2002) Approximate Bayesian computation in population genetics. *Genetics*, **162**, 2025–2035.

Bertl J, Ewing G, Kosiol C, Futschik A (2015) *Approximate Maximum Likelihood Estimation*. ArXiv:1507.045533 [stat.CO].

Bingham D, Ranjan P, Welch WJ (2014) Design of computer experiments for optimization, estimation of function contours, and related objectives. In: *Statistics in Action: A Canadian Outlook*(ed Lawless J. F.), pp. 109–124. Chapman and Hall/CRC, New York.

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

Casella G, Berger RL (2002) *Statistical Inference*. Duxbury, Pacific Grove, CA.

Cox DR (2006) *Principles of Statistical Inference*. Cambridge University Press, Cambridge, UK.

Cox DR, Hinkley DV (1974) *Theoretical Statistics*. Chapman & Hall, London, UK.

Davison AC (2003) *Statistical Models*. Cambridge University Press, New York.

Diggle PJ, Gratton RJ (1984) Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society B*, **46**, 193–227.

Fearnhead P, Prangle D (2012) Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation (with discussion). *Journal of the Royal Statistical Society B*, **74**, 419–474.

Garza J, Williamson E (2001) Detection of reduction in population size using data from microsatellite loci. *Molecular Ecology*, **10**, 305–318.

Habel K, Grasman R, Gramacy RB, Stahel A, Sterratt DC (2015) GEOMETRY: *Mesh Generation and Surface Tesselation*. R package version 0.3-6.

Leblois R, Estoup A, Rousset F (2009) IBDSIM: a computer program to simulate genotypic data under isolation by distance. *Molecular Ecology Resources*, **9**, 107–109.

Lebret R, Iovleff S, Langrognet F, Biernacki C, Celeux G, Govaert G (2015) RMIXMOD: the R package of the model-based unsupervised, supervised, and semi-supervised classification mixmod library. *Journal of Statistical Software*, **67**, 1–29.

Marjoram P, Tavaré S (2006) Modern computational approaches for analysing molecular genetic variation data. *Nature Reviews Genetics*, **7**, 759–770.

Peter BM, Wegmann D, Excoffier L (2010) Distinguishing between population bottleneck and population subdivision by a Bayesian model choice procedure. *Molecular Ecology*, **19**, 4648–4660.

Pudlo P, Marin JM, Estoup A, Cornuet JM, Gautier M, Robert CP (2016) Reliable ABC model choice via random forests. *Bioinformatics*, **32**, 859–866.

Rousset F (2016) BLACKBOX: black box optimization and exploration of parameter space. R package version 1.0.8.

Rousset F, Ferdy JB (2014) Testing environmental and genetic effects in the presence of spatial autocorrelation. *Ecography*, **37**, 781–790.

Rousset F, Beeravolu CR, Leblois R (submitted) Likelihood analysis of population genetic data under coalescent models: computational and inferential aspects. *J. Société Française Statistique*.

Rubio FJ, Johansen AM (2013) A simple approach to maximum intractable likelihood estimation. *Electron. J. Stat.*, **7**, 1632–1654.

Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Statistical Science*, **4**, 409–435.

Severini TA (2000) *Likelihood Methods in Statistics*. Oxford University Press, Oxford.

Stein ML (1999) *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York, NY.

Tavaré S (1994) Ancestral inference in population genetics. In: *Lecture Notes in Mathematics 1837*(ed Picard J.), pp. 1–188. Springer-Verlag, Berlin.

Welch WJ, Buck RJ, Sachs J, Wynn HP, Mitchell TJ, Morris MD (1992) Screening, prediction, and computer experiments. *Technometrics*, **34**, 15–25.

Zimmerman DL (2006) Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, **17**, 635–652.

F.R. conceived the project and wrote the package. A.G., C.M.-A. and A.C. tested several versions of the package. A.G. performed simulations for the population-size change model. F.R., A.G. and A.C. wrote the manuscript.

Data accessibility

`Infusion` package: available on the Comprehensive R Archive Network (CRAN.R-project.org/mirrors.html); additional R scripts: online Supporting information; simulations for first iteration of the PSC analyses: online Supporting information.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Appendix S1 R script for inference under PSC model. This script allows both to perform simulations using the other script (Appendix S2) and the software IBDsim, and to perform inferences from the simulations. The simulation set from the first iteration is also provided (Appendix S3).

Appendix S2 R script for simulation under PSC model.

Appendix S3 Simulations for first iteration of PSC analyses (R data file).