Trajectory-matching ABC-MCMC for simulating heterogeneous dynamics in mechanistic models

Fatemeh Beigmohammadi[1,2], Jordan J.A. Weaver[3], Solène Hegarty-Cremer[1,2], Cailan Jeynes-Smith[3], Amber M. Smith[3], Morgan Craig[1,2,*]

[1]Sainte-Justine University Hospital Research Centre, Montréal, Québec, Canada
[2]Department of Mathematics and Statistics, Université de Montréal, Montréal, Québec, Canada
[3]Department of Pediatrics, University of Tennessee Health Science Center, Memphis, TN USA

*Corresponding author: Morgan Craig, morgan.craig@umontreal.ca

Abstract: The inherent heterogeneity of complex biological systems makes it difficult to experimentally and clinically explore individual outcomes within them. Mechanistic mathematical models are essential tools for studying such heterogeneity. Thus, there is increasing interest in integrating newer mechanistic model-based techniques, like virtual patient cohorts and virtual clinical trials, within experimental and regulatory pipelines to probe relationships that may be difficult or impossible to ascertain through traditional wet-lab or clinical experimentation alone. Computational approaches like Approximate Bayesian Computation (ABC) are attractive methods for generating virtual patient cohorts and running virtual clinical trials. Among the various ABC approaches, ABC combined with Markov chain Monte Carlo (ABC-MCMC) is widely used to improve sampling efficiency and guarantee convergence to the approximate posterior. However, ABC-MCMC must meet the acceptance criteria of both approaches, which results in a high rejection rate. In response, we developed a model-based technique called trajectory-matching ABC-MCMC (TM-ABC-MCMC). TM-ABC-MCMC captures the variability of complex biological systems by constraining model trajectories between the upper and lower bounds of available data to generate heterogeneity in model parameters. By testing the method's performance on existing mechanistic models and comparing to existing ABC-MCMC algorithms, we show that TM-ABC-MCMC accurately reproduces the observed noise in biological systems of varying complexity, all while maintaining computational efficiency. Thus, TM-ABC-MCMC is a new approach for generating heterogeneity in mechanistic mathematical models with implications for model-based experimental design, virtual patient cohorts, and virtual clinical trials.

Keywords: virtual patient cohorts; trajectory matching; approximate Bayesian computation; Markov chain Monte-Carlo; mechanistic mathematical models

1. Introduction

One of the main challenges of mathematically modelling disease progression and treatment lies in capturing the diversity and complexity of biological systems. Numerous factors, including age, sex, comorbidities, and immune status, contribute to variability in dynamics and responses. Experimental studies and clinical trials are often limited by time, cost, and ethical considerations. Thus, exploring the full range of plausible scenarios through data collection alone is generally infeasible. Mechanistic mathematical models are a powerful way to describe biological heterogeneity and are, therefore, essential tools to explore variability that would be impractical to assess experimentally[1-6].

Several existing approaches generate realistic heterogeneity in mechanistic models using global optimization routines, such as adaptive simulated annealing, genetic algorithms, or Metropolis-Hastings[7-9]. Among them, Approximate Bayesian Computation (ABC) is a widely used rejection-based method for estimating posterior parameter distributions in simulation-based models[10-16]. ABC-Markov Chain Monte Carlo (MCMC), a common variant, samples one candidate at a time using MCMC, but is largely inefficient due to its high rejection rate and computational cost[17-23]. Other variants improve efficiency by proposing multiple candidates in parallel, though they may still struggle with high-dimensional models[11,12,22,24-27]. In general, methods incorporating ABC rely on approximations (i.e., summary statistics) and assumptions (i.e., prior choice, distance metrics, and tolerance thresholds) that can compromise accuracy[25]. To overcome these challenges, we developed an algorithm based on ABC-MCMC designed to enhance both robustness and inference quality without sacrificing computational efficiency.

Here, we introduce an improved version of ABC-MCMC that is easily applicable to systems with many parameters, a common feature of mechanistic models. Our method combines the advantages of ABC-MCMC by simulating model outputs and comparing them to experimental data (ABC step) using parameters drawn from normal distributions, whose means are updated at each iteration (MCMC step). In our approach, the standard deviations of these normal distributions are typically determined based on previously estimated best-

fit values. Sample acceptance occurs when model simulated trajectories lie within the upper and lower bounds of the observed experimental data, a criterion we term trajectory matching. We show that TM-ABC-MCMC reduces rejection rates, improves sampling efficiency, and enhances inference robustness. Thus, TM-ABC-MCMC effectively captures complex, dynamic behaviors in disease models and provides insight into biological uncertainty and response variability. These factors are helpful to improve the integration of mechanistic modelling within experimental designs, model diseases using virtual patient cohorts, and establish effective treatments using virtual clinical trials[3,28-32].

2. Methods

2.1. TM-ABC-MCMC pseudocode

A flow chart of the TM-ABC-MCMC algorithm is given in **Figure 1**. To initialize the algorithm, let $\mu_i = \theta_i^*$ for the first iteration (i.e., when $n = 1,\ 1 \leq n \leq N$). θ_i^* is then updated at each subsequent iteration based on newly accepted parameter sets as follows.

Step 1. For $2 \leq n \leq N$ and $1 \leq m \leq M$, sample parameters from normal distributions:

$$\theta_1^{nm} \sim \mathcal{N}(\theta_1^*, {\sigma_1}^2), \ldots, \theta_g^{nm} \sim \mathcal{N}(\theta_g^*, {\sigma_g}^2).$$

Step 2. Simulate Y from the model $\mathcal{M}$ given candidate parameters $\{\theta_1^{nm}, \ldots, \theta_g^{nm}\}$.

Step 3a. Assess the suitability of predicted outputs Y with respect to the data at each time point s. This is performed by checking whether the predicted model trajectory falls within the bounds defined above in Eq. (2) or Eq. (3). If the new parameter set meets these acceptance criteria, i.e., Eqs. (2) or (3) are satisfied for a pre-defined number of time points q $(q < T)$, accept Y, and update the mean in each normal distribution for the next iteration by

$$\theta_1^* = \theta_1^{nm}, \ldots, \theta_g^* = \theta_g^{nm},$$

to define a new set of parameters $\Theta^{nm} = \{\theta_1^{nm}, .., \theta_g^{nm}\}$.

Step 3b. Otherwise, continue the MCMC sampling using the previous mean:

$$\Theta^* = \{\theta_1^*, .., \theta_g^*\}.$$

Repeat **Steps 1-3** for N iterations in each of the M parallel runs.

Step 4. Apply convergence tests (**Figure 2**) to select M^*(first row in **Figure 2**), which is the optimal value of M that yields the most stable and reliable outcomes. For M^*, estimate the posterior distributions of the parameters $\{P(\theta_1), P(\theta_2), \dots, P(\theta_g)\}$ using the accepted samples, after removing duplicates that resulted from rejections during the process.

2.2. Computational implementation

All simulations and analyses were performed in Julia (version 1.10). The implementation uses several open-source packages, including *DelayDiffEq* and Sundials for solving differential equations, *Distributions*, *StatsBase*, and *KernelDensity* for statistical routines, *PlotlyJS* for interactive visualizations, and functions DataFrames, CSV, and XLSX for data management. Additional functions *Interpolations*, *QuadGK*, and *Parameters* were used for numerical integration and parameter handling.

