

RESEARCH ARTICLE

Likelihood-free nested sampling for parameter inference of biochemical reaction networks

Jan Mikelson , Mustafa Khammash *

D-BSSE, ETH-Zurich, Zurich, Switzerland

* mustafa.khammash@bsse.ethz.ch

Abstract

The development of mechanistic models of biological systems is a central part of Systems Biology. One major challenge in developing these models is the accurate inference of model parameters. In recent years, nested sampling methods have gained increased attention in the Systems Biology community due to the fact that they are parallelizable and provide error estimates with no additional computations. One drawback that severely limits the usability of these methods, however, is that they require the likelihood function to be available, and thus cannot be applied to systems with intractable likelihoods, such as stochastic models. Here we present a likelihood-free nested sampling method for parameter inference which overcomes these drawbacks. This method gives an unbiased estimator of the Bayesian evidence as well as samples from the posterior. We derive a lower bound on the estimator variance which we use to formulate a novel termination criterion for nested sampling. The presented method enables not only the reliable inference of the posterior of parameters for stochastic systems of a size and complexity that is challenging for traditional methods, but it also provides an estimate of the obtained variance. We illustrate our approach by applying it to several realistically sized models with simulated data as well as recently published biological data. We also compare our developed method with the two most popular other likelihood-free approaches: pMCMC and ABC-SMC. The C++ code of the proposed methods, together with test data, is available at the github web page https://github.com/Mijan/LFNS_paper.

OPEN ACCESS

Citation: Mikelson J, Khammash M (2020) Likelihood-free nested sampling for parameter inference of biochemical reaction networks. *PLoS Comput Biol* 16(10): e1008264. <https://doi.org/10.1371/journal.pcbi.1008264>

Editor: Stacey Finley, University of Southern California, UNITED STATES

Received: November 15, 2019

Accepted: August 16, 2020

Published: October 9, 2020

Copyright: © 2020 Mikelson, Khammash. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: The software along with all examples and numerical results can be found at https://github.com/Mijan/LFNS_paper.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 730964 (TRANSVAC2). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

The behaviour of mathematical models of biochemical reactions is governed by model parameters encoding for various reaction rates, molecule concentrations and other biochemical quantities. As the general purpose of these models is to reproduce and predict the true biological response to different stimuli, the inference of these parameters, given experimental observations, is a crucial part of Systems Biology. While plenty of methods have been published for the inference of model parameters, most of them require the availability of the likelihood function and thus cannot be applied to models that do not allow for the computation of the likelihood. Further, most established methods do not

provide an estimate of the variance of the obtained estimator. In this paper, we present a novel inference method that accurately approximates the posterior distribution of parameters and does not require the evaluation of the likelihood function. Our method is based on the nested sampling algorithm and approximates the likelihood with a particle filter. We show that the resulting posterior estimates are unbiased and provide a way to estimate not just the posterior distribution, but also an error estimate of the final estimator. We illustrate our method on several stochastic models with simulated data as well as one model of transcription with real biological data.

This is a *PLOS Computational Biology Methods* paper.

Introduction

The accurate modelling and simulation of biological processes such as gene expression or signalling has gained a lot of interest over the last years, resulting in a large body of literature addressing various types of models along with the means for their identification and simulation. The main purpose of these models is to qualitatively or quantitatively describe observed biological dynamics while giving insights into the underlying bio-molecular mechanisms.

One important aspect in the design of these models is the determination of the model parameters. Often there exists a mechanistic model of the cellular processes, but their parameters (e.g. reaction rates or initial molecule concentrations) are largely unknown. Since the same network topology may result in different behaviour depending on the chosen parameters ([1]), this presents a major challenge for modelling and underscores the need for effective parameter estimation techniques.

The models used in Systems Biology can be coarsely classified into two groups: deterministic and stochastic models. Deterministic models usually rely on ordinary differential equations which, given the parameters and initial conditions, can describe the time evolution of the biological system in a deterministic manner. However, many cellular processes like gene expression are subject to random fluctuations ([2, 3]), which can have important biological functions ([4–6]) as well as contain useful information about the underlying molecular mechanisms ([7]). The important role of stochastic fluctuations in biological systems has led to increased interest in stochastic models and methods for their parameter inference ([8–12]). Such stochastic models are usually described in the framework of stochastic chemical reaction networks that can be simulated using Gillespie's Stochastic Simulation Algorithm (SSA) [13]. In recent years, the availability of single-cell trajectory data has drastically increased, providing detailed information about the (potentially stochastic) development of single cells throughout time.

Despite the increasing interest in stochastic systems, performing inference on them is still challenging and the available methods are computationally very demanding (see for instance [8, 14, 15]). Several algorithms have been put forward to deal with such problems, such as various kinds of sequential Monte Carlo methods (SMC) ([16, 17]), Markov Chain Monte Carlo (MCMC) methods ([8, 18, 19]), approximate Bayesian computation (ABC) methods ([20, 21]), iterative filtering ([22]) and nested sampling (NS) approaches ([23–25]). However, for the problem of likelihood-free Bayesian inference the two most widely used methods are particle

MCMC (pMCMC) ([26]) and ABC-SMC ([20]), as discussed in various recent review papers [27–29]. Furthermore, to reduce computational complexity, several of these inference methods rely on approximating the model dynamics (for instance using the diffusion approximation ([30]) or linear noise approximation ([31]). However, these approximations may not always be justifiable (in the case of low copy numbers of the reactants for example) and might obscure crucial system behaviour.

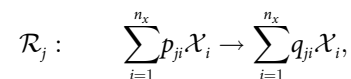
In this paper, we focus on nested sampling methods and investigate its applicability to stochastic systems. Coming originally from the cosmology community, NS has gained increasing popularity and found also applications in Systems Biology (see for instance [32–35]). Several implementations of NS are available ([36, 37]) and in [38] the authors even provide a NS implementation specifically for a Systems Biology context. Even though the original purpose of NS was to efficiently compute the Bayesian evidence, it has more and more become a viable alternative to MCMC methods for the approximation of the posterior (see for instance [39, 40]).

There are various reasons for the interest in NS which are discussed in detail in [41, 42] and the references within. Some of the rather appealing features of NS is that it performs well for multimodal distributions ([37, 39]), is easily parallelizable ([33, 43]) and provides a natural means to compute error bars on all of its results without needing multiple runs of the algorithm ([41, 44]). For a comparison of MCMC and NS see for instance [35, 41], for a discussion of other methods to compute the Bayesian evidence using MCMC see [41, 45]. Like standard MCMC methods, NS requires the availability of the likelihood which limits its use to models that allow for the computation of the likelihood such as deterministic models and simple stochastic models. In this paper, we consider an extension to the original NS framework that, similarly to the particle MCMC method ([46]) and particle SMC ([47]), allows the use of approximated likelihoods instead of the actual likelihood to be used for NS. In the following we introduce the notation and problem formulation, the “Materials and methods” section is dedicated to the likelihood-free NS formulation and in the section “Results” we demonstrate its performance on several chosen examples.

Chemical reaction networks

We are considering a n_x -dimensional Markov Process $X(t)$ depending on a d -dimensional parameter vector θ . We denote with $X_i(t)$ the i^{th} entry of the state vector at time t and with $X(t) = \{X_i(t)\}_{i=1, \dots, n_x}$ the state vector at time t . We will write $X_\tau = X(t_\tau)$ when talking about the state vector at a timepoint t_τ indexed with τ .

In the context of stochastic chemical reaction networks this Markov process describes the abundances of n_x species $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{n_x}$, reacting through n_R reactions $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{n_R}$ written as



where p_{ji} is the numbers of molecules of species \mathcal{X}_i involved in reaction \mathcal{R}_j , and q_{ji} is the number of molecules of species \mathcal{X}_i produced by that reaction. The random variable $X_i(t)$ corresponds to the number of molecules of species \mathcal{X}_i at time t . Each reaction \mathcal{R}_j has an associated propensity. The reaction propensities at a given time t depend on the current state $X(t)$ and on the parameter vector θ .

General task

The process $X(t)$ is usually not directly observable but can only be observed indirectly through a n_y -dimensional observation vector

$$Y_\tau \sim p(\cdot | X_\tau, \theta),$$

which depends on the state X_τ and on the parameter vector $\theta \in \Omega$, where $\Omega \subseteq \mathbb{R}^d$ denotes the parameter space. We use the notation $p(\cdot | X_\tau, \theta)$ to emphasize that we think of the measurement Y_τ as a random variable sample from a conditional distribution $p(\cdot | X_\tau, \theta)$. We shall assume that the variable Y is not observed at all times but only on T timepoints t_1, \dots, t_T and only for M different trajectories. With \mathbf{y} we denote the collection of observations at all time points. In the Bayesian approach the parameter vector θ is treated as a random variable with associated prior $\pi(\theta)$. The goal is not to find just one set of parameters, but rather to compute the posterior distribution $\mathcal{P}(\theta | \mathbf{y})$ of θ

$$\mathcal{P}(\theta | \mathbf{y}) = \frac{1}{Z} l(\mathbf{y} | \theta) \pi(\theta),$$

where $l(\mathbf{y} | \theta)$ (we will also write $l(\theta)$ if the dependence on \mathbf{y} is clear from the context) is the likelihood of θ for the particular observation \mathbf{y} and Z is the Bayesian evidence

$$Z = \int_{\Omega} l(\mathbf{y} | \theta) d\pi(\theta). \quad (1)$$

This has several advantages over a single point estimate as it gives insight into the areas of the parameter space resulting in model behaviour similar to the observations as well as about their relevance for the simulation outcome (a wide posterior indicates non-identifiability for example). The notation $d\pi(\theta)$ above indicates that the integral is taken over the prior distribution. For a detailed discussion of Bayesian approaches see for instance [45]. In this paper we follow the Bayesian approach and aim to recover the posterior $\mathcal{P}(\theta | \mathbf{y})$. In the following we briefly outline the basic nested sampling approach.

