Using Approximate Bayesian Computation (ABC) for parameter inference in SimInf

OH-EJP FULL_FORCE WP5-T5.2, deliverable D-JRP19-WP5.D3
Stefan Widgren, SVA, Sweden

Introduction

The objectives of OH-EJP FULL_FORCE WP5 (modelling) are to address: i) gaps in quantitative knowledge on the spread of AMR which will be essential to direct future focused research, ii) insight in the uncertainty around the effect of measures reducing AMR prevalence in the food production chains, and iii) identification of key elements in the production chains to mitigate the risk of human exposure. Estimating model parameters from observed data is a major challenge in stochastic modelling. However, it is a necessary and critical step to evaluate a model's explanatory power. Parameterization is preferably conducted within a Bayesian framework, and in this document, we will focus on Approximate Bayesian computation (ABC) (Toni et al., 2009), a simulation-based computational approach for parameterisation of complex problems, for example, in epidemiology (McKinley et el., 2018).

Overview of SimInf

The **SimInf** R package provides an efficient and very flexible framework to conduct data-driven epidemiological modelling in realistic large-scale disease spread simulations (Widgren et al., 2019). The framework integrates infection dynamics in subpopulations as continuous-time Markov chains using the Gillespie stochastic simulation algorithm and incorporates available data such as births, deaths, and movements as scheduled events at predefined time-points. Using C code for the numerical solvers and 'OpenMP' (if available) to divide work over multiple processors ensures high performance when simulating a sample outcome. The package contains template models and can be extended with user-defined models. The **SimInf** software is open source and licensed under the GNU General Public License version 3 (https://opensource.org/licenses/GPL-3.0). The most recent stable version of **SimInf** is available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/package=SimInf. The development version is available on GitHub at https://github.com/stewid/SimInf.

Approximate Bayesian computation

Here, we will only briefly introduce ABC. For an in-depth description and tutorial of ABC, see, for example, Toni et al, (2009), Sisson et al. (2018) or Turner et al. (2012). ABC methods were developed for inferring the posterior distribution of parameter values where the likelihood functions are to computationally intractable or too costly to evaluate. Instead, a comparison of the distance between data simulated from a model and observed data is used, and the assumption is that parameter values which leads to small distances approximates the posterior distribution. ABC rejection sampling is the the most basic algorithm, it samples and evaluates proposals randomly from the prior. A more efficient approach is to consider ABC with sequential Monte Carlo sampling (ABC-SMC) (Toni et a., 2009), where the proposals are resampled from the accepted proposals and the distance for accepting proposals gradually decreas.

There exist several R packages at CRAN that implements ABC functionality. For example, the **abc** (Csilléry et al., 2012) package implements several ABC algorithms for performing parameter estimation, model selection, and goodness-of-fit. It also contains cross-validation tools for measuring the accuracy of ABC estimates, and to calculate the misclassification probabilities of different models. The **abcrf** (Marin et al., 2022) package performs ABC model choice and parameter inference via random forests.

Since ABC is computationally challenging, the efficiency of **SimInf** for generating data is valuable. Although, other R packages could be used for the actual ABC calculations, implementing ABC-SMC in **SimInf** allowed for using the internal data structures efficiently and adapt the simulations depending on the underlying model specification. Moreover, to facilitate usage of ABC, adaptive tolerance selection (Simola et al., 2021) was added. The recently implemented ABC-SMC functionality in **SimInf** has been used to estimate parameter values for modelling, for example, transmission of ESBL-producing Escherichia coli in Dutch broiler production chain (Furusawa, 2022), environmentally mediated spread of livestock-associated methicillin-resistant *Staphylococcus aureus* in a pig herd (Tuominen et al., 2022), and disease transmission on a cattle movement network (Bronstein et al., 2022).

Basic example of compartment model construction and ABC analysis in SimInf

One of the design goals of **SimInf** was to make it extendable to facilitate construction of various epidemiological compartment models. The easiest way to create a new compartment model for **SimInf** is to use the `mparse` method, the built-in model parser functionality (Widgren et al., 2019). The `mparse` method takes text strings representing transitions in the form of "`X → propensity → Y`" and machine generates the required code for the model. The left-hand side of the first "→"-sign is the initial state, for example, susceptible individuals `S`. The right-hand side of the last "→"-sign is the state they transition to, for example, infectious individuals `I`. The transition rate for the state change is written between the "→"-signs.