2.3. Case Study 1: Lotka Volterra model

The Lotka-Volterra (LV) model is a deterministic system that describes the interaction between prey (x) and predator (y) populations that is commonly used throughout biology. Here, we used the LV model given by:

$$\frac{dx}{dt} = ax - xy, \qquad (6a)$$

$$\frac{dy}{dt} = bxy - y. \qquad (6b)$$

Parameters a and b describe the growth rate of the prey, x, in the absence of predators, y, and the rate of predator reproduction due to interaction with prey, respectively [24,33,34]. Toni et al. applied ABC and its variants to the LV model to compare their efficiency based on acceptance rates. For this, they generated synthetic data for both species x and y (D_x and D_y) by adding Gaussian noise ($\mathcal{N}(0, (0.5)^2)$) to simulated trajectories at eight randomly selected time points. Because Toni et al. reported similar performance between ABC and ABC-MCMC, we limited our benchmarking comparisons to ABC and TM-ABC-MCMC. For this, we repeated the ABC rejection procedure following the approach described in Toni et

al. When a system exhibits well-defined extrema (e.g., stable oscillations), synthetic data bounds can be constructed by applying a fixed percentage of uncertainty (e.g., 20%) around those points. Thus, to compare TM-ABC-MCMC to ABC, we selected synthetic data points based on the extrema of the model's prediction for one of the components (here, x) to generate trajectories and parameter distributions. Using the parameters as in Toni et al., the component x has two minima and three maxima over fifteen days. Therefore, we defined bounds at five time points according to:

$$x_i^{upper} = \max\left(x_i^{max} + 0.2x_i^{max}, x_i^{max} - 0.2x_i^{max}, x_i^{min} + 0.2x_i^{min}, x_i^{min} - 0.2x_i^{min}\right) \quad (7.a)$$

$$x_i^{lower} = \min\left(x_i^{max} + 0.2x_i^{max}, x_i^{max} - 0.2x_i^{max}, x_i^{min} + 0.2x_i^{min}, x_i^{min} - 0.2x_i^{min}\right) \quad (7.b)$$

Let the set of these points be $D_{extrema}$. Parameters a and b were then sampled from normal distributions with means $\mu_a = \mu_b = 1$ and variances ${\sigma_a}^2 = {\sigma_b}^2 = 0.1$. To accurately compare TM-ABC-MCMC to ABC, in which a distance metric is used as acceptance criteria, the same distance function (sum of squared errors, SSE) used by Toni et al. was applied to the TM-ABC-MCMC results. Only those accepted parameter sets for which the SSE was less than or equal to the ABC acceptance threshold were retained ($\epsilon = 4.3$ as defined in Toni et al.[24] (Table S1)).

While extrema are often informative points in a model's dynamics, other dynamical features, including waning rates and plateaus, may be more salient properties for a given system. In such cases, time points must be selected strategically. See the Supplementary Information for a discussion of some strategies for generating the synthetic data and their influence on TM-ABC-MCMC performance, including how varying the level of agreement between data and model affects the resulting trajectories and parameter distributions (Table S8, Figure S12, S13). The importance of time point selection is discussed in Supplementary Table S5 and Supplementary Figure S9.

2.4. Case Study 2: Models of influenza virus infection

The second model to which we applied TM-ABC-MCMC is a system of ordinary differential equations (ODEs) describing the viral kinetics of influenza A infection[35,36]. The model

includes four variables: susceptible epithelial cells (T), eclipse infected cells (I_1), productively infected cells (I_2), and virus (V):

$$\frac{dT}{dt} = -\beta TV. \qquad (8a)$$

$$\frac{dI_1}{dt} = \beta TV - kI_1, \qquad (8b)$$

$$\frac{dI_2}{dt} = kI_1 - \delta(I_2)I_2, \qquad (8c)$$

$$\frac{dV}{dt} = pI_2 - cV, \qquad (8d)$$

where $\delta(I_2) = \frac{\delta_d}{K_\delta + I_2}$. Parameters were fit to viral titers measured from the lungs of infected mice on 12 consecutive days using adaptive simulated annealing (ASA). Smith et al.[37] reported the best fit values and 95% CIs of each parameter (Table S2). Note that the value of k, the rate of transition through the eclipse phase, must be restricted between 4 and 6 to ensure biological relevance. This model was then extended by Myers et al. to study CD8⁺ T cell immune responses by adding equations for effector (E) and memory (E_M) CD8⁺ T cells[38]:

$$\frac{dE}{dt} = \xi(E)I_2 + \eta E I_2(t - \tau_E) - d_E E, \qquad (9a)$$

$$\frac{dE_M}{dt} = \zeta(t - \tau_M), \qquad (9b)$$

where $\zeta(E) = \frac{\zeta}{K_E + E}$, and the T cell clearance of infected cells in Eq. ($8c$) is described by $\delta_E(I_2, E) = \frac{\delta_E E}{K_{\delta_E} + I_2}$. The extended version has 14 parameters. Myers et al. similarly obtained best-fit parameter values and their 95% CIs by fitting their model using simulated annealing and bootstrapping viral load and total CD8⁺ T cell count data from days 0 to 12 ($T = 13$); all parameter values were deemed to be well defined[23] (Table S3).

2.5. Case Study 3: Model of tumour growth after combination immunotherapy

Finally, we applied TM-ABC-MCMC to a cancer immunotherapy model describing the tumor response to a combination of an oncolytic virus and injections of dendritic cells (DCs)[39]. The

treatment schedule considered includes injections of 2.5×10^9 oncolytic virions on days 0, 2, and 4, and 10^6 dendritic cells on days 1, 3, and 5[36]. The model tracks uninfected tumor cells (U), tumor cells infected by an injected oncolytic virus (I), oncolytic virus (V), naïve CD8⁺ T cells (A), tumor antigen-specific CD8⁺ T cells (T), and injected dendritic cells (D):

$$\frac{dU}{dt} = rU - \beta \frac{UV}{N} - (k_0 + c_{kill}\, I)\frac{UT}{N}, \qquad (10a)$$

$$\frac{dI}{dt} = \beta \frac{UV}{N} - \delta_I I - (k_0 + c_{kill}\, I)\frac{IT}{N}, \qquad (10b)$$

$$\frac{dV}{dt} = u_v(t) + \alpha\delta_I I - \delta_V V, \qquad (10c)$$

$$\frac{dT}{dt} = c_T I + \chi_A A + \chi_D D - \delta_T T, \qquad (10d)$$

$$\frac{dA}{dt} = c_A I - \delta_A A, \qquad (10e)$$

$$\frac{dD}{dt} = u_D(t) - \delta_D D, \qquad (10f)$$

This model has fourteen parameters, six of which (the net tumor growth rate, r, infection rate β, naïve T cell recruitment rate c_A, T cell recruitment rate c_T, enhanced T cell cytotoxicity rate, c_{kill}, and the rate of T cell simulation by DCs χ_D) were estimated and reported with corresponding 95% credible intervals (CrI; Table S4). In Barish et al.[40], parameters were estimated hierarchically using simulated annealing based on aggregated data. Here, we generated the necessary distributions about the best-fit values of the six parameters with CrIs in a single step using TM-ABC-MCMC, i.e., without employing the hierarchical fitting procedure.

3. Results

3.1. TM-ABC-MCMC algorithm

To reduce the computational costs of ABC-MCMC, we developed trajectory-matching ABC-MCMC. TM-ABC-MCMC is a model-based algorithm inspired by ABC-MCMC, but with key differences in acceptance criteria. As in ABC-MCMC, experimental data and a mathematical model form the backbones of TM-ABC-MCMC. To apply the method, we suppose the

availability of (1) observed data for at least one model component, (2) point estimates of parameters, and (3) confidence intervals (CIs). However, scenarios may arise where one or more of these elements are missing, e.g., when CIs are not reported, parameter estimates have low confidence, or estimating plausible bounds on the data is not possible. In such cases, TM-ABC-MCMC can still be applied, but the performance and accuracy of the algorithm may be affected (see Supplementary Tables S6 and S7).

The TM-ABC-MCMC procedure is as follows. Assume that we have an existing mechanistic model describing a biological system with previously defined biologically plausible bounds on its relevant parameters. Define this mechanistic model, $\mathcal{M}$, as the system of differential equations:

$$\frac{dY}{dt} = f(Y, \Theta, t), \qquad (1)$$

where $Y(t)$ is the set of L model state variables $Y(t) = \{y_1, .., y_L\}$, Θ is the set of g model parameters $\Theta = \{\theta_1, .., \theta_g\}$, and t is time. Simulation of $\mathcal{M}$ yields time-dependent trajectories, $y_i(t)$, that predict the model behavior with parameters Θ.