Nested sampling (NS)

Nested sampling is a Bayesian inference technique that was originally introduced in [23] to compute the Bayesian evidence (1). NS can be viewed as an importance sampling technique (as for instance discussed in [48]) as it approximates the evidence by generating samples θ_i , weights w_i and likelihoods $l_i = l(\theta_i)$ such that the weighted samples can be used to obtain numerical approximations \hat{Z} of the evidence (1)

$$\hat{Z} = \sum_i w_i l_i \approx \int l(\theta) d\pi(\theta). \quad (2)$$

NS samples parameter vectors (particles) from the prior distribution constrained to super-level sets of the likelihood

$$\pi(\theta | l(\theta) > \epsilon_i) \quad (3)$$

corresponding to an increasing sequence of thresholds ϵ_i . The NS sampling scheme iteratively removes at each iteration i the particle θ_i with the lowest likelihood $l(\theta_i) = \epsilon_i$ from the current set of particles \mathcal{L}_i (called “live” set) and replaces it with a newly sampled particle with a likelihood higher than ϵ_i to obtain the next set \mathcal{L}_{i+1} . This way, while still suffering from the “curse of dimensionality”, NS exponentially shrinks the sample space to regions of high likelihood and

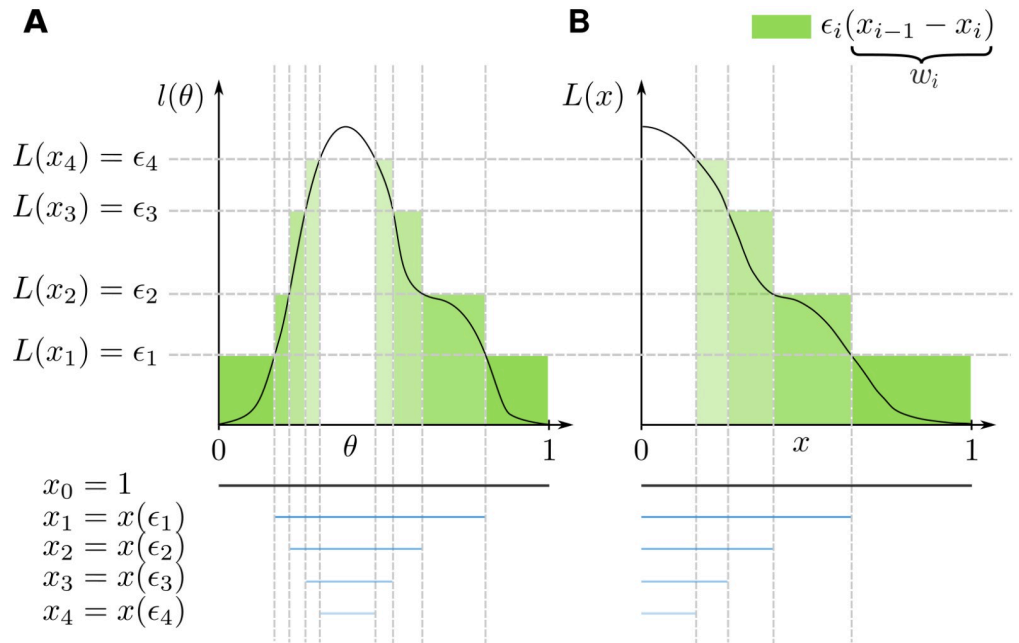


Fig 1. Illustration of the nested sampling approximation with a uniform prior on [0, 1]. **A:** The integral over the parameter space $\int_{\Omega} l(\theta)d\theta$. **B:** The transformed integral $\int_0^1 L(x)dx$ over the prior volume x .

<https://doi.org/10.1371/journal.pcbi.1008264.g001>

allows one to reach these regions in a reasonable number of iterations. Nested sampling exploits the fact that the Bayesian evidence (1) can also be written (see [23]) as a one-dimensional integral

$$Z = \int_0^1 L(x)dx,$$

over the prior volume

$$x(\epsilon) := \pi(l(\theta) > \epsilon) = \int_{l(\theta) > \epsilon} d\pi(\theta),$$

where $L(x)$ denotes the likelihood corresponding to the constrained prior with volume x

$$L(x) = \arg \inf_{\epsilon} \{x(\epsilon) \geq x\}. \tag{4}$$

For this reformulation to hold some weak conditions have to be satisfied, see for details [49] and [25]. For an illustration of the above quantities see Fig 1A and 1B. The NS sampling scheme approximates the prior volumes x_i through $\hat{x}_i = t^{(i)}x_{i-1}$, where $t^{(i)}$ denotes the i^{th} sample from a Beta distribution. These approximated prior volumes allow to compute the weights as $w_i = \hat{x}_{i+1} - \hat{x}_i$ in (2). One can also use the weights $l_i \times w_i$ instead of w_i to approximate functions over the posterior $\mathcal{P}(\theta)$

$$\frac{1}{Z} \sum f(\theta_i) l_i w_i \approx \int f(\theta) d\mathcal{P}(\theta).$$

For a more detailed description of NS see [S1 Appendix](#).

Materials and methods

In many cases (such as most of the above mentioned stochastic models) the likelihood $l(\theta)$ cannot be directly computed, making approaches like MCMC methods or nested sampling not applicable. Fortunately, many variations of MCMC have been described circumventing this problem by introducing likelihood-free MCMC methods such as [50] or [46] as well as other likelihood-free methods such as ABC ([20]) or likelihood-free SMC methods ([51]). These approaches usually rely on forward simulation of a given parameter vector θ to obtain a simulated data set that can then be compared to the real data or can be used to compute a likelihood approximation $\hat{l}(\theta) \approx l(\theta)$. In the following we briefly illustrate one such likelihood approximation.

Likelihood approximation using particle filters

A common way to approximate the likelihood through forward simulation is using a particle filter (see for instance [52]), which iteratively simulates the stochastic system with H particles and then resamples these particles. The main idea behind particle filters is to exploit the recursive relationship

$$p(y_1, \dots, y_t | \theta) = p(y_1, \dots, y_{t-1} | \theta) \int p(y_t | X_t) p(X_t | y_1, \dots, y_{t-1}, \theta) dX_t$$

where y_1, \dots, y_t denotes all observations until timepoint t and X_t the (possibly unobserved) system state at time t . The above integral can be approximated by creating samples $x_{t,i}$ from the distribution $p(X_t | y_1, \dots, y_{t-1}, \theta)$, through forward simulation of the system and resampling the simulated paths weighted by the likelihood at each timepoint. Then the likelihood for all observations up until timepoint t can be approximated by

$$\int p(y_t | X_t) p(X_t | y_1, \dots, y_{t-1}, \theta) dX_t \approx \frac{1}{H} \sum_{h=1}^H p(y_t | x_{t,i}).$$

As the above approximation is unbiased, the resulting particle filter approximation of the likelihood is unbiased as well. In the following we illustrate such a particle filter likelihood approximation on a simple birth death model, where one species (mRNA) is produced at rate $k = 1$ and degrades at rate $\gamma = 0.1$. We simulated one trajectory of this system using SSA and, using the finite state projection (FSP [53]), computed the likelihood $l(k)$ for different values of k while keeping γ fixed to 0.1. The true likelihood for different k is shown as the solid red line in [Fig 2A and 2B](#). We also illustrated the likelihood approximation $\hat{l}(k)$ using a particle filter with $H = 100$ particles for three values of k . For each of the values for k we computed 1000 realizations of $\hat{l}(k)$ and plotted the empirical distributions in [Fig 2B](#). Note that $\hat{l}(k)$ is itself a random variable with distribution $p(\hat{l}(k) | k)$ and has a mean equal to the true likelihood $\mathbb{E}(\hat{l}(k)) = l(k)$ (see for instance [52]). We also sampled 10^6 values of k from a log uniform prior and approximate for each k its likelihood with the same particle filter with $H = 100$ particles. We plotted the contour lines of this joint distribution

$$\Pi(k, \hat{l}(k)) = \pi(k) p(\hat{l}(k) | k)$$

in [Fig 2A](#).

In the following we discuss how to utilize such a likelihood approximation to apply the above described NS procedure to cases where the likelihood is not available. Throughout the

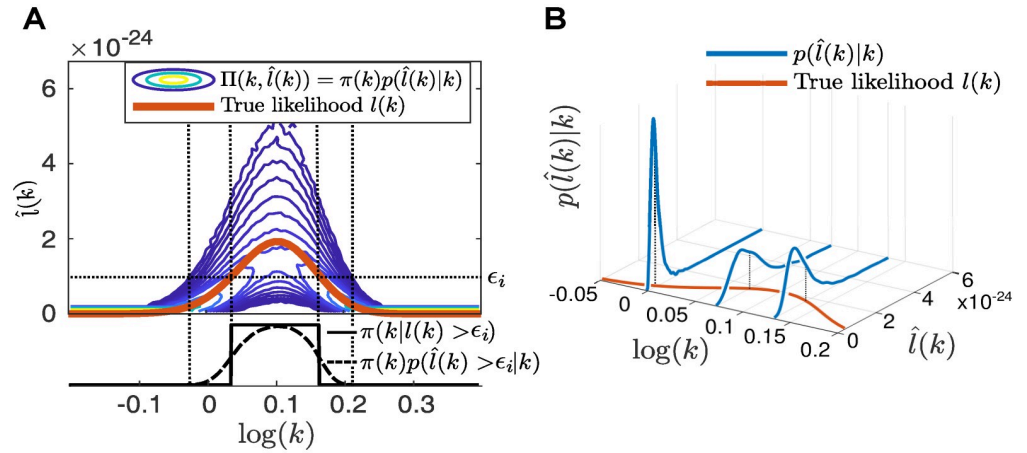


Fig 2. Illustration of likelihood approximation with a particle filter. **A:** Top: Likelihood for different parameters k (red) and contour lines of the joint distribution $\Pi(k, \hat{l}(k)) = \pi(k)p(\hat{l}(k)|k)$ of the parameter k and its likelihood approximation $\hat{l}(k)$, based on 10^6 samples of the likelihood approximation obtained with a particle filter with 100 particles. Bottom: The constrained priors $\pi(k|l(k) > \epsilon)$ and $\pi(k)p(\hat{l}(k)) > \epsilon|k$ for $\epsilon = 1e - 24$. **B:** Example distributions $p(\hat{l}(k)|k)$ (blue) for $k = 1, 1.2$ and 1.4 and the true likelihood $l(k)$ red.

<https://doi.org/10.1371/journal.pcbi.1008264.g002>

paper we assume that the likelihood approximation $\hat{l}(\theta)$ is obtained using a particle filter, but our result hold for any unbiased likelihood estimator.

The LF-NS scheme

For NS, the constraint prior $\pi(\theta|l(\theta) > \epsilon_i)$ needs to be sampled. Since in the likelihood-free case, the likelihood $l(\theta)$ is not available and $\hat{l}(\theta)$ is itself a random variable, the set $\{\theta \in \Omega|\hat{l}(\theta) > \epsilon_i\}$ (which is the support of the constrained prior) is not defined. To apply the NS idea to the likelihood-free case, we propose to perform the NS procedure on the joint prior

$$\Pi(\theta, \hat{l}(\theta)) = \pi(\theta)p(\hat{l}(\theta)|\theta) \tag{5}$$

on the set $\Omega \times \mathbb{R}_{>0}$. This joint prior can be sampled by drawing a sample θ^* from the prior $\pi(\theta)$ and then drawing one sample \hat{l}^* from the distribution of likelihood approximations $p(\hat{l}(\theta^*)|\theta^*)$. With such a sampling scheme we perform the NS steps of constructing the set of “dead” particles \mathcal{D} on the joint prior (5). As in standard NS, we sample a set of N “live” particles $\{\theta, \hat{l}\}$ from $\Pi(\theta, \hat{l}(\theta))$, then we iteratively remove the particle $\{\theta, \hat{l}\}$ with the lowest likelihood sample \hat{l} from the set of live points and add it to the dead points. The LF-NS algorithm is shown in Algorithm 1.