To illustrate the ABC-SMC functionality in **SimInf**, we will consider an SIR model in a closed population, i.e., no births or deaths. Please note the text with a grey background below indicates code that has been entered in an R session. After installing the package

```
R> install.packages("SimInf")
```

it is loaded in R with the following command

```
R> library("SimInf")
```

Now, let `beta` denote the transmission rate, `gamma` the recovery rate, `u0` the initial population consisting of 100 susceptible individuals, one infected and no recovered. Additionally, the duration `tspan` to simulate over and at which time-points to record the population status must be specified. In this case, we let `tspan = 1, 4, 7, …, 100`. Consult the help page for other `mparse()` parameter options. The model can be described as,

```
R> model <- mparse(transitions = c("S -> beta*S*I/(S+I+R) -> I",
+                                  "I -> gamma*I -> R"),
+                  compartments = c("S", "I", "R"),
+                  ldata = data.frame(beta = 0.16, gamma = 0.077),
+                  u0 = data.frame(S = 100, I = 1, R = 0),
+                  tspan = seq(from = 1, to = 100, by = 3))
```

We are now ready to use the SIR model to create an outbreak with observed data (Figure 1), from which we will infer the, in this case, the known `beta` and `gamma` parameters. In a real outbreak, these parameters are not known, but in this example, we generate synthetic data to illustrate using the ABC methodology in **SimInf**. For reproducibility, we first call the `set.seed()` function.

```
R> set.seed(22)
R> infected <- trajectory(run(model), "I")[, c("time", "I")]
R> colnames(infected) <- c("time", "Iobs")
```

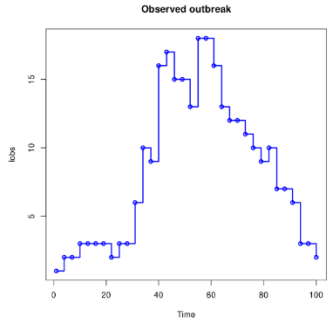

Figure 1: The number of infected at the time-points 1, 3, …, 100, in the simulated outbreak.

To perform ABC-SMC, the distance function to accept or reject a proposal must be defined. This function estimates the distance between the observed and simulated data. Each node in the in the simulated trajectory (contained in the 'result' object) represents one proposal.

```
R> distance <- function(result, ...) {
+     ## Extract the time-series of infectious in each node.
+     sim <- trajectory(result, "I")
+
+     ## Split the 'sim' data by node and calculate the mean
+     ## squared error (MSE) at each time-point for each node.
+     dist <- tapply(sim$I, sim$node, function(sim_infectious) {
+         mean((infected$Iobs - sim_infectious)^2)
+     })
+
+     ## Return the distance for each node. Each proposal will
+     ## be accepted or rejected depending on if the distance
+     ## is less than the tolerance for the current generation.
+     return(dist)
```

```
+   }
```

Now, fit the model parameters using ABC-SMC and adaptive tolerance selection. The priors for the `beta` and `gamma` parameters are specified using a formula notation. Here we use a uniform distribution for each parameter with lower bound equal to zero and upper bound equal to one. Furthermore, let `npart=1000`, for the number of particles, i.e., the number of parameter values to accept in each generation. In the first generation, sample `ninit=10000` particles from the prior distribution. Finally, use the distance function previously defined. Consult the help page for other `abc()` parameter options.

```
R> fit <- abc(model = model,
+             priors = c(beta ~ uniform(0, 1),
+                        gamma ~ uniform(0, 1)),
+             npart = 1000,
+             ninit = 10000,
+             distance = distance)
```

The return value from `abc()` is a `'SimInf_abc'` object with data from the parameter inference attached to it. The show some basic information about the inference, such as the number of particles, number of generations, and the extremes, the mean, and the quartiles of the parameter inference.

```
R> fit
Number of particles per generation: 1000
Number of generations: 6

Generation 6:
-------------
 Accrate: 4.91e-02
 ESS: 6.75e+02

         Min. 1st Qu. Median   Mean 3rd Qu.   Max.
 beta  0.0361  0.1323 0.1706 0.1717  0.2081 0.3664
 gamma 0.0243  0.0711 0.0858 0.0869  0.0999 0.1625
```

In Figure 2, the proposed and accepted parameter values are displayed for each generation. As observed, the parameter space shrinks per generation such that the accepted parameter values becomes more concentrated around `beta=0.16` and `gamma=0.077`, the values that were used to generate the observed outbreak.