Let x_k $1 \leq k \leq L$ be the set of experimentally observed outcomes and let T denote the total number of measurements. The first step of TM-ABC-MCMC is to define the response interval given the upper and lower bounds for each x_k. If there are individual-level data available ($x_{k_{ind}}$), we define the response interval to be

$$min(x^s_{k_{ind}}) \leq y^s_k \leq max(x^s_{k_{ind}}), \qquad (2)$$

where $min(x^s_{k_{ind}})$ and $max(x^s_{k_{ind}})$ are the maximum and minimum values, respectively, of each data point at the discrete time point s, for $1 \leq s \leq T$. Alternatively, if the observed data are reported as summary statistics for the population, $x_{k_{pop}}$ represents the population-level data, and the response interval is instead defined as

$$(x_{k_{pop}} - SE)^s \leq y^s_k \leq (x_{k_{pop}} + SE)^s, \qquad (3)$$

i.e., the $mean \pm standard\ error\ (SE)$. TM-ABC-MCMC uses these response intervals as acceptance criteria.

Once response intervals are established, several parameters must be defined to set up the run. The first is the sample size or number of iterations, N. The second is the number of parallel runs (or the number of times the loop initializes from best fit parameters), M. The appropriate values of N and M depend on a given model, but they are generally in the range of $[10^4, 10^6]$ and $[1,5]$, respectively. For models with well-estimated parameters (i.e., parameters estimated with a high degree of confidence), a higher N and lower M are selected, whereas for models in which there is less confidence in the estimated parameter values, a lower N and higher M are used.

For the MCMC step, proposal distributions around parameter means must be defined. These proposal distributions are assumed to be normal, $\mathcal{N}(\mu, \sigma^2)$. We set the distribution means to be the best-fit parameter estimates, $\Theta^* = \{\theta_1^*, .., \theta_g^*\}$. These best-fit values are the set of parameters that best minimize the difference between model predictions and the observed data (see Discussion) and are assumed to be available before running TM-ABC-MCMC. Thus, the resulting normal distribution of parameter θ_i is given by:

$$\theta_i \sim \mathcal{N}(\theta_i^*, {\sigma_i}^2), \quad 1 \leq i \leq g. \qquad (4)$$

where θ_i^* is the best fit value of parameter θ_i, and U_{θ_i} and L_{θ_i} are the upper and lower bounds, respectively, of the estimates on parameter θ_i, which are derived from available confidence intervals. The standard deviation, σ_i^2, is defined according to the difference between these upper and lower bounds:

$${\sigma_i}^2 = \left(U_{\theta_i} - L_{\theta_i}\right) \times C, \qquad (5)$$

where the coefficient C is a value between $[0.05,0.1]$. If U_{θ_i} and $L_{\theta_i} \in [10^{-2}, 10^2)$, the sampling space will be linear. Otherwise, sampling from logarithmic space and logarithmic values of U_{θ_i} and L_{θ_i} is used to calculate ${\sigma_i}^2$. Typical values of σ_i^2 fall between $[0.05, 0.5]$. However, when sampling from log space with ${\sigma_i}^2$ very small, we neglect the coefficient C

and simply define ${\sigma_i}^2$ to be $\left(U_{\theta_i} - L_{\theta_i}\right)$. When confidence intervals (CIs) are not available, the standard deviation can be defined as a constant value $C = 0.1$ across all normal distributions (see Supplementary Information).

TM-ABC-MCMC follows an MCMC-like sampling strategy, where parameters are drawn from a normal distribution (as defined above) with dependence on previously sampled values. While the comparison with experimental data is in TM-ABC-MCMC is inspired by ABC, rather than using a distance metric and tolerance threshold as in ABC-MCMC, our approach only requires that a model-predicted trajectory of a candidate set of parameters remains within U_{θ_i} and L_{θ_i}, i.e., the upper and lower bounds of the observed data, at the corresponding measured time points.

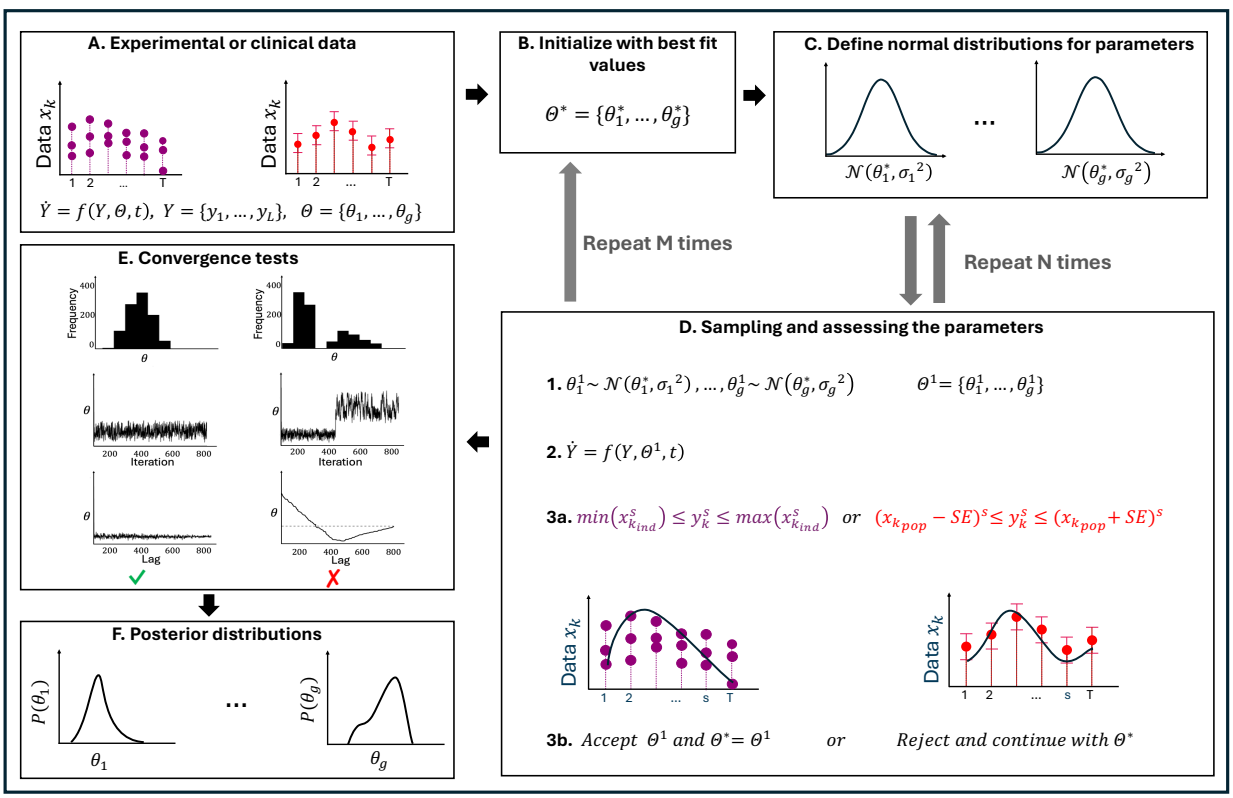

Figure 1. Overview of trajectory matching-ABC-MCMC algorithm. *A) The TM-ABC-MCMC algorithm starts with experimental or clinical data x_k, corresponding mathematical model Y, and previously estimated parameters $\Theta^* = \{\theta_1^*, .., \theta_g^*\}$. B and C) To initialize the algorithm, define the means of normal distributions as $\mu_1 = \theta_1^*, \dots, \mu_g = \theta_g^*$. Then, repeat steps 1 to 3 for N iterations to generate $\Theta^1, \dots, \Theta^N$. D) Update each mean in normal distributions if step 3a is fulfilled for a (pre-determined and) sufficient*

number of time points. Restart the process from $\mu_1 = \theta_1^*, \ldots, \mu_g = \theta_g^*$ for M rounds. E) Finally, use the results of convergence tests to determine the F) posterior distribution of parameters.