Algorithm 1: Likelihood-free nested sampling algorithm

- 1: Given observations \mathbf{y} , a prior $\pi(\theta)$ for θ and a likelihood approximation $p(\hat{l}(\theta)|\theta)$.
- 2: Sample N particles $\{\theta^k, \hat{l}^k\}$ from the prior $\Pi(\theta, \hat{l}(\theta))$ and save it in the set \mathcal{L}_0 , set $\mathcal{D} = \{\emptyset\}$
- 3: **for** $i = 1, 2, \dots, m$ **do**
- 4: Find $i' = \arg \min_k \hat{l}^k | \{\theta^k, \hat{l}^k\} \in \mathcal{L}_{i-1}$ and set $\theta_i = \theta^{i'}$ and $\epsilon_i = \hat{l}^{i'}$
- 5: Add $\{\theta_i, \epsilon_i\}$ to \mathcal{D}
- 6: Set $\mathcal{L}_i = \mathcal{L}_{i-1} \setminus \{\theta_i, \epsilon_i\}$
- 7: Sample $\{\theta^*, \hat{l}^* \sim \Pi(\theta, \hat{l}(\theta)) | \hat{l}(\theta) > \epsilon_i$ and add it to \mathcal{L}_i
- 8: **end for**

The parallel version of LF-NS with r parallel processes is analogous to the parallelization of the standard NS algorithm as described in [S2 Appendix](#).

LF-NS is unbiased

As for standard NS, the sampling procedure for LF-NS guarantees that each set of live points \mathcal{L}_i contains N samples uniformly distributed according to the constrained joint prior $\Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon_i)$, thus removing the sample with the lowest likelihood approximation \hat{l}^k results in the same shrinkage of prior volume as the standard NS scheme. The prior volumes $x_i = t^{(i)} x_{i-1}$ now correspond to the volumes of the constraint joint priors $\Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon_i)$ and the resulting weights $w_i = \hat{x}_{i-1} - \hat{x}_i$ can be used, similarly as in [Eq \(2\)](#), to integrate the likelihood l over the constrained prior. Writing the density functions of $\pi(\theta)$ and $p(\hat{l}(\theta) | \theta)$ as $f_\pi(\theta)$ and $f_p(\hat{l}(\theta) | \theta)$ respectively we have

$$\begin{aligned} \sum_i^m w_i \epsilon_i &\approx \int \hat{l}(\theta) d\Pi(\theta, \hat{l}(\theta)) \\ &= \int_{\Omega} \int_0^{\infty} \hat{l}(\theta) f_\pi(\theta) f_p(\hat{l}(\theta) | \theta) d\hat{l}(\theta) d\theta = \int_{\Omega} f_\pi(\theta) \int_0^{\infty} \hat{l}(\theta) f_p(\hat{l}(\theta) | \theta) d\hat{l}(\theta) d\theta \\ &= \int_{\Omega} l(\theta) f_\pi(\theta) d\theta = Z, \end{aligned}$$

where the last equality relies on the unbiasedness of $\hat{l}(\theta)$. While the procedure for LF-NS is very similar to the standard NS algorithm, the new samples θ^* have to be drawn from the constraint joint prior $\Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon)$ instead from the constrained prior $\pi(\theta | l(\theta) > \epsilon)$. In the following we discuss the resulting difficulties and show how to overcome them.

Sampling from the super-level sets of the likelihood

One of the main challenges ([\[54, 55\]](#)) in the classical NS algorithm is the sampling from the prior constrained to higher likelihood regions $\pi(\theta | l(\theta) > \epsilon)$. A lot of effort has been dedicated to finding ways to sample from the constrained prior efficiently, the most popular approaches include slice sampling ([\[37\]](#)) and ellipsoid based sampling ([\[39\]](#)).

In the case of LF-NS, at the i^{th} iteration we are sampling not just a new parameter vector θ^* but also a realization of its likelihood approximation \hat{l}^* from

$$\Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon_i) = \pi(\theta) p(\hat{l}(\theta) | \theta, \hat{l}(\theta) > \epsilon_i). \tag{6}$$

Since it is in general not possible to sample \hat{l}^* from the constraint distribution $p(\hat{l}(\theta^*) | \theta^*, \hat{l}(\theta^*) > \epsilon_i)$ directly, we use rejection sampling. We sample θ^* from the prior $\pi(\theta)$, then sample \hat{l}^* from the unconstrained distribution $p(\hat{l}(\theta^*) | \theta^*)$ and accept the pair (θ^*, \hat{l}^*) only if $\hat{l}^* > \epsilon_i$. While this procedure guarantees that the resulting samples are drawn from [\(6\)](#), the acceptance rate might become very low. Each live set \mathcal{L}_i consists of N pairs (θ^k, \hat{l}^k) distributed according to [\(6\)](#), thus the parameter vectors θ^k in \mathcal{L}_i are distributed according to

$$\theta \sim \int \Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon_i) d\hat{l}(\theta) = \pi(\theta) p(\hat{l}(\theta) > \epsilon_i). \tag{7}$$

We plotted an example of the distributions [\(3\)](#) and [\(7\)](#) for the example of the birth-death

process in Fig 2A. The distribution (7) has usually an infinite support, although in practice the density function of (7) will be close to zero for large areas of the parameter space Ω . Similarly to NS, we propose to use the set \mathcal{L}_i to draw from the areas where (7) is non-zero. Slice sampling methods ([32, 37]) are unfortunately not applicable for LF-NS since they require a way to evaluate its target distribution at each of its samples. We can still use ellipsoid sampling schemes, but unlike in the case of NS where the target distribution $\pi(\theta|l(\theta) > \epsilon)$ has compact support, the target distribution (7) has potentially infinite support framing ellipsoid based sampling approaches rather unfitting. Sampling using MCMC methods (as suggested in [23]) is expected to work even for target distributions with infinite support, but suffers from the known MCMC drawbacks, as they produce correlated samples and might get stuck in disconnected regions.

To account for the smooth shape of (7) we propose to employ a density estimation approach. At each iteration i , we estimate the density $\pi(\theta)p(\hat{l}(\theta) > \epsilon_i)$ from the live points and employ a rejection sampling approach to sample uniformly from the prior on the domain of this approximation. As density estimation technique, we use Dirichlet Process Gaussian Mixture Model (DP-GMM) ([56]). For further details and an illustration of the different sampling schemes see S3 Appendix.

Even though for the presented examples we employ DP-GMM, we note that in theory any sampling scheme that samples uniformly from the prior $\pi(\theta)$ over the support of $\pi(\theta)p(\hat{l}(\theta) > \epsilon_i)$ will work.

A lower bound on the estimator variance

Unlike for NS, for LF-NS, even if at each iteration the proposal particle θ^* is sampled from the support of $\pi(\theta)p(\hat{l}(\theta) > \epsilon_i)$, it will only be accepted with probability $p(\hat{l}(\theta^*) > \epsilon_i)$. This means that depending on the variance of the likelihood estimation $p(\hat{l}(\theta)|\theta)$ and the current likelihood threshold ϵ_i , the acceptance rate for LF-NS will change and with it the computational cost. For an illustration see S4 Appendix.

Due to this possible increase in computational time, it is important to terminate the LF-NS algorithm as soon as possible. We propose to use for the Bayesian evidence estimation not only the dead particles \mathcal{D} , but also the current live points \mathcal{L}_m . This possibility has been already mentioned in other places (for instance in [40, 49, 57]) but is usually not applied, since the contribution of the live particles decreases exponentially with the number of iterations. Since for standard NS each iteration is expected to take the same amount of time, most approaches simply increase the number of iterations to make the contribution of the live particles negligibly small.

The Bayesian evidence can be decomposed as

$$Z = \underbrace{\int_0^{x_m} L(x) dx}_{=: Z_{\mathcal{L}}^m} + \underbrace{\int_{x_m}^1 L(x) dx}_{=: Z_{\mathcal{D}}^m} \quad (8)$$

where x_m is the prior volume for iteration m . The first integral $Z_{\mathcal{L}}^m$ is the part that can be approximated through the N live samples at any given iteration, while the integral $Z_{\mathcal{D}}^m$ is approximated through the dead samples.

Writing

$$\bar{L}_m := \frac{1}{N} \sum_{\{\theta, l\} \in \mathcal{L}_m} \hat{l} \approx \underbrace{\int \hat{l}(\theta) d\Pi(\theta, \hat{l}(\theta) | \hat{l}(\theta) > \epsilon_m)}_{=L_m}$$

for the estimator of the integral of the likelihoods in the live set, we propose the following estimator for Z

$$\hat{Z}_{\text{tot}}^m = \underbrace{\hat{x}_m \bar{L}_m}_{=Z_{\mathcal{L}}^m \approx Z_{\mathcal{L}}^m} + \underbrace{\sum_{i=1}^m \epsilon_i w_i}_{=Z_{\mathcal{D}}^m \approx Z_{\mathcal{D}}^m}, \tag{9}$$

where $\hat{Z}_{\mathcal{D}}^m$ approximates the finite sum $\tilde{Z}_{\mathcal{D}}^m = \sum_{i=1}^m \epsilon_i (x_{i-1} - x_i)$ by replacing the random variables x_i with their means \hat{x}_i . Since $\hat{x}_m \bar{L}_m$ is an unbiased estimator of $Z_{\mathcal{L}}^m$ and $\hat{Z}_{\mathcal{D}}^m$ is an unbiased estimator of $Z_{\mathcal{D}}^m$, the estimator \hat{Z}_{tot}^m is an unbiased estimator of the Bayesian evidence Z for any m . In particular, this implies that terminating the LF-NS algorithm at any iteration m will result in an unbiased estimate for Z . However, terminating the LF-NS algorithm early on will still result in a very high variance of the estimator. This variance is a result of the variances in x_i and the variance in the Monte Carlo estimate \bar{L}_m . As pointed out in [40], when using nested sampling approximations to approximate the integral of arbitrary functions f over the posterior, an additional error is introduced by approximating the average value of $f(\theta)$ on the contour line of $l(\theta) = \epsilon_i$ with the value $f(\theta_i)$. In the following we formulate a lower bound σ_{min}^{2m} on the estimator variance $\sigma_{\text{tot}}^{2m} = \text{Var}(\tilde{Z}_{\mathcal{D}}^m + \hat{Z}_{\mathcal{L}}^m)$ at iteration m , show that this lower bound is monotonically increasing in m and propose to terminate the LF-NS algorithm as soon as the current estimator variance differs from this lower bound by no more than a predefined threshold δ .