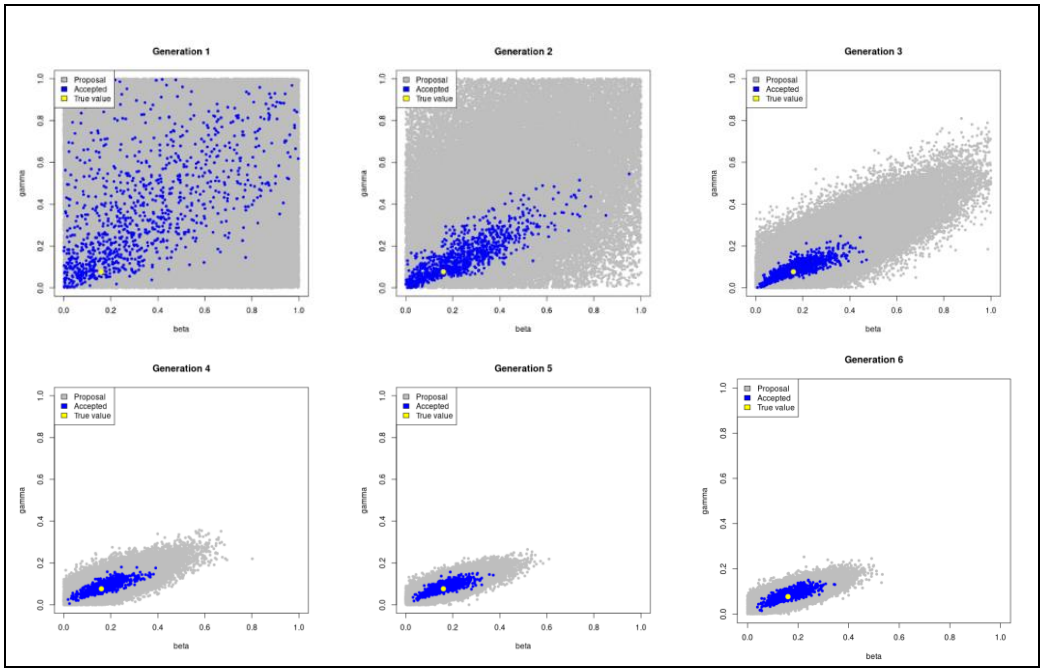

Figure 2: Illustration of each generation of candidates for the parameters from the proposal distribution (grey) and the accepted values for the parameters (blue). The known parameter value (`beta=0.16` and `gamma=0.077`) for this example is indicated in yellow. In each generation, the tolerance for the distance between the observed and simulated data for accepting a proposed parameter value is adaptively reduced.

Conclusion

In this report we have demonstrated the newly added ABC-SMC functionality for parameter inference in the **SimInf** R package. We hope that this new functionality will facilitate epidemiological research to better understand disease transmission and improve design of intervention strategies for endemic and emerging threats. Future development of **SimInf** will focus on adding functionality for other inference algorithms that are useful for infectious disease modelling and parameter inference, such as the maximum likelihood via iterated filtering (MIF) approach (Ionides et al., 2006), and particle Markov chain Monte Carlo (PMCMC) (Andrieu et al., 2010).

Acknowledgements

This work has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 773830: One Health European Joint Programme.

References

Andrieu C, Doucet A, Holenstein R. Particle markov chain monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2010 Jun;72(3):269-342.

Bronstein S, Engblom S, Marin R. Bayesian inference in Epidemics: linear noise analysis. arXiv preprint arXiv:2203.10906. 2022 Mar 21.

Csilléry K, François O, Blum MG. abc: an R package for approximate Bayesian computation (ABC). Methods in ecology and evolution. 2012 Jun;3(3):475-9. https://doi.org/10.1111/j.2041-210X.2011.00179.x

Furusawa M. Transmission models of ESBL-producing Escherichia coli in Dutch broiler production chain. Master's Thesis, Utrecht University. 2022. https://studenttheses.uu.nl/handle/20.500.12932/42869

Ionides EL, Bretó C, King AA. Inference for nonlinear dynamical systems. Proceedings of the National Academy of Sciences. 2006 Dec 5;103(49):18438-43.

Marin J, Raynal L, Pudlo P, Robert CP, Estoup A (2022). abcrf: Approximate Bayesian Computation via Random Forests. R package version 1.9, https://CRAN.R-project.org/package=abcrf

McKinley TJ, Vernon I, Andrianakis I, McCreesh N, Oakley JE, Nsubuga RN, Goldstein M, White RG. Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. Statistical science. 2018 Feb;33(1):4-18. https://doi.org/10.1214/17-STS618

Simola U, Cisewski-Kehe J, Gutmann MU, Corander J. Adaptive approximate Bayesian computation tolerance selection. Bayesian analysis. 2021 Jun;16(2):397-423. https://doi.org/10.1214/20-BA1211

Sisson SA, Fan Y, Beaumont M, editors. Handbook of approximate Bayesian computation. CRC Press; 2018 Sep 3.

Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MP. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. Journal of the Royal Society Interface. 2009 Feb 6;6(31):187-202. https://doi.org/10.1098/rsif.2008.0172

Tuominen, K. S., Lewerin, S. S., Jacobson, M., & Rosendal, T. (2022). Modelling environmentally mediated spread of livestock-associated methicillin-resistant Staphylococcus aureus in a pig herd. Animal, 16(2), 100450. https://doi.org/10.1016/j.animal.2021.100450

Turner BM, Van Zandt T. A tutorial on approximate Bayesian computation. Journal of Mathematical Psychology. 2012 Apr 1;56(2):69-85. https://doi.org/10.1016/j.jmp.2012.02.005

Widgren, S., Bauer, P., Eriksson, R., & Engblom, S. (2019). SimInf: An R Package for Data-Driven Stochastic Disease Spread Simulations. Journal of Statistical Software, 91(12), 1–42. https://doi.org/10.18637/jss.v091.i12