3.2. Assessing the convergence of TM-ABC-MCMC

After N iterations, diagnostic plots are used to determine whether the Markov chain has converged[41,42]. As outlined in Step 4 of the pseudocode, we must assess the convergence of the TM-ABC-MCMC algorithm. For this, several visual checks are assessed simultaneously. First, trace plots of the value of each sample at every MCMC iteration are generated. Trace plots (**Figure 2A**) that show constrained fluctuations without sudden jumps (i.e., noisy, "wandering" lines across a stable range) indicate that the algorithm is exploring the parameter space effectively and confirm that the MCMC step is adequately exploring the parameter distribution, rather than simply minimizing error. Next, autocorrelation plots are used to assess how strongly current samples are correlated with earlier ones. A sharp drop in autocorrelation indicates that the samples are nearly independent, whereas high autocorrelation suggests that the drawn samples do not fully represent the posterior distribution (**Figure 2B**). Because MCMC methods are being implemented, samples will be correlated from iteration to iteration. However, the goal here is to ensure that this effect is at its minimum. Lastly, parameter value density plots are used to summarize sampled values and approximate the posterior distribution. The sharper a peak in these histograms, the more informative the data were about that parameter. A flat posterior distribution suggests weak parameter identifiability, while a bimodal distribution indicates that the sampler is oscillating between distinct regions of the parameter space. Both patterns can arise from data limitations or a mismatch between the model and the observed data. In practice, a unimodal posterior distribution is often preferred, as it facilitates efficient sampling and typically reflects well-identified parameters. However, care must be taken to distinguish between genuine multimodality and artifacts arising from model–data mismatch or data limitations. Trace, autocorrelation, and density plots should be considered in combination to assess MCMC convergence. However, convergence tests are not definitive. In cases where parameter estimates exhibit high uncertainty or the underlying distributions are poorly defined, the convergence assessment may also be inconclusive[43]. Reliable and

unreliable diagnostic plots are shown in **Figure 2**. Although such perfect convergence patterns are often achievable in simple models, more complex systems typically yield less ideal yet still acceptable patterns. Importantly, convergence should not resemble the erratic behavior seen in the unreliable examples.

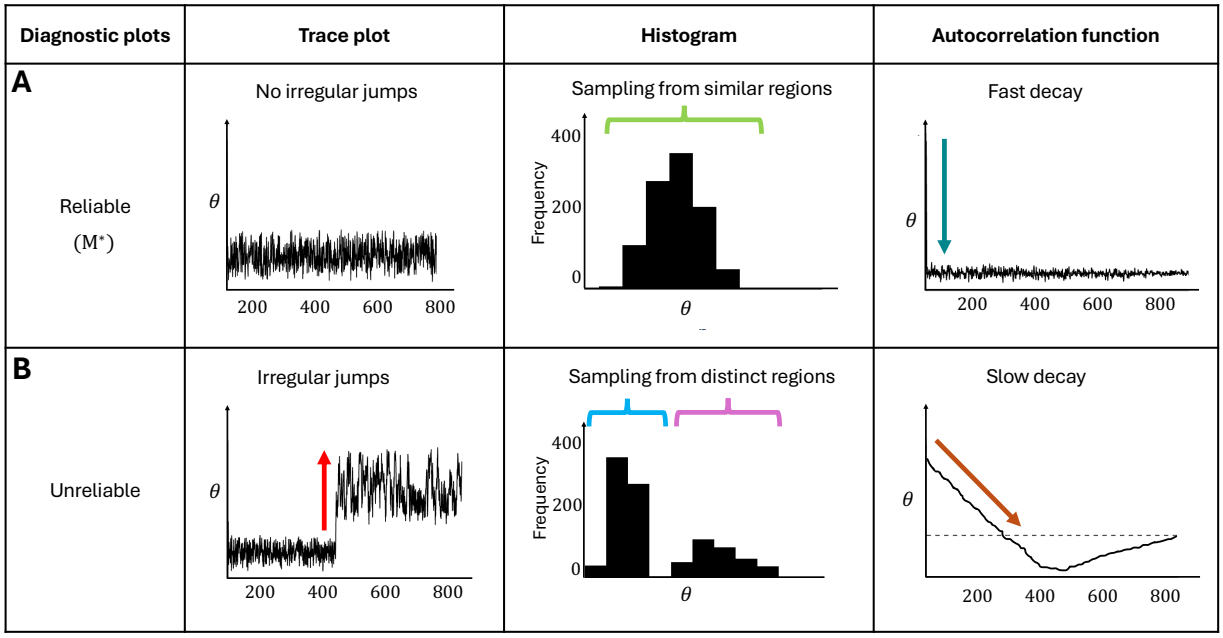

Figure 2. Diagnostic plots are used to assess convergence in the MCMC sampling step. *A) Illustrative example of reliable diagnostic plots that display well-mixed trace plots, unimodal histograms, and rapidly decaying autocorrelation. B) Illustrative example of unreliable diagnostic plots characterized by poor exploration of the parameter space, abrupt shifts in trace plots, and slowly decaying or non-converging autocorrelation functions.*

3.3 Testing the performance of TM-ABC-MCMC for complex biological systems

To evaluate the performance of TM-ABC-MCMC, we selected four models of varying complexity and precision in parameter estimates (see Tables S1- S4). For all tested models (except the one in Case Study 1), the original studies obtained posterior parameter distributions using simulated annealing, a robust parameter estimation technique that typically outperforms Bayesian methods[35]. We systematically evaluated the performance of the TM-ABC-MCMC method by beginning with the model that had the fewest equations and parameters and then progressing to more complex models. In each case, we verified we had

correctly implemented the model by reproducing figures in the original papers[37,38,40] before applying TM-ABC-MCMC.

Case Study 1: TM-ABC-MCMC significantly outperforms ABC methods in efficiency on the Lotka Volterra Model

We began by benchmarking the TM-ABC-MCMC algorithm against the standard ABC rejection algorithm applied to the Lotka–Volterra (LV) model[24,33,34]. We first compared TM-ABC-MCMC performance against ABC with the goal of quantifying the computational cost of generating approximately 1000 accepted samples from each of the algorithms. We compared the resulting parameter distributions of a and b, and the corresponding trajectories of x. The same comparison could be made using trajectories of y; however, since the process is identical, we arbitrarily chose to focus on x. Our results show that TM-ABC-MCMC required a dramatically shorter runtime (9 seconds versus 7 hours), yielded a higher rate of acceptance (12% vs 0.006%), and required fewer data points (5 versus 8) (**Table 1**). Importantly, TM-ABC-MCMC required only $N = 10^4$ samples versus the $N = 14.1 \times 10^6$ necessary for ABC. The range of ϵ values calculated for TM-ABC-MCMC was $[3.8, 5.5]$. To ensure a fair comparison with ABC, we focused on the subset of accepted parameters with $\epsilon \leq 4.3$, as in Toni et al. However, the remaining parameter sets (with ϵ values in the range $\epsilon \leq 5.5$) produced broader parameter distributions, leading to greater heterogeneity in the resulting trajectories, even though their ϵ values were only slightly above the comparison threshold. (**Figure 3**).