Treating the prior volumes x_i and the Monte Carlo estimate \bar{L}_m as random variables, the variance σ_{tot}^{2m} of the NS estimator at iteration m can be estimated at each iteration without additional computational effort (see S5 Appendix and [57]). This variance depends on the variance $\text{Var}(\bar{L}_m)$ of the Monte Carlo estimate \bar{L}_m and is monotonically increasing in $\text{Var}(\bar{L}_m)$ (see S5 Appendix). We define the term σ_{min}^{2m} which is the same variance $\text{Var}(\tilde{Z}_{\mathcal{D}}^m + \hat{Z}_{\mathcal{L}}^m)$ but with the term $\text{Var}(\bar{L}_m)$ set to 0. Clearly we have for any m (see S6 Appendix)

$$\sigma_{\text{tot}}^{2m} \geq \sigma_{\text{min}}^{2m}.$$

More importantly, as we show in S6.2 Appendix, σ_{min}^2 is monotonically increasing in m

$$\sigma_{\text{min}}^{2m'} \geq \sigma_{\text{min}}^{2m}, \quad \forall m' \geq m.$$

This allows us to bound the lowest achievable estimator variance $\sigma_{\text{min}}^2 = \sup_{m \rightarrow \infty} \sigma_{\text{min}}^{2m}$ from below

$$\sigma_{\text{min}}^2 \geq \sigma_{\text{min}}^{2m}.$$

The terms for σ_{tot}^{2m} and σ_{min}^{2m} both contain the unknown value L_m which can be approximated using its Monte Carlo estimate \bar{L}_m giving us the estimations of the above variances $\hat{\sigma}_{\text{tot}}^{2m}$ and

$\hat{\sigma}_{\min}^{2m}$. We use these variance estimates to formulate a termination criteria by defining

$$\Delta_{LFNS}^m := \frac{\sqrt{\hat{\sigma}_{\text{tot}}^{2m}} - \sqrt{\hat{\sigma}_{\min}^{2m}}}{\hat{Z}_{\text{tot}}^m} \quad (10)$$

and terminate the algorithm as soon as $\Delta_{LFNS}^m < \delta$ for some predefined δ . The term in the numerator in (10) is the difference between the current estimator standard deviation $\hat{\sigma}_{\text{tot}}^m$ and the lowest standard deviation $\hat{\sigma}_{\min}^m$ that can possibly be achieved by continuing to run the LF-NS algorithm beyond the current iteration m . The denominator is used to normalized this possible reduction in standard deviation by the current evidence estimate \hat{Z}_{tot}^m . This termination criteria seems intuitive since it terminates the LF-NS algorithm as soon as a continuation of the algorithm is not expected to make the final estimator significantly more accurate. As a final remark we note that the final estimator \hat{Z}_{tot}^m as well as the termination criteria using Δ_{LFNS}^m can of course also be applied in the standard NS case.

Results

We test our proposed LF-NS algorithm on three examples for stochastic reaction kinetic models. The first example is the birth death model, already introduced above, the second example is the Lac-Gfp model used for benchmarking in [10] and the third example is a transcriptional model from [58] with corresponding real data. We also compare the performance of the LF-NS algorithm with two of the most widely used likelihood-free inference methods pMCMC and ABC-SMC. We point out that for deterministic systems with available likelihoods, our LF-NS algorithm reduces to the standard NS method and has been discussed in several other places (see for instance [41]). In the following examples all priors are chosen as uniform or log-uniform in the bounds indicated in the posterior plots.

The stochastic birth-death model

We first revisit the birth-death example from above to compare our inference results to the solution obtained by FSP. We use the same data as above and use the same log-uniform prior. We ran our LF-NS algorithm as described above using DP-GMM for the sampling. We used $N = 100$ LF-NS particles, $H = 100$ particle filter particles and sample at each iteration $r = 10$ particles. We ran the LF-NS algorithm until Δ_{LFNS}^m is smaller than 0.001. We show the obtained posterior in Fig 3A. Fig 3B shows the obtained estimates of the Bayesian evidence, where the shaded areas indicate the standard error at each iteration. The dashed red line indicates the true BE computed from 10^6 samples from $\Pi(\theta, \hat{l}(\theta))$. We observe that due to the simplicity of the birth-death model, the BE is already well estimated in the very first iteration. However, the estimates of the lower and upper bound of the variance $\hat{\sigma}_{\min}^{2m}$ and $\hat{\sigma}_{\text{tot}}^{2m}$ indicate that a continuation of LF-NS scheme may result in a lower estimator variance. The development of these upper and lower bound estimators are shown in Fig 3C and we can clearly see how they converge to the same value. For our termination criteria we show the quantities $\Delta_{\text{max}}^m = x_m \max(\hat{l} \in \mathcal{L}_m)$, that is frequently used as a termination criteria for regular nested sampling (see S1.3 Appendix) and Δ_{LFNS}^m in Fig 3D.

The Lac-Gfp model

We demonstrate how our algorithm deals with a realistic sized stochastic model, by inferring the posterior for the parameters of the Lac-Gfp model illustrated in S5 Fig. This model has been already used in [10] as a benchmark, although with distribution-data. Here we use the

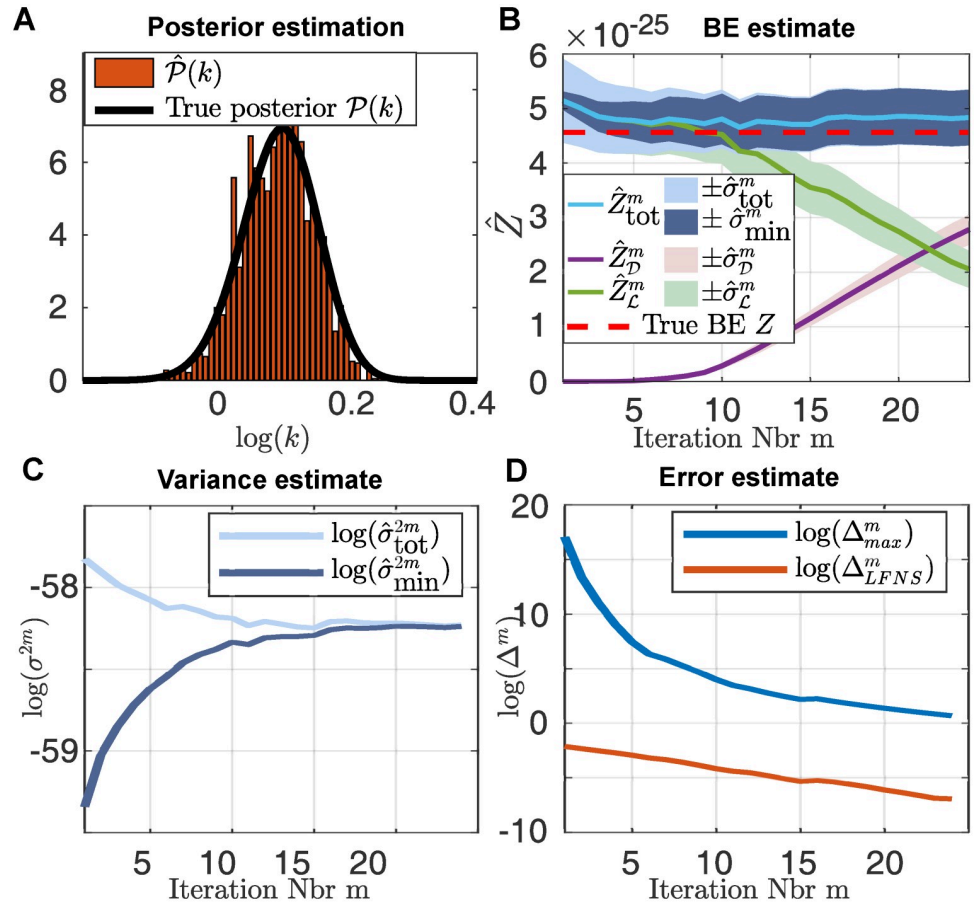


Fig 3. Inference on the birth-death model. **A:** Histogram of the posterior $\mathcal{P}(k)$ estimate obtained with LF-NS using $N = 100$ and $H = 100$. The true posterior is indicated in black. **B:** Development of the estimation of the Bayesian evidence using the estimation based solely on the dead points \hat{Z}_D , the estimate approximation from the live points \hat{Z}_L and the estimation based on both \hat{Z}_{tot} . The corresponding standard errors are indicated as the shaded areas. The true Bayesian evidence is indicated with the dashed red line. **C:** Estimate of the current variance estimate $\hat{\sigma}_{tot}^{2m}$ and the lower bounds for the lowest achievable variance $\hat{\sigma}_{min}^2$. **D:** Developments of the different error estimations for each iteration.

<https://doi.org/10.1371/journal.pcbi.1008264.g003>

model to simulate a number of trajectories and illustrate how our approach infers the posterior of the used parameters. This model is particularly challenging in two aspects. First, the number of parameters is 18, making it a fairly large model to infer. Secondly, the model exhibits switch-like behaviour which makes it very hard to approximate the likelihood of such a switching trajectory (see S7.2 Appendix and particular S6 Fig for further details). We used $N = 500$ LF-NS particles, $H = 500$ particle filter particles and sample at each iteration $r = 50$ particles.

