Efficient Parameter Estimation for DNA Kinetics Modeled as Continuous-Time Markov Chains

Sedigheh Zolaktaf[1(✉)], Frits Dannenberg[2], Erik Winfree[2], Alexandre Bouchard-Côté[1], Mark Schmidt[1], and Anne Condon[1]

[1] University of British Columbia, Vancouver, BC, Canada
nasimzf@cs.ubc.ca
[2] California Institute of Technology, Pasadena, CA, USA

Abstract. Nucleic acid kinetic simulators aim to predict the kinetics of interacting nucleic acid strands. Many simulators model the kinetics of interacting nucleic acid strands as continuous-time Markov chains (CTMCs). States of the CTMCs represent a collection of secondary structures, and transitions between the states correspond to the forming or breaking of base pairs and are determined by a nucleic acid kinetic model. The number of states these CTMCs can form may be exponentially large in the length of the strands, making two important tasks challenging, namely, mean first passage time (MFPT) estimation and parameter estimation for kinetic models based on MFPTs. Gillespie's stochastic simulation algorithm (SSA) is widely used to analyze nucleic acid folding kinetics, but could be computationally expensive for reactions whose CTMC has a large state space or for slow reactions. It could also be expensive for arbitrary parameter sets that occur in parameter estimation. Our work addresses these two challenging tasks, in the full state space of all non-pseudoknotted secondary structures of each reaction. In the first task, we show how to use a reduced variance stochastic simulation algorithm (RVSSA), which is adapted from SSA, to estimate the MFPT of a reaction's CTMC. In the second task, we estimate model parameters based on MFPTs. To this end, first, we show how to use a generalized method of moments (GMM) approach, where we minimize a squared norm of moment functions that we formulate based on experimental and estimated MFPTs. Second, to speed up parameter estimation, we introduce a fixed path ensemble inference (FPEI) approach, that we adapt from RVSSA. We implement and evaluate RVSSA and FPEI using the Multistrand kinetic simulator. In our experiments on a dataset of DNA reactions, FPEI speeds up parameter estimation compared to inference using SSA, by more than a factor of three for slow reactions. Also, for reactions with large state spaces, it speeds up parameter estimation by more than a factor of two.

C. Thachuk and Y. Liu (Eds.): DNA 25, LNCS 11648, pp. 80–99, 2019.
https://doi.org/10.1007/978-3-030-26807-7_5

1 Introduction

Nucleic acid kinetic simulators [9,29,34,35] aim to predict the kinetics of interacting nucleic acid strands, such as the rate of a reaction or the sequence of interactions between the strands. These simulators are desirable for building nucleic acid-based devices whose nucleic acid sequences need to be carefully designed to control their behaviour. For example, neural networks can be realized in DNA using strand displacement reactions [6]. However, the rates of reactions vary by several orders of magnitude depending on sequence and conditions and are hard to predict, making the design of artifacts challenging. Accurate kinetic simulators would allow many, though not all, unanticipated design flaws to be identified prior to conducting wet-lab experiments, and would allow more complex molecular devices to be designed and successfully implemented with fewer deficiencies needing to be debugged experimentally.

Because of these pressing needs, there has been great progress on simulators that can model the kinetics of interacting nucleic acid strands. The simulators range from coarse-grained models that consider large rearrangements of the base pairs [34,35], and often factor in tertiary structure, to elementary step models that consider the forming or breaking of a single base pair [9,29], and to molecular dynamics models that follow the three-dimensional motion of the polymer chains [27,31]. Elementary step models are of interest to us here because they are computationally more efficient than molecular dynamics, yet they also can represent and thus discover unexpected sequence-dependent secondary structures within intermediate states. Continuous-time Markov chains (CTMCs) play a central role in modeling nucleic acid kinetics with elementary