Algorithm	N	Data type	# Data points	Distance function	Execution time	# Accepted	ϵ	AR (%)
ABC	14.1×10^6	D_x, D_y	8	SSE	7 hours	912	4.3	0.006
TM-ABC-MCMC	10^4	$D_{Extrema}$	5	SSE	9 seconds	1263	4.3	12

Table 1. Comparison of ABC and TM-ABC-MCMC performance on the Lotka-Volterra model. *N indicates the total number of simulations, and AR is the acceptance rate. Both methods used the same distance function (sum of squared errors, SSE) and threshold ($\epsilon = 4.3$).*

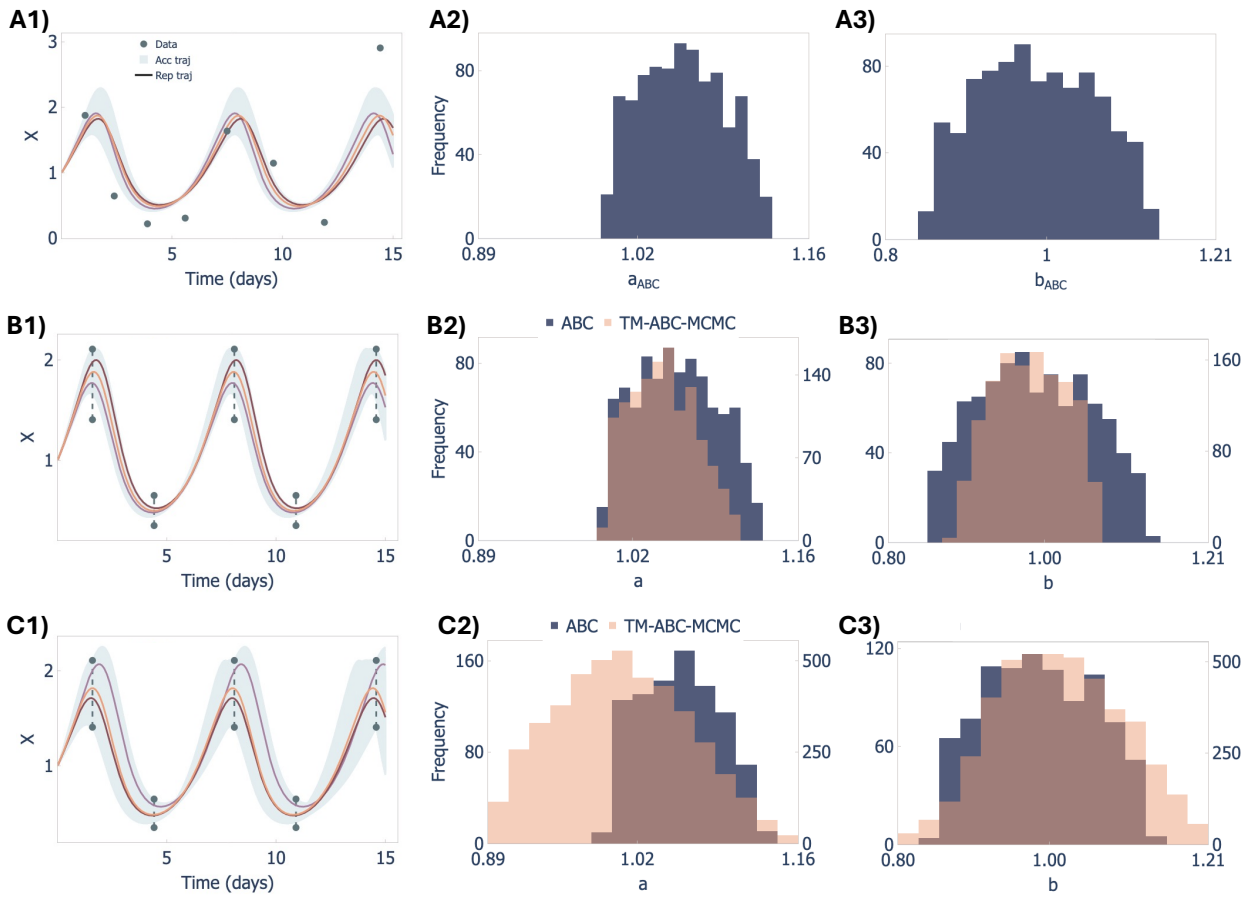

Figure 3. Benchmarking results comparing ABC to TM-ABC-MCMC for the Lotka-Volterra model. *A1) Accepted model trajectories for component x using the synthetic data D_x, A2) posterior distributions for parameter a, A3) posterior distributions for parameter b, using the ABC algorithm with an acceptance threshold of $\epsilon \leq 4.3$. B1-B3) Same as for panel A but using the TM-ABC-MCMC algorithm with an acceptance threshold of $\epsilon \leq 4.3$ and $D_{Extrema}$. C1-C3) Same as panel B but with a relaxed acceptance threshold ($\epsilon \leq 5.5$), illustrating how the TM-ABC-MCMC recovers broader and more diverse trajectories while maintaining good agreement with data.*

Case Study 2: Model of influenza virus infection and CD8⁺ T cell immune responses

Comparing TM-ABC-MCMC to ABC using a straightforward model like the Lotka-Volterra model allowed us to validate the computational performance of our algorithm. We now extend our approach to more mechanistic and biologically realistic models to assess their performance in increasingly complex systems. Specifically, we applied TM-ABC-MCMC to a model of influenza virus infection that includes density-dependent clearance of infected

cells[37]. To perform TM-ABC-MCMC, we used $M = 2$ parallel runs, each with $N = 2 \times 10^5$ iterations. At each iteration, a candidate parameter set $\Theta = \{\beta, p, c, k, \delta_d, K_\delta\}$ was sampled from normal distributions about best-fit values (see Methods). For a candidate solution to be accepted, predicted viral loads were required to be within the data's upper and lower bounds ($Eq.(2)$) with $q = 11$ for the $T = 11$ time points. Accepted model trajectories represented 1184 of the total simulations ($N = 2 \times 10^5$), yielding an acceptance rate of $0.6\ \%$. Resulting trajectories and illustrative distributions are shown in **Figure 4**. TM-ABC-MCMC completed in 13 minutes and produced parameter distributions consistent with Smith et al., demonstrating both significant computational efficiency and robustness. The accepted parameter distributions closely reproduced the heterogeneity reported in Smith et al[22] (Figure S1), which was further confirmed by recreating scatter plots between parameter pairs (Figure S2). We also implemented this model in pyABC package, i.e., a Python implementation of Approximate Bayesian Computation Sequential Monte Carlo (ABC-SMC)[12,24,27,44,45]. Given comparable scenarios, we found that TM-ABC-MCMC similarly to pyABC, though the acceptance threshold ϵ and prior definitions have a considerable effect on execution time and on the posterior distributions of some model parameters (see Section S10 in the Supplementary Information).

We then applied TM-ABC-MCMC to the extended model of murine influenza infection describing viral titers and CD8+ effector and memory T cell concentrations[38] (see Methods). TM-ABC-MCMC was implemented by performing $M = 2$ parallel runs, each with $N = 2 \times 10^5$ iterations. At each iteration, a candidate parameter set $\Theta = \{\beta,\ p,\ c,\ k,\ \delta_E,\ K_{\delta_E},\ \sigma,\ d_E,\ \tau_E,\ \zeta,\ \xi,\ \eta,\ \tau_M,\ K_E\}$ was sampled from a normal distribution. Because the data were a time-series with 10 independent individuals at each collection time, a candidate parameter set Θ was accepted only if the simulated outcomes for the model solutions for the virus fell within the virus data bounds and for the $CD8^+\ T$ cell fell within the $CD8^+\ T$ cell data bounds, for $q = 13$ total timepoints. TM-ABC-MCMC yielded an acceptance rate of $0.9\ \%$ (see resulting accepted trajectories in **Figure 5A** with illustrative distributions in **Figure 5B**; full parameter distributions are in Figures S3-S6) and completed in 10 minutes. Together,

these results show that even with increased model complexity, TM-ABC-MCMC maintains its strong computational performance and a high degree of accuracy, indicating its robustness and adaptability to more complex systems. The higher acceptance rate observed in this model, compared to the previous one, is attributable to setting the coefficient C to 0.07 instead of 0.1 (see Discussion). Scatter plots and results of diagnostic plots are also shown in Figures S3-S6.