The measured species in this example is fluorescent Gfp (mGFP) where it is assumed that each Gfp-molecule emits fluorescence according to a normal distribution. We used one trajectory to infer the posterior, whose marginals are shown in Fig 4D. The posterior covers the true parameters (indicated in blue) and we can observe that while several parameters (particularly $\theta_1 - \theta_7$) cannot be well identified by the posterior, the remaining parameters seem to have been identified well. Fig 4A shows the estimated Bayesian evidence with corresponding standard errors for each iteration. Fig 4B shows the corresponding estimations of the bounds of the lowest achievable variance. As we see, the estimated Bayesian evidence, as well as the estimated variance bounds, do several jumps in the process of the LF-NS run. These jumps are

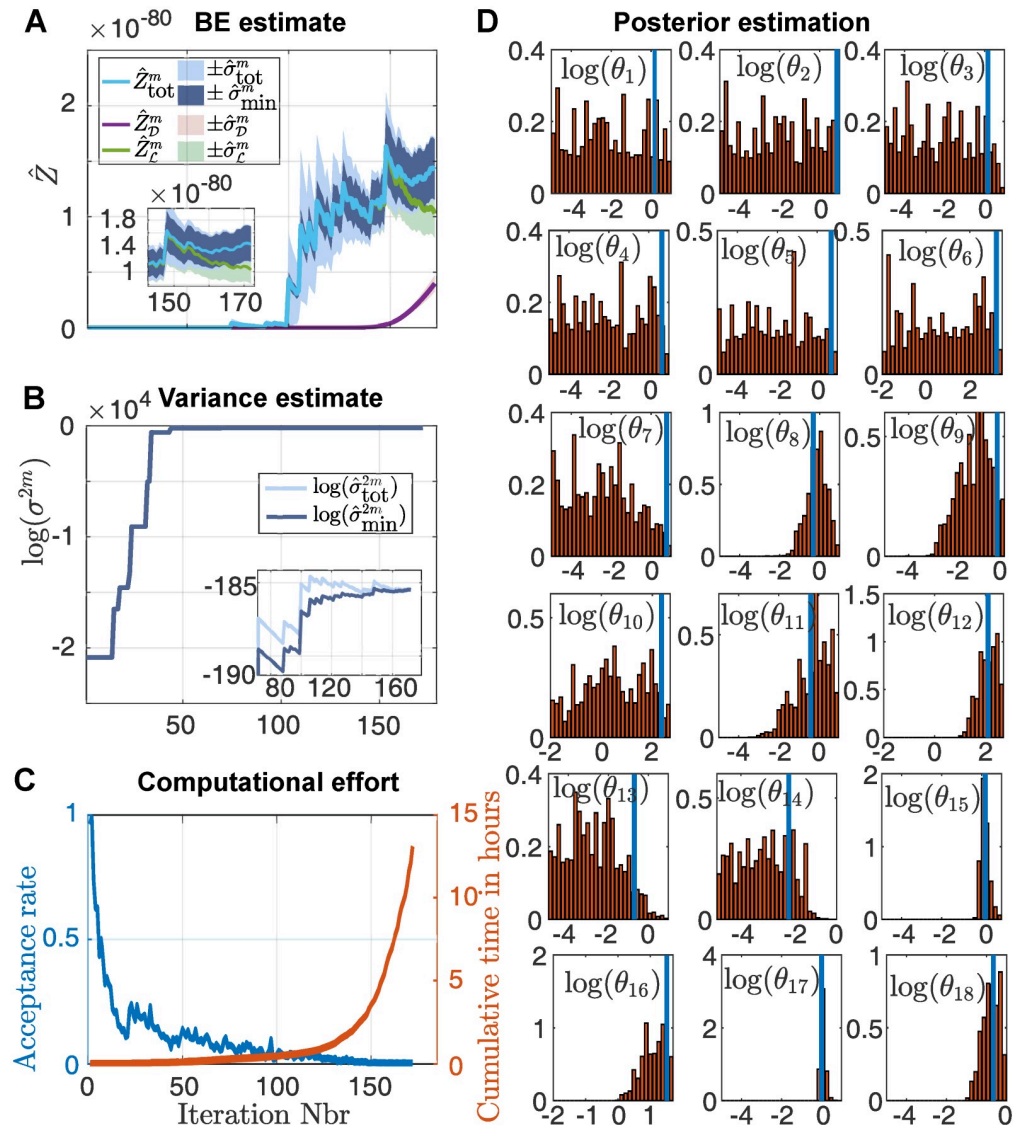


Fig 4. Inference on the Lac-Gfp model. **A:** Development of the estimation of the Bayesian evidence using the estimation based solely on the dead points \hat{Z}_D^m , the estimate approximation from the live points \hat{Z}_L^m and the estimation that uses both \hat{Z}_{tot}^m . The corresponding standard errors are indicated as the shaded areas. **B:** Estimate of the current variance estimate $\hat{\sigma}_{tot}^{2m}$ and the lower bounds for the lowest achievable variance $\hat{\sigma}_{min}^{2m}$. **C:** The acceptance rate of the LF-NS algorithm for each iteration (blue) and the cumulative time needed for each iteration in hours (red). The computation was performed on 48 cores in parallel on the Euler cluster of the ETH Zurich. **D:** Marginals of the inferred posterior distributions of the parameters based on one simulated trajectory. The blue lines indicate the parameters used for the simulation of the data.

<https://doi.org/10.1371/journal.pcbi.1008264.g004>

due to the exploration of the parameter space and correspond to iterations in which previously unsampled areas of the parameter space got sampled with a new maximal likelihood. In Fig 4C we plotted the acceptance rate of the LF-NS algorithm for each iteration as well as the cumulative computational time. The computation was performed on 48 cores of the Euler cluster of the ETH Zurich. The inference for this model took well over 12 hours and as we see, the computational time for each iteration seems to increase exponentially as the acceptance rate decreases. The low acceptance rate is expected, since the number of particle filter particles

$H = 500$ results in a very high variance of the particle filter estimate (see S6B Fig). From Fig 4A we can clearly see that using only the dead points for the BE estimation would result in a very strong underestimation of the true BE. The use of the live sets and our developed termination criteria is what allows us to terminate the algorithm in a reasonable time. We can also see from Fig 4A that our LF-NS algorithm seems to approximate the BE already quite well after around 110 iterations. To reach this iteration the LF-NS needed only approximately 35 minutes (as can be seen in Fig 4C), and we conclude that the majority of the computational time is used not to find new regions of high likelihood, but rather to reduce the variance of the final BE estimator. We point out that we could have terminated the algorithm after the first hour and would have had already a good estimate of the BE, where the final variance (as seen in Fig 4B) would still be in an acceptable range. The runtime of around 13 hours is therefore mainly due to our very high standards of accuracy.

A stochastic transcription model

As a third example we use a transcription model recently used in [58], where an optogenetically inducible transcription system is used to obtain live readouts of nascent RNA counts. The model consists of a gene that can take two configurations “on” and “off”. In the “on” configuration mRNA is transcribed from this gene and can be individually measured during this transcription process (see [58] for details). We modelled the transcription through $n = 8$ subsequent RNA species that change from one to the next at a rate λ . This is done to account for the observed time of 2 minutes that one transcription event takes. An illustration of the model is shown in Fig 5A. For the inference of the model parameters we chose five trajectories of real biological data, shown in Fig 5C. Clearly, the system is inherently stochastic and requires corresponding inference methods. We ran the LF-NS algorithm for $N = 500$ and $H = 500$ on these five example trajectories. The resulting marginal posteriors are shown in Fig 5B, we also indicated the parameter ranges considered in [58]. These ranges were chosen in [58] in an ad-hoc manner but, apart from the values for k_{off} seem to fit very well with our inferred results. To make sure that our approach inferred the right posterior, we also ran a pMCMC algorithm (see S8.1 Appendix) on the problem. pMCMC methods have been shown to target the true posterior distribution, which is why we chose the obtained pMCMC posterior as ground truth. To make sure that the pMCMC run converged in an acceptable time, we used the posterior obtained from the LF-NS run to pick the initial sample of pMCMC. We also picked the perturbation kernel q for pMCMC to be a Gaussian with covariance equal to the covariance matrix of the obtained LF-NS posterior. We ran the pMCMC algorithm for 24 hours and indicated the obtained marginal posteriors in blue. As can be seen, our obtained LF-NS posterior fits very well with the posterior obtained through pMCMC. In S7 Fig we show the development of the evidence approximation as well as the corresponding standard errors and the development of the upper and lower bound estimation for the lowest achievable variance σ_{min}^2 .

Comparison with other likelihood-free approaches

To further evaluate and compare our LF-NS algorithm with other state-of-the-art likelihood-free approaches, we compare it to two of the most popular current methods in this field, particle MCMC (pMCMC) and Approximate Bayesian Computation—Sequential Monte Carlo (ABC-SMC), that have both been the subject of several recent review papers [27–29].

The pMCMC method (as presented for instance in [46]) is in its core a Metropolis-Hastings MCMC (MH-MCMC) method, where the likelihood in the acceptance ratio is replaced with its particle filter approximation (for more details see S8.1 Appendix). It therefore inherits many of the strengths and weaknesses of the MH-MCMC method. Among its strengths is that

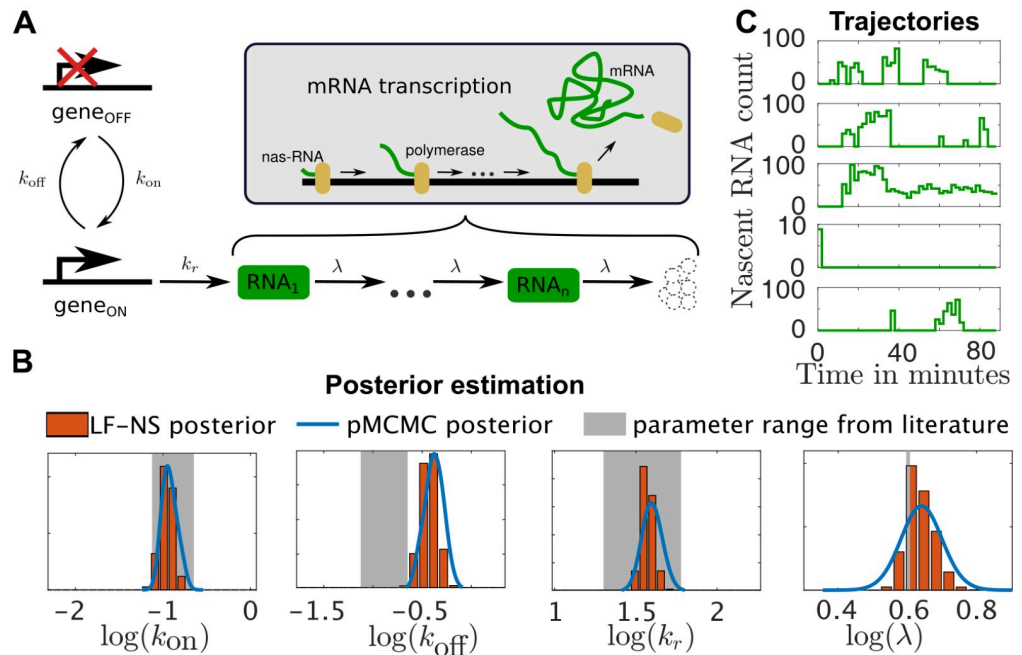


Fig 5. Inference on the transcription model. **A:** Schematic representation of the gene expression model. The model consists of a gene that switches between an “on” and an “off” state with rates k_{on} and k_{off} . When “on” the gene is getting transcribed at rate k_r . The transcription process is modelled through n RNA species that sequentially transform from one to the next at rate λ . The observed species are all of the intermediate RNA_i species. **B:** The marginal posterior distribution of the parameters of the system. The histogram indicates the posterior obtained through LF-NS, the blue line indicates the posterior obtained from a long pMCMC run and the shaded areas indicate the parameter ranges that were considered in [58]. The scale of the x-axis does not represent the range of the prior and has been chosen for presentational purposes. **C:** The five trajectories used for the parameter inference.

<https://doi.org/10.1371/journal.pcbi.1008264.g005>

it targets the true posterior $\mathcal{P}(\theta)$, is easy to implement and has been very well studied. The downsides of pMCMC include that for realistic problems its performance depends on the initial sample θ_0 of the Markov Chain as well as the choice of the perturbation kernel q that is used to create the Markov Chain. Further, as pMCMC produces a sequence of dependent samples it is rather difficult to parallelize and can get stuck in local regions of high-likelihood. Like our LF-NS method, the pMCMC method is the result of using an established inference method (for pMCMC it is MH-MCMC and in the case of LF-NS it is nested sampling) and replacing the likelihood computation with a particle filter approximation. Like the LF-NS method, pMCMC requires the unbiasedness of the particle filter approximation to guarantee the convergence to the true posterior.

Unlike pMCMC, ABC methods do not target the true posterior $p(\theta|y)$, but instead aim at obtaining an approximate posterior $p(\theta|d(y_\theta, y) < \epsilon)$, where y_θ is the simulated dataset using parameter θ , y is the experimental data, d is some chosen distance and $\epsilon > 0$ is some pre-defined threshold. ABC-SMC ([51]) further improves on this idea, as it creates the distribution $p(\theta|d(y_\theta, y) < \epsilon)$ by constructing a sequence of intermediate distributions $p(\theta|d(y_\theta, y) < \epsilon_i)$, corresponding to a decreasing sequence of $\epsilon_0 > \dots > \epsilon_F = \epsilon$. In each iteration i , ABC-SMC samples a set of N particles from the distribution obtained in the previous iteration and accepts the new particles if $d(y_\theta, y) < \epsilon_i$ (see for details [20, 51] and S8.2). This sequential approach is very similar to our LF-NS algorithm and differs from it mainly in the way the intermediate distributions are defined. The ABC-SMC methods create them so that the distance $d(y_\theta, y)$ lies

under a certain threshold ϵ_i , while the LF-NS algorithm requires the likelihood approximation $\hat{l}(\theta)$ to be above a certain threshold ϵ_i .

The advantages of ABC-SMC include that, unlike pMCMC methods they create uncorrelated samples and can therefore be easily parallelized, are not in danger of getting stuck in isolated parameter regions and do not require the specification of a good initial sample. Additionally, the computation of the distance $d(y_\theta, y)$ is usually much cheaper than running a particle filter. The obvious downsides of ABC-SMC methods is that they do not target the true posterior and depend on the definition of the distance d as well as a suitable sequence of thresholds $\epsilon_1, > \dots > \epsilon_F$. For further discussions on pMCMC and ABC methods we refer the reader to the above mentioned review papers [27–29] and [S8 appendix](#) in the supplementary.

To give an idea of the performance of these algorithms compared to our LF-NS algorithm, we implemented both the pMCMC algorithm as well as the ABC-SMC method and used them to infer the posterior distribution of the parameters of the Lac-Gfp model. As discussed earlier, this model is particularly challenging due to its large parameter space and its switch-like behavior. We ran the pMCMC method for 48 hours on the same machine as the LF-NS algorithm. To make the comparison as favorable as possible to the pMCMC method, we used the previously obtained LF-NS posterior for the Lac-Gfp model to tune the parameters of the pMCMC run. We chose as perturbation kernel q , a Gaussian distribution with the covariance being the sample covariance matrix of the posterior distribution of the previously obtained posterior. We used the same number of particle filter particles $H = 500$ as for the LF-NS run. We performed a total of two runs, one where the initial sample θ_0 was chosen randomly from the prior $\pi(\theta)$ and one where it was sampled from the posterior $\mathcal{P}(\theta|y)$ previously obtained with LF-NS. The resulting posteriors, as well as the detailed pMCMC runs are shown in [S8](#) and [S9](#) Figs. We observed that when choosing the initial sample θ_0 from the true posterior, the pMCMC method converges to the true posterior. While we ran the pMCMC algorithm sequentially for 48 hours, this runtime can be improved by combining various approaches from the literature such as running several parallel MCMC or parallelizing the particle filter approximation. However, these modifications take a significant effort to implement and tune, and a thorough discussion of all possible options is beyond the scope of this simple comparison. When sampling θ_0 from the prior distribution $\pi(\theta)$ (and not from the previously obtained posterior $\mathcal{P}(\theta|y)$), we observed that even after 48 hours the pMCMC method did not converge to the parameter region of the posterior and thus failed to give any meaningful parameter estimate (see [S8](#) and [S9](#) Figs). We conclude that the pMCMC method is generally capable of inferring the parameter posterior for models of size and complexity comparable to the Lac-Gfp model, but requires extensive tuning and an initial sample that is already in a high likelihood region. We also point out that in this example we used as the perturbation kernel the covariance of the actual posterior distribution. In a realistic setting, this perturbation kernel and the initial sample need to be determined without any knowledge of the true posterior, which proves to be considerably more difficult. Further, it is difficult to know when the pMCMC algorithm reaches the posterior distribution without the knowledge of the true distribution. This can be seen from [S9 Fig](#), where the pMCMC with initial sample taken from the prior $\theta_0 \sim \pi(\theta)$ seems to reach a region of high likelihood, but is in truth far away from true posterior. This issue of tuning the pMCMC method is well known and discussed in the literature. For a further discussion on the tuning problem see for instance [27, 28].

We also ran the ABC-SMC method for the Lac-Gfp model, using as distance metric the euclidean distance between the simulated dataset y_θ and experimental data y . As in [27] we created the sequence of ϵ_i by picking it to be the 30% quantile of the distances computed in the previous iteration. Due to its easy parallelization we ran the ABC-SMC algorithm in parallel

on 48 cores for 24 hours. The resulting marginal posteriors are shown in S10 Fig. We observe that for many of the parameters (θ_1 - θ_7 , θ_{10} , θ_{15} - θ_{18}), the inferred marginal posteriors correspond to the marginal posteriors obtained with the LF-NS method. For several other parameters (θ_8 , θ_9 , θ_{11} - θ_{14}) the inferred distributions do not seem to match the posterior distribution inferred with LF-NS. This can be due to several reasons. The ABC-SMC method may not have been run for a sufficiently long time or the final ϵ_F was not small enough. This result is not surprising, as the ABC-SMC method is not expected to return the true posterior. However, if the true posterior is not of interest and the modeler wishes to only get an idea about the likely location of the model parameters, the ABC-SMC method provides a viable option.

We also compared our LF-NS method with previously published results on the inference for the Lotka-Volterra model (see S7.4 Appendix) that was performed in [27]. There, the authors compared the development of the pMCMC and ABC-SMC methods on this example. We used our LF-NS algorithm to infer the posterior for the same problem. First, we compare the obtained posteriors for the three methods. As ground truth we take the posterior obtained from a very long (12 hours) pMCMC run. We ran all three algorithms sequentially for the same amount of time (12 minutes). The obtained posteriors are shown in S11 Fig. We can clearly see that the LF-NS and the pMCMC algorithms target the true posterior distribution. The ABC-SMC algorithm, while still identifying the true parameters θ^* , seem to infer a distribution slightly different than the posterior distribution.

To compare the computational efficiency of the three algorithms, we followed [27] and used the LF-NS, the pMCMC and the ABC-SMC algorithm to infer the posteriors for the Lotka-Volterra model while limiting ourselves to comparable computational effort. We first ran the LF-NS algorithm with different number of live particles ($N = 10, 20, 40, 60, 80, 100, 200$) using as termination criteria $\Delta_{LFNS}^m = 0.01$. The runs were performed sequentially (rather than in parallel) to make the results more comparable between the three algorithms. We ran the ABC-SMC algorithm for the same problem using the same number of particles as in the LF-NS runs. As the ABC-SMC algorithm does not provide any termination criteria, we terminate it after the same runtime as the LF-NS algorithm. We also performed several pMCMC runs, one with initial sample from the posterior distribution and one with the initial sample from the prior distribution. We ran each of the pMCMC runs for the same amount of time as the LF-NS runs took and plotted the obtained results in Fig 6.

In Fig 6A we show the development of the Bayesian evidence estimate for each of the LF-NS runs, as well as the estimator variance. Fig 6B shows the estimated marginal posterior

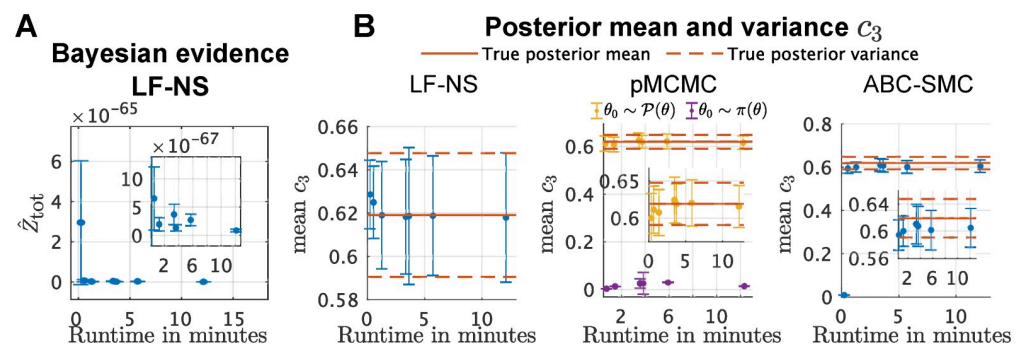


Fig 6. Runtime comparison between LF-NS, pMCMC and ABC. A: The estimated Bayesian evidence for the different LF-NS runs. The error bars indicate the standard deviation of the final BE estimate. B: The estimated mean and standard deviation of the marginal posterior for c_3 for the different algorithm runs. The solid red line indicates the mean and standard deviation of the true marginal posterior for c_3 .

<https://doi.org/10.1371/journal.pcbi.1008264.g006>

means and variance for the parameter c_3 . The only algorithm that clearly failed at obtaining the true parameter mean and variance is the pMCMC run where θ_0 was picked from the prior. We also see that the ABC-SMC algorithm gets very close to the true parameter mean, but consistently underestimates it, which is a strong indicator that the targeted distribution is not the true posterior. For the LF-NS algorithm as well as the pMCMC algorithm with θ_0 picked from the posterior, we see that every run results in an estimate very close to the true mean and variance. For the Lotka-Volterra model, the runtime of the pMCMC as well as the inferred posterior distribution depends on the initial sample. When θ_0 is picked close enough to high posterior regions, the convergence is very quick, while when picked from the prior, the convergence can take very long. The speed of the converges will in general also depend on the used perturbation kernel q and can be greatly improved by tuning the pMCMC algorithm further. The LF-NS algorithm captures the posterior for all runtimes reliably, where larger runtimes result in a lower estimator variance. Unlike pMCMC, the LF-NS algorithm as well as the ABC-SMC algorithm both explore the full parameter space and then converge to the (approximate) posterior. Further LF-NS and ABC-SMC can easily be parallelized which will greatly improve the runtime in practical problems.