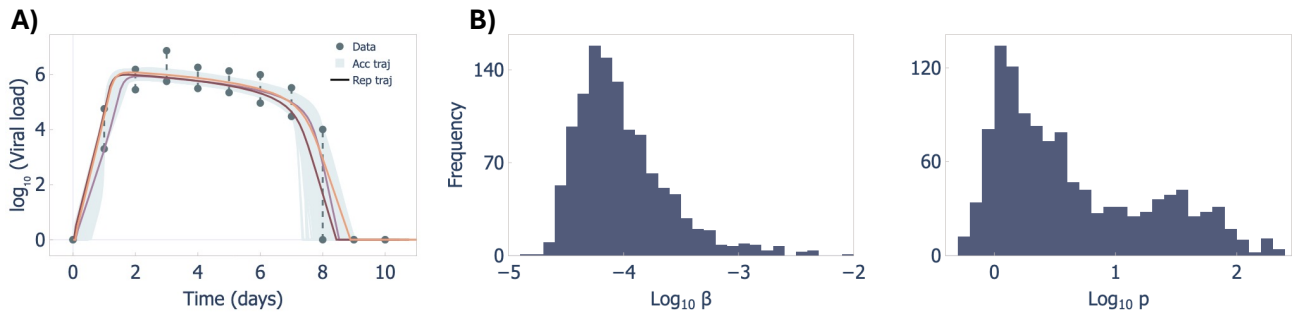

Figure 4. Outputs of the TM-ABC-MCMC algorithm on the density dependence viral kinetic model. *A) Accepted trajectories for virus dynamics. B-left) Posterior distribution of virus infection rate β, and B-right) virus production rate p.*

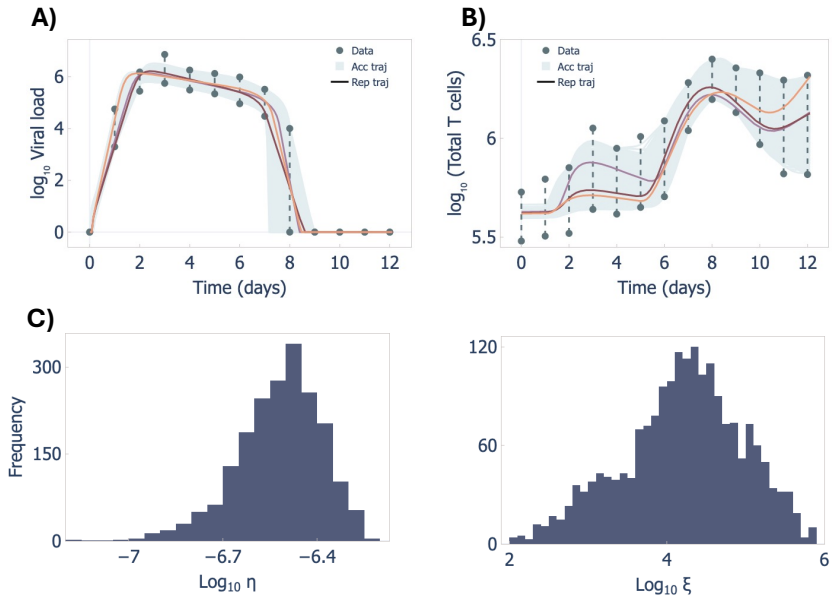

Figure 5. Output of TM-ABC-MCMC algorithm on extended model of influenza infection including $CD8^+$ T cell dynamics. *A) Accepted trajectories for viral dynamics and B) total number of $CD8^+$ T cells. C-left) Posterior distribution of the $CD8_E$ expansion rate η, and C-right) $CD8_E$ infiltration rate ξ.*

Case Study 3: Model of cancer immunotherapy

Lastly, we applied TM-ABC-MCMC to a model describing a combination cancer immunotherapy model[39,40,46]. To implement the TM-ABC-MCMC algorithm, we performed $M = 3$ parallel runs, each with $N = 5 \times 10^4$ iterations. At each iteration, a candidate parameter set $\Theta = \{r,\ \beta, \chi_D, c_A, c_T, c_{kill}\}$ was sampled from a normal distribution. In this case, only population-level data were available. Thus, to ensure acceptance ($Eq.(3)$), simulated trajectories of the candidate parameter set had to remain within the tumor volume bounds defined by $Mean \pm SE\ (Standard\ Error)$, with a 10% tolerance added to the standard error. In the original paper, the model tracks well with the experimental data after day 7 but fails to adequately capture the earlier dynamics from days 0-7. Thus, we adjusted the acceptance window to $q = 22$ time points and excluded the first seven points. We further included a check so that predicted viral loads remained below $100\ \times 10^7$ virions at day 30, consistent with the viral dynamics reported in Wares et al.[39]. This constraint effectively incorporates prior knowledge of viral dynamics into the selection of accepted trajectories, as in the previous Case Study (influenza infection) where we included information about both viral load and T cell dynamics. Given that we only required that model trajectories match 22 of 30 time points, the value of *N* could be reduced to 5×10^4, which is smaller than in the previously tested models. This also reflects the lower complexity of the mathematical model. Accepted model trajectories (**Figure 6A**; illustrative distributions in **Figure 6B**) comprised 2% of the total simulations (i.e., acceptance rate of 2%) and completed in 22 minutes. All other parameter distributions and results of diagnostic plots are shown in Supplementary Figure S7. The longer execution time for this model versus those of the previous models can be attributed to the additional time points (22 compared to 11–13 in earlier models).

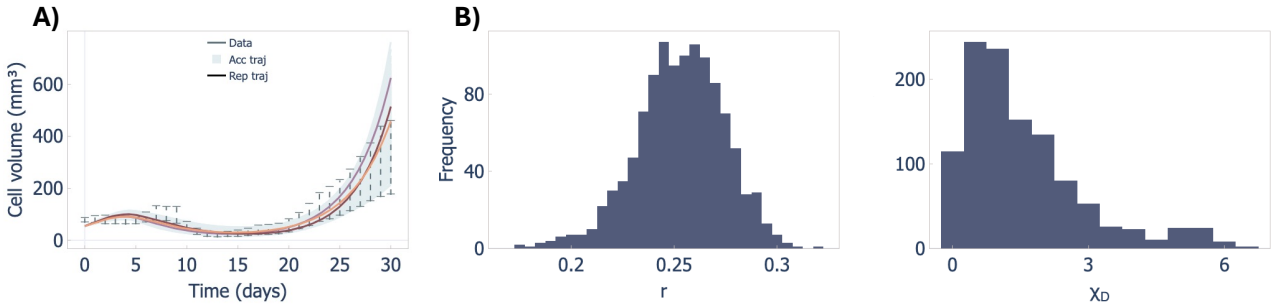

Figure 6. Output of the TM-ABC-MCMC algorithm on cancer immunotherapy model. *A) Accepted trajectories for the total number of tumor cells based on the TM-ABC-MCMC algorithm. B-left) Posterior distribution of tumor growth rate r and B-right) T cell stimulation rate χ_D by DCs.*

4. Discussion

Complex biological systems can exhibit substantial heterogeneity, making it challenging to experimentally or clinically capture the mechanisms driving individual variability. Existing computational methods like ABC-MCMC can help generate such variability but often suffer from high rejection rates[35], limiting efficiency. Here, we showed that trajectory-matching (TM)-ABC-MCMC effectively captures the variability inherent in biological systems by generating diverse model trajectories and parameter sets that reflect both data uncertainty and system complexity. By replacing distance-based comparison with an acceptance criterion that selects parameter sets whose outputs fall within observed data bounds, the high rejection rate observed in traditional ABC-MCMC was resolved. By leveraging best-fit parameter values estimated during model construction for run initialization, TM-ABC-MCMC also significantly reduces the risk of being in low-probability regions during the MCMC step, a problem in high-dimensional settings that can reduce accuracy. We also assessed the sensitivity of ABC variants to the acceptance threshold ϵ and the specification of priors using the Lotka-Volterra model in Case Study 1 and the infection model in Case Study 2A (see Section S10 in the Supplementary Information).

We showed that TM-ABC-MCMC accurately and efficiently generates plausible heterogeneous trajectories for biological systems. In each of our test cases, the TM-ABC-MCMC accurately and more rapidly reproduced parameter variability when compared to

both Bayesian and non-Bayesian techniques (**Table 1** and **Table 2**). This is particularly important as we move towards the integration of TM-ABC-MCMC for virtual patient generation, since VPCs must capture the heterogeneity of biological systems to be representative of real clinical populations.