Discussion

We have introduced a likelihood-free formulation of the well known nested sampling algorithm and have shown that it is unbiased for any unbiased likelihood estimator. While the utilization of NS for systems without an available likelihood is straight forward, one has to take precautions to avoid infeasibly high computational times. Unlike for standard NS, it is crucial to include the estimation of the live samples to the final BE estimation as well as terminate the algorithm as soon as possible. We have shown how using a Monte Carlo estimate over the live points not only results in an unbiased estimator of the Bayesian evidence Z , but also allows us to derive a formulation for a lower bound on the achievable variance in each iteration. This lower bound at each iteration has been shown to be a lower bound for the best achievable variance and has allowed us to formulate a novel termination criterion that stops the algorithm as soon as a continuation can at best result in an insignificant improvement in accuracy. While the formulation of the variances and its lower bound were derived having a parallel LF-NS scheme in mind, they can equally well be used in the standard NS case and can be added effortlessly to already available toolboxes such as [36] or [37]. We emphasize that the lower variance bound approximation $\hat{\sigma}_{\min}^{2m}$ is neither a strict error term, as it only gives information of the variance of the estimator, nor a strict lower bound of the estimator variance since it contains the unknown term L_m . Instead, it gives an estimate of the lowest achievable estimator variance that depends on the Monte Carlo estimate of the likelihood average over the live points \bar{L}_m . This can be seen Fig 4A and S7A Fig, where the lower bound estimate $\hat{\sigma}_{\min}^{2m}$ does not only make jumps, but also decreases after each jump (the actual lower bound estimate σ_{\min}^{2m} is monotonically increasing in m as shown in S6.2 Appendix). Our suggested LF-NS scheme has three different parameters that govern the algorithm behaviour. The number of LF-NS particles N determines how low the minimal variance of the estimator can get, where low values for N result in a rather high variance and high values for N result in a lower variance. The number of particles for the particle filter H determines how wide or narrow the likelihood estimation is and thus determines the development of the acceptance rate of the LF-NS run, while the number of LF-NS iterations determines how close the variance of the final estimate comes to the minimal variance. We have demonstrated the applicability of our method on several models with simulated as well as real biological data.

We also compared the performance of the LF-NS scheme with the most widely used other likelihood-free inference schemes in the field, pMCMC and ABC-SMC. We have shown that for the considered example of the Lac-Gfp model, the ABC-SMC method failed to recover the desired posterior in any reasonable time, while the pMCMC algorithm did recover the true posterior, but only after extensive tuning and after a much longer run time than our proposed method. Further we have compared the LF-NS method with the performance of pMCMC and ABC-SMC on the Loktka-Volterra model and have demonstrated that on this example, the computational effort as well as the accuracy of the results compares very well with the other two state-of-the-art methods. Both the issues with ABC-SMC as well as the extensive tuning with pMCMC are well known and discussed in the literature. Further, we have introduced a clear termination criteria for the LF-NS method that allows to terminate the algorithm as soon as no improvement of accuracy can be expected.

We believe that our method, while in its individual elements similar to pMCMC and ABC-SMC, combines the strengths of each approach while circumventing the crucial tuning required by pMCMC and the issue of targeting an approximate rather than the true posterior distribution of the ABC-SMC.

Our LF-NS can, similarly to ABC, pMCMC or SMC models deal with stochastic models with intractable likelihoods and has all of the advantages of classic NS. We believe that particularly the variance estimation that can be performed from a single LF-NS run proves to be useful as well as the straight forward parallelization.

Supporting information

S1 Appendix. Nested sampling.

(PDF)

S2 Appendix. Parallelization schemes.

(PDF)

S3 Appendix. Sampling from the super-level set.

(PDF)

S4 Appendix. Accuracy of the likelihood approximation \hat{l} .

(PDF)

S5 Appendix. Estimating the variance for the parallel LF-NS scheme.

(PDF)

S6 Appendix. Lower bound on the variance $\text{Var}(\eta_C^{m,r} + \eta_D^{m,r})$.

(PDF)

S7 Appendix. Examples used.

(PDF)

S8 Appendix. Comparison with other methods for likelihood-free Bayesian inference.

(PDF)

S1 Fig. Illustration of the nested sampling approximation with a uniform prior on $[0, 1]$.

A: The integral over the parameter space $\int_{\Omega} l(\theta) d\theta$. B: The transformed integral $\int_0^1 L(x) dx$ over the prior volume x .

(PDF)

S2 Fig. Variance of parallel vs serial LF-NS scheme. A: Values for \tilde{Z}_D^{ij} for LF-NS run. The shaded areas indicate the standard error. B: The standard deviation of \tilde{Z} .
(PDF)

S3 Fig. Comparison of different sampling schemes for the super-level sets of the likelihood. A: Contour lines of the distribution of $\pi(\theta)p(\hat{l}(\theta) > \epsilon)$ for the birth death example, where $\theta = \{k, \gamma\}$, the number of particle filter particles to approximate $\hat{l}(\theta)$ is $H = 20$ and $\log(\epsilon) = -118.75$. The density was approximated with 10^6 samples. The red dots indicate 90 samples in \mathcal{L} . B: The estimations of \mathcal{L} as obtained through an ellipsoid estimation, kernel density estimation (KDE) and Dirichlet process Gaussian mixture models (DP-GMM) based on the samples in \mathcal{L} . C: 2000 samples from the corresponding estimations of \mathcal{L} where the samples for KDE were obtained according to (3.2) and the samples from DP-GMM were obtained using rejection sampling as described in [S3 Appendix](#).
(PDF)

S4 Fig. Illustration of likelihood approximation using particle filter. A: Top: Likelihood for different parameters k (red) and contour lines of the joint distribution $\Pi(k, \log(k))$ of the parameter k and its likelihood approximation $\hat{l}(k)$, based on 10^6 samples of the likelihood approximation obtained with a particle filter with 100 particles. Bottom: The constrained priors $\pi(k|l(k) > \epsilon)$ and $\pi(k)p(\hat{l}(k) > \epsilon|k)$ for $\epsilon = 1e - 24$. B: Acceptance rates $\alpha(m)$ and Z_D^m from 10^6 samples of $\Pi(k, \hat{l}(k))$ for the birth death model for different values of particle filter particles H and different iteration numbers m .
(PDF)

S5 Fig. The Lac-Gfp model. Schematic of the Lac-Gfp Model where the final measurement is the mature GFP (mGFP) and the input is IPTG (assumed to be constant $10\mu\text{M}$).
(PDF)

S6 Fig. Lac-Gfp trajectories and likelihood approximation. A: 3 example trajectories of simulated Lac-Gfp data, measured at 29 time points. B: Log likelihood approximation for the real parameter θ^* for the first trajectory with different number of particles H for the particle filter.
(PDF)

S7 Fig. Evidence and variance estimation for the transcription model. A: Development of the estimation of the Bayesian evidence using the estimation based solely on the dead points \hat{Z}_D , the estimate approximation from the live points \hat{Z}_L and the estimation that uses both \hat{Z}_{tot} . The corresponding standard errors are indicated as the shaded areas. B: Estimate of the current variance estimate $\hat{\sigma}_{\text{tot}}^{2m}$ and the lower bounds for the lowest achievable variance $\hat{\sigma}_{\text{min}}^2$.
(PDF)

S8 Fig. Posteriors obtained from the pMCMC and ABC-SMC runs. The true parameter θ^* is indicated as the blue line. The posterior obtained from the LF-NS run (as described in the main paper) is plotted as thick black line. The posteriors obtained from the two pMCMC runs are plotted in red (θ_0 sampled from the posterior) and yellow (θ_0 sampled from the prior).
(PDF)

S9 Fig. pMCMC runs for the Lac-Gfp model. pMCMC development of the individual parameters from the two pMCMC runs with different initial samples θ_0 **Left:** pMCMC run with $\theta_0 \sim \mathcal{P}(\theta)$ **Right:** pMCMC run with $\theta_0 \sim \pi(\theta)$.
(PDF)

S10 Fig. Distributions $p(\theta|d(y_\theta|y) < \epsilon_T)$ obtained from ABC-SMC. The true parameter θ^* is indicated as the blue line. The marginal posterior obtained from the LF-NS run (as described in the main paper) is plotted as thick black line. The distributions $p(\theta|d(y_\theta|y) < \epsilon_F)$ are shown in thin lines.

(PDF)

S11 Fig. Posteriors for the Lotka-Volterra model obtained from LF-NS, pMCMC and ABC-SMC. The obtained posteriors for all three algorithms, LF-NS, pMCMC and ABC-SMC are indicated. The posteriors were obtained using 12 minutes of computation for all three algorithms. The thick black line indicates the posterior as obtained with a long run of pMCMC.

(PDF)

S12 Fig. Detailed performance of each LF-NS run. Left: Evidence using the estimation based solely on the dead points \hat{Z}_D , the estimate approximation from the live points \hat{Z}_L and the estimation based on both \hat{Z}_{tot} . The corresponding standard errors are indicated as the shaded areas. Right: Acceptance rate and cumulative runtime for each iteration.

(PDF)

S1 Table. Species and initial numbers of the Lac-Gfp model.

(PDF)

S2 Table. Prior distributions of the Lac-GFP parameters and the real values θ^* used for the simulation of y .

(PDF)

S3 Table. Prior distributions of the parameters for the transcription model.

(PDF)

S4 Table. Species and initial numbers of the Lotka-Volterra model.

(PDF)

S5 Table. Prior distributions of the parameters for the Lotka-Volterra model.

(PDF)

Author Contributions

Conceptualization: Jan Mikelson, Mustafa Khammash.

Funding acquisition: Mustafa Khammash.

Methodology: Jan Mikelson.

Software: Jan Mikelson.

Writing – original draft: Jan Mikelson.

Writing – review & editing: Jan Mikelson, Mustafa Khammash.