Model	N	M	# Data points	C	Execution time	# Accepted	AR (%)
Case Study 2A: Influenza infection	2×10^5	2	11	0.1	13 minutes	1184	0.6
Case Study 2B: Extended influenza infection	2×10^5	2	13	0.07	10 minutes	1982	0.9
Case Study 3: Cancer immunotherapy	5×10^4	3	22	0.1	22 minutes	1131	2.2

Table 2. Summary of TM-ABC-MCMC performance for Case studies 2 and 3. *For each model, the number of samples (N), number of parallel runs (M), number of data points used for trajectory matching, standard deviation coefficient (C), execution time, number of accepted parameter sets, and acceptance rate (AR) are reported.*

Interestingly, we found that TM-ABC-MCMC improved the goodness of fit for the immuno-oncology model of Barish et al. In particular, when compared to the original model predictions, TM-ABC-MCMC generated tumor trajectories that better capture the experimental data from days 1-5, 15-25 and 29-30 (**Figure 7**). Notably, TM-ABC-MCMC was able to generate optimal distributions for the parameters c_A and c_{kill}, which were not resolved in the original analysis. In the original study, the authors fit their model hierarchically by fitting a smaller model, fixing estimated parameters, and progressively including more equations from which parameters were estimated by matching experimental data. It is important to exercise caution with hierarchical approach, as it may influence the resulting parameter distributions by limiting the extent to which the optimizer explores the full parameter landscape at each estimation step. Because the credible intervals in Barish et al.[40] were obtained through a hierarchical estimation process, we could not rely on the upper and lower bounds of the parameters to define the sampling spaces in normal distributions when simultaneously generating trajectories. We selected linear space for all the parameters (See SI for details). Further, the local sensitivity analysis performed by Barish et al. showed that variations up to 0.7% in β result in model fits within 10% of the optimum (SI of Barish et al.[40]). Based on this, we evaluated two scenarios, unrestricted and restricted infection rate, with details provided in the SI. We found that constraining the range of β,

resulted in greater consistency with the findings of Barish et al.[40], likely because β was estimated early in the hierarchical fitting procedure described in the original study.

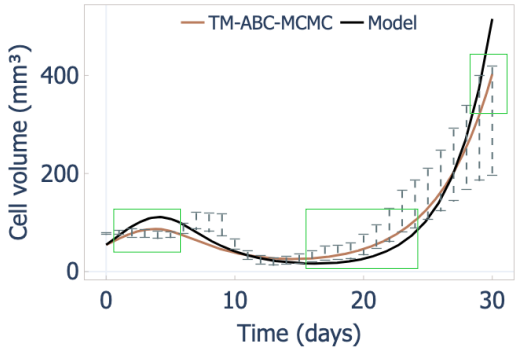

Figure 7. Improved goodness at specific time points of fit using TM-ABC-MCMC. *TM-ABC-MCMC generated tumor trajectories better capturing the experimental data at days 1-5, 15–25 and 29–30, highlighted with green boxes.*

The computational time of TM-ABC-MCMC scales with several interacting factors, including the number of measurements, model complexity, the selected differential equation solver, and the standard deviation coefficient C. First, as mentioned in the results of Case Study 3 (immuno-oncology), increasing the number of data/time points used during the trajectory matching step increases the execution time of TM-ABC-MCMC. Further, as each sampled parameter set must be applied, solved, and assessed within the model, the speed of any given simulation function will be a limiting factor for efficiency. We also observed that lower values of C improve the acceptance rate and reduce N, i.e., the total number of required samples. In general, as the complexity of the model increases and more parameters are introduced, lower values of C are preferable. In this study, we used $C = 0.1$ in influenza infection and cancer immunotherapy model (both models with six parameters) and $C = 0.07$ in the extended influenza infection model (with 14 parameters) to generate the desired 1000-2000 accepted samples (See Figure S8 for details on the sufficient number of accepted samples). Lastly, when confidence intervals are lacking, our results show that using well-defined confidence intervals slightly improved performance compared to those using constant standard deviations (Figure S10), whereas the choice of the standard deviation for the proposal distribution $\mathcal{N}(\mu, \sigma^2)$ critically affects sampling efficiency. Indeed, a small σ may result in limited exploration of the parameter space, whereas a large σ can lead to

frequent rejections. Therefore, selecting an appropriate σ is essential to balance exploration and acceptance rate[18].

TM-ABC-MCMC has two main limitations. First, its performance depends on multiple factors, including the quality of the available clinical or experimental data, the ability of the mechanistic model to capture the data, and the precision of parameter estimation. Second, the algorithm may not fully explore the entire parameter space for certain parameters in some cases. Although using uniform distributions would help ensure the parameter space is broadly explored, they increase computational times and are conceptually inconsistent with the methodology of TM-ABC-MCMC that uses normal distributions for the MCMC sampling. Further, our results show that the complex and non-uniform parameter patterns of each of the models studied here are not adequately captured when starting from uniform distributions with fixed bounds. This is again due to the use of MCMC sampling, where each new set of parameters depends on the ones accepted in previous steps. Future improvements to TM-ABC-MCMC could focus on extending its applicability to scenarios with limited data availability, thereby broadening its potential use in generating virtual patient cohorts and scenarios where parameters are more sloppily estimated (see Section S8 in the Supplementary Information). However, although we initialized TM-ABC-MCMC using the best-fit parameter values derived from prior analyses in this study, the algorithm does not require initialization strictly at such best-fit values. We tested TM-ABC-MCMC initialization with half of the parameters deliberately set away from their best-fit values (randomly selected), and the results were still acceptable. This suggests that the algorithm performs best when initialized with all best-fit values, but acceptable results can still be obtained even when some parameters are not at their best-fit values, provided they lie within their corresponding confidence intervals (Figure S11). This flexibility implies that the algorithm requires only a reasonable starting value within the plausible parameter range, rather than an optimal one. This feature enhances the robustness and applicability of the method, especially in scenarios where best-fit values may be uncertain or computationally expensive to obtain.

In summary, in the context of more complex models, existing methods for virtual patient cohort and virtual clinical trial generation often require execution times on the order of several days. By comparison, TM-ABC-MCMC consistently completes within an order of minutes, offering a substantial improvement in computational efficiency. This advantage becomes particularly critical when used to capture population heterogeneity or when repeated simulations are necessary for model-based analyses. Hence, TM-ABC-MCMC provides a novel, rigorous framework for capturing and quantifying heterogeneity in mechanistic mathematical models, enabling more realistic virtual patient cohort generation and improving the design and interpretation of virtual clinical trials.

ACKNOWLEDGMENTS

This work was funded by the Fondation CHUSJ and a Vanier Canada Graduate Scholarship (SHC), NIH NIAID R01 AI170115 (AMS and MC) and, in part, thanks to the Canada Research Chair in Computational Immunology with funding from the Canada Research Chairs Program (MC).

AUTHOR CONTRIBUTIONS

Conceptualized the research: FB, JJAW, SHC, CJS, AMS, MC
Designed the algorithm: FB and MC
Performed the research: FB
Wrote the manuscript: FB and MC
Edited the manuscript: FB, JJAW, SHC, CJS, AMS, MC

DECLARATION OF INTERESTS

The authors have no conflict of interests to declare.

CODE AVAILABILITY

The code to run Trajectory-Matching Approximate Bayesian Markov chain Monte-Carlo (TM-ABC-MCMC) and to generate the figures in this paper is available at:

https://github.com/Craig-Lab/TM-ABC-MCMC.

References

1 Sinisi, S., Alimguzhin, V., Mancini, T., Tronci, E. & Leeners, B. Complete populations of virtual patients for in silico clinical trials. *Bioinformatics* **36**, 5465-5472 (2020).

2 Allen, R., Rieger, T. R. & Musante, C. J. Efficient generation and selection of virtual populations in quantitative systems pharmacology models. *CPT: pharmacometrics & systems pharmacology* **5**, 140-146 (2016).

3 Craig, M., Gevertz, J. L., Kareva, I. & Wilkie, K. P. A practical guide for the generation of model-based virtual clinical trials. *Frontiers in Systems Biology* **3**, 1174647 (2023).

4 Wang, H., Arulraj, T., Ippolito, A. & Popel, A. S. From virtual patients to digital twins in immuno-oncology: lessons learned from mechanistic quantitative systems pharmacology modeling. *NPJ Digital Medicine* **7**, 189 (2024).

5 Braniff, N. *et al.* An integrated quantitative systems pharmacology virtual population approach for calibration with oncology efficacy endpoints. *CPT: Pharmacometrics & Systems Pharmacology* (2025).

6 Derippe, T., Fouliard, S., Declèves, X. & Mager, D. E. Accelerating robust plausible virtual patient cohort generation by substituting ODE simulations with parameter space mapping. *Journal of Pharmacokinetics and Pharmacodynamics* **49**, 625-644 (2022).

7 Rieger, T. R. *et al.* Improving the generation and selection of virtual populations in quantitative systems pharmacology models. *Progress in biophysics and molecular biology* **139**, 15-22 (2018).

8 Pappalardo, F., Russo, G., Tshinanu, F. M. & Viceconti, M. In silico clinical trials: concepts and early adoptions. *Briefings in bioinformatics* **20**, 1699-1708 (2019).

9 Romero, P. *et al.* Clinically-driven virtual patient cohorts generation: an application to aorta. *Frontiers in Physiology* **12**, 713118 (2021).

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

11 Beaumont, M. A., Cornuet, J.-M., Marin, J.-M. & Robert, C. P. Adaptive approximate Bayesian computation. *Biometrika* **96**, 983-990 (2009).

12 Beaumont, M. A. Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics* **41**, 379-406 (2010).

13 Csilléry, K., Blum, M. G., Gaggiotti, O. E. & François, O. Approximate Bayesian computation (ABC) in practice. *Trends in ecology & evolution* **25**, 410-418 (2010).

14 Marin, J.-M., Pudlo, P., Robert, C. P. & Ryder, R. J. Approximate Bayesian computational methods. *Statistics and computing* **22**, 1167-1180 (2012).

15 Gourieroux, C., Monfort, A. & Renault, E. Indirect inference. *Journal of applied econometrics* **8**, S85-S118 (1993).

16 Price, L. F., Drovandi, C. C., Lee, A. & Nott, D. J. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics* **27**, 1-11 (2018).

17 Brooks, S. Markov chain Monte Carlo method and its application. *Journal of the royal statistical society: series D (the Statistician)* **47**, 69-100 (1998).

18 Andrieu, C., De Freitas, N., Doucet, A. & Jordan, M. I. An introduction to MCMC for machine learning. *Machine learning* **50**, 5-43 (2003).

19 Bernardo, J. *et al.* in *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*. 181 (Citeseer).

20 Marjoram, P., Molitor, J., Plagnol, V. & Tavaré, S. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* **100**, 15324-15328 (2003).

21 Wegmann, D., Leuenberger, C. & Excoffier, L. Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics* **182**, 1207-1218 (2009).

22 Sadegh, M. & Vrugt, J. A. Approximate bayesian computation using markov chain monte carlo simulation: Dream (abc). *Water Resources Research* **50**, 6767-6787 (2014).

23 Schirru, M. *et al.* Generation of virtual populations for quantitative systems pharmacology through advanced sampling methods. *bioRxiv*, 2025.2006.2024.661262 (2025). https://doi.org/10.1101/2025.06.24.661262

24 Toni, T., Welch, D., Strelkowa, N., Ipsen, A. & Stumpf, M. P. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface* **6**, 187-202 (2009).

25 Turner, B. M. & Van Zandt, T. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology* **56**, 69-85 (2012).

26 Hartig, F., Calabrese, J. M., Reineking, B., Wiegand, T. & Huth, A. Statistical inference for stochastic simulation models–theory and application. *Ecology letters* **14**, 816-827 (2011).

27 Del Moral, P., Doucet, A. & Jasra, A. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **68**, 411-436 (2006). https://doi.org/10.1111/j.1467-9868.2006.00553.x

28 Moingeon, P., Chenel, M., Rousseau, C., Voisin, E. & Guedj, M. Virtual patients, digital twins and causal disease models: Paving the ground for in silico clinical trials. *Drug discovery today* **28**, 103605 (2023).

29 Polasek, T. M. & Rostami-Hodjegan, A. Virtual Twins: Understanding the Data Required for Model-Informed Precision Dosing. *Clinical Pharmacology & Therapeutics* **107** (2020).

30 Alfonso, S., Jenner, A. L. & Craig, M. Translational approaches to treating dynamical diseases through in silico clinical trials. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30** (2020).

31 Cheng, Y. *et al.* Virtual Populations for Quantitative Systems Pharmacology Models. *Methods Mol Biol* **2486**, 129-179 (2022). https://doi.org/10.1007/978-1-0716-2265-0_8

32 Huang, L. *et al.* A surrogate workflow for virtual population development: designing an amenable proxy for a hypothesis-test inspired nonlinear goodness-of-fit. *bioRxiv*, 2025.2004.2026.649406 (2025). https://doi.org/10.1101/2025.04.26.649406

33 Volterra, V. *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*. (Società anonima tipografica" Leonardo da Vinci", 1926).

34 Lotka, A. J. *Elements of physical biology*. (Williams & Wilkins, 1925).

35 Chung, M. *et al*. PARAMETER AND UNCERTAINTY ESTIMATION FOR DYNAMICAL SYSTEMS USING SURROGATE STOCHASTIC PROCESSES. *SIAM J Sci Comput* **41**, A2212-a2238 (2019). https://doi.org/10.1137/18m1213403

36 Huang, J.-H. *et al*. Therapeutic and tumor-specific immunity induced by combination of dendritic cells and oncolytic adenovirus expressing IL-12 and 4-1BBL. *Molecular Therapy* **18**, 264-274 (2010).

37 Smith, A. P., Moquin, D. J., Bernhauerova, V. & Smith, A. M. Influenza virus infection model with density dependence supports biphasic viral decay. *Frontiers in microbiology* **9**, 1554 (2018).

38 Myers, M. A. *et al*. Dynamically linking influenza virus infection kinetics, lung injury, inflammation, and disease severity. *Elife* **10**, e68864 (2021).

39 Wares, J. R. *et al*. Treatment strategies for combining immunostimulatory oncolytic virus therapeutics with dendritic cell injections. *Mathematical Biosciences & Engineering* **12**, 1237-1256 (2015).

40 Barish, S., Ochs, M. F., Sontag, E. D. & Gevertz, J. L. Evaluating optimal therapy robustness by virtual expansion of a sample population, with a Case Study in cancer immunotherapy. *Proceedings of the National Academy of Sciences* **114**, E6277-E6286 (2017).

41 Hamra, G., MacLehose, R. & Richardson, D. Markov chain Monte Carlo: an introduction for epidemiologists. *International journal of epidemiology* **42**, 627-634 (2013).

42 Roberts, G. O. & Rosenthal, J. S. General state space Markov chains and MCMC algorithms. (2004).

43 Van Oijen, M., Rougier, J. & Smith, R. Bayesian calibration of process-based forest models: bridging the gap between models and data. *Tree physiology* **25**, 915-927 (2005).

44 Schälte, Y., Klinger, E., Alamoudi, E. & Hasenauer, J. pyABC: Efficient and robust easy-to-use approximate Bayesian computation. *arXiv preprint arXiv:2203.13043* (2022).

45 Klinger, E., Rickert, D. & Hasenauer, J. pyABC: distributed, likelihood-free inference. *Bioinformatics* **34**, 3591-3593 (2018). https://doi.org/10.1093/bioinformatics/bty361

46 Kim, P. S., Crivelli, J. J., Choi, I.-K., Yun, C.-O. & Wares, J. R. Quantitative impact of immunomodulation versus oncolysis with cytokine-expressing virus therapeutics. *Mathematical Biosciences & Engineering* **12**, 841-858 (2015).