References

1. Ingram PJ, Stumpf MP, Stark J. Network motifs: structure does not determine function. *BMC genomics*. 2006; 7(1):108. <https://doi.org/10.1186/1471-2164-7-108> PMID: 16677373
2. Elowitz MB, Levine AJ, Siggia ED, Swain PS. Stochastic gene expression in a single cell. *Science*. 2002; 297(5584):1183–1186. <https://doi.org/10.1126/science.1070919> PMID: 12183631
3. McAdams HH, Arkin A. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences*. 1997; 94(3):814–819. <https://doi.org/10.1073/pnas.94.3.814>

4. Paulsson J, Berg OG, Ehrenberg M. Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences*. 2000; 97(13):7148–7153. <https://doi.org/10.1073/pnas.110057697>
5. Samoilov M, Plyasunov S, Arkin AP. Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences*. 2005; 102(7):2310–2315. <https://doi.org/10.1073/pnas.0406841102>
6. Ko CH, Yamada YR, Welsh DK, Buhr ED, Liu AC, Zhang EE, et al. Emergence of noise-induced oscillations in the central circadian pacemaker. *PLoS biology*. 2010; 8(10):e1000513. <https://doi.org/10.1371/journal.pbio.1000513> PMID: 20967239
7. Munsky B, Trinh B, Khammash M. Listening to the noise: random fluctuations reveal gene network parameters. *Molecular systems biology*. 2009; 5(1). <https://doi.org/10.1038/msb.2009.75> PMID: 19888213
8. Boys RJ, Wilkinson DJ, Kirkwood TB. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*. 2008; 18(2):125–135. <https://doi.org/10.1007/s11222-007-9043-x>
9. Hilfinger A, Paulsson J. Separating intrinsic from extrinsic fluctuations in dynamic biological systems. *Proceedings of the National Academy of Sciences*. 2011; 108(29):12167–12172. <https://doi.org/10.1073/pnas.1018832108>
10. Lillacci G, Khammash M. The signal within the noise: efficient inference of stochastic gene regulation models using fluorescence histograms and stochastic simulations. *Bioinformatics*. 2013; 29(18):2311–2319. <https://doi.org/10.1093/bioinformatics/btt380> PMID: 23821649
11. Neuert G, Munsky B, Tan RZ, Teytelman L, Khammash M, van Oudenaarden A. Systematic identification of signal-activated stochastic gene regulation. *Science*. 2013; 339(6119):584–587. <https://doi.org/10.1126/science.1231456> PMID: 23372015
12. Zechner C, Unger M, Pelet S, Peter M, Koepl H. Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings. *Nature methods*. 2014; 11(2):197–202. <https://doi.org/10.1038/nmeth.2794> PMID: 24412977
13. Gillespie DT. Exact stochastic simulation of coupled chemical reactions. *The journal of physical chemistry*. 1977; 81(25):2340–2361. <https://doi.org/10.1021/j100540a008>
14. Golightly A, Wilkinson DJ. Bayesian inference for Markov jump processes with informative observations. arXiv preprint arXiv:14094362. 2014.
15. Stathopoulos V, Girolami MA. Markov chain Monte Carlo inference for Markov jump processes via the linear noise approximation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2013; 371(1984):20110541. <https://doi.org/10.1098/rsta.2011.0541>
16. Doucet A, De Freitas N, Gordon N. An introduction to sequential Monte Carlo methods. In: *Sequential Monte Carlo methods in practice*. Springer; 2001. p. 3–14.
17. Cappé O, Godsill SJ, Moulines E. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*. 2007; 95(5):899–924. <https://doi.org/10.1109/JPROC.2007.893250>
18. Golightly A, Wilkinson DJ. Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*. 2011; 1(6):807–820. <https://doi.org/10.1098/rsfs.2011.0047> PMID: 23226583
19. Pitt MK, dos Santos Silva R, Giordani P, Kohn R. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*. 2012; 171(2):134–151. <https://doi.org/10.1016/j.jeconom.2012.06.004>
20. 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; 6(31):187–202. <https://doi.org/10.1098/rsif.2008.0172>
21. Jagiella N, Rickert D, Theis FJ, Hasenauer J. Parallelization and High-Performance Computing Enables Automated Statistical Inference of Multi-scale Models. *Cell Systems*. 2017. <https://doi.org/10.1016/j.cels.2016.12.002> PMID: 28089542
22. Ionides EL, Bretó C, King AA. Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*. 2006; 103(49):18438–18443. <https://doi.org/10.1073/pnas.0603181103>
23. Skilling J, et al. Nested sampling for general Bayesian computation. *Bayesian analysis*. 2006; 1(4):833–859. <https://doi.org/10.1214/06-BA127>
24. Mukherjee P, Parkinson D, Liddle AR. A nested sampling algorithm for cosmological model selection. *The Astrophysical Journal Letters*. 2006; 638(2):L51. <https://doi.org/10.1086/501068>
25. Feroz F, Hobson M, Cameron E, Pettitt A. Importance nested sampling and the MultiNest algorithm. arXiv preprint arXiv:13062144. 2013.

26. Wilkinson DJ. Stochastic modelling for systems biology. Chapman and Hall/CRC; 2006.
27. Owen J, Wilkinson DJ, Gillespie CS. Likelihood free inference for Markov processes: a comparison. *Statistical applications in genetics and molecular biology*. 2015; 14(2):189–209. <https://doi.org/10.1515/sagmb-2014-0072> PMID: 25720092
28. Warne DJ, Baker RE, Simpson MJ. A practical guide to pseudo-marginal methods for computational inference in systems biology. *Journal of theoretical biology*. 2020; p. 110255. <https://doi.org/10.1016/j.jtbi.2020.110255> PMID: 32223995
29. Warne DJ, Baker RE, Simpson MJ. Simulation and inference algorithms for stochastic biochemical reaction networks: from basic concepts to state-of-the-art. *Journal of the Royal Society Interface*. 2019; 16(151):20180943. <https://doi.org/10.1098/rsif.2018.0943>
30. Gillespie DT. The chemical Langevin equation. *The Journal of Chemical Physics*. 2000; 113(1):297–306. <https://doi.org/10.1063/1.481811>
31. Elf J, Ehrenberg M. Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome research*. 2003; 13(11):2475–2484. <https://doi.org/10.1101/gr.1196503> PMID: 14597656
32. Aitken S, Akman OE. Nested sampling for parameter inference in systems biology: application to an exemplar circadian model. *BMC systems biology*. 2013; 7(1):72. <https://doi.org/10.1186/1752-0509-7-72> PMID: 23899119
33. Burkoff NS, Várnai C, Wells SA, Wild DL. Exploring the energy landscapes of protein folding simulations with Bayesian computation. *Biophysical journal*. 2012; 102(4):878–886. <https://doi.org/10.1016/j.bpj.2011.12.053> PMID: 22385859
34. Dybowski R, McKinley TJ, Mastroeni P, Restif O. Nested sampling for bayesian model comparison in the context of salmonella disease dynamics. *PloS one*. 2013; 8(12):e82317. <https://doi.org/10.1371/journal.pone.0082317> PMID: 24376528
35. Pullen N, Morris RJ. Bayesian Model Comparison and Parameter Inference in Systems Biology Using Nested Sampling. *PloS one*. 2014; 9(2):e88419. <https://doi.org/10.1371/journal.pone.0088419> PMID: 24523891
36. Feroz F, Hobson M, Bridges M. MultiNest: an efficient and robust Bayesian inference tool for cosmology and particle physics. *Monthly Notices of the Royal Astronomical Society*. 2009; 398(4):1601–1614. <https://doi.org/10.1111/j.1365-2966.2009.14548.x>
37. Handley W, Hobson M, Lasenby A. POLYCHORD: next-generation nested sampling. *Monthly Notices of the Royal Astronomical Society*. 2015; 453(4):4384–4398. <https://doi.org/10.1093/mnras/stv1911>
38. Johnson R, Kirk P, Stumpf MP. SYSBIONS: nested sampling for systems biology. *Bioinformatics*. 2015; 31(4):604–605. <https://doi.org/10.1093/bioinformatics/btu675> PMID: 25399028
39. Feroz F, Hobson M. Multimodal nested sampling: an efficient and robust alternative to Markov Chain Monte Carlo methods for astronomical data analyses. *Monthly Notices of the Royal Astronomical Society*. 2008; 384(2):449–463. <https://doi.org/10.1111/j.1365-2966.2007.12353.x>
40. Higson E, Handley W, Hobson M, Lasenby A, et al. Sampling errors in nested sampling parameter estimation. *Bayesian Analysis*. 2018. <https://doi.org/10.1214/17-BA1075>
41. Murray I. *Advances in Markov chain Monte Carlo methods*. Citeseer; 2007.
42. Liphardt T. *Efficient computational methods for sampling-based metabolic flux analysis*. ETH Zurich; 2018.
43. Henderson RW, Goggans PM. Parallelized nested sampling. In: *AIP Conference Proceedings*. vol. 1636. AIP; 2014. p. 100–105.
44. Skilling J. Nested sampling's convergence. In: *AIP Conference Proceedings*. vol. 1193. AIP; 2009. p. 277–291.
45. MacKay DJ. *Information theory, inference and learning algorithms*. Cambridge university press; 2003.
46. Wilkinson DJ. Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology. In: *Proceedings of 9th Valencia International Meeting on Bayesian Statistics*; 2010. p. 679–705.
47. Andrieu C, Doucet A, Holenstein R. Particle Markov chain Monte Carlo for efficient numerical simulation. In: *Monte Carlo and quasi-Monte Carlo methods 2008*. Springer; 2009. p. 45–60.
48. Robert CP, Wraith D. Computational methods for Bayesian model choice. In: *AIP Conference Proceedings*. vol. 1193. AIP; 2009. p. 251–262.
49. Chopin N, Robert CP. Properties of nested sampling. *Biometrika*. 2010; 97(3):741–755. <https://doi.org/10.1093/biomet/asq021>

50. Marjoram P, Molitor J, Plagnol V, Tavaré S. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*. 2003; 100(26):15324–15328. <https://doi.org/10.1073/pnas.0306899100>
51. Sisson SA, Fan Y, Tanaka MM. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*. 2007; 104(6):1760–1765. <https://doi.org/10.1073/pnas.0607208104>
52. Del Moral P. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*. 1996; 2(4):555–581.
53. Munsky B, Khammash M. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of chemical physics*. 2006; 124(4):044104. <https://doi.org/10.1063/1.2145882> PMID: 16460146
54. Chopin N, Robert C. *Contemplating evidence: properties, extensions of, and alternatives to nested sampling*. Citeseer; 2007.
55. Brewer BJ, Pártay LB, Csányi G. Diffusive nested sampling. *Statistics and Computing*. 2011; 21(4):649–656. <https://doi.org/10.1007/s11222-010-9198-8>
56. Görür D, Rasmussen CE. Dirichlet process Gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*. 2010; 25(4):653–664. <https://doi.org/10.1007/s11390-010-9355-8>
57. Keeton CR. On statistical uncertainty in nested sampling. *Monthly Notices of the Royal Astronomical Society*. 2011; 414(2):1418–1426. <https://doi.org/10.1111/j.1365-2966.2011.18474.x>
58. Rullan M, Benzinger D, Schmidt GW, Miliás-Argeitis A, Khammash M. An Optogenetic Platform for Real-Time, Single-Cell Interrogation of Stochastic Transcriptional Regulation. *Molecular cell*. 2018; 70(4):745–756. <https://doi.org/10.1016/j.molcel.2018.04.012> PMID: 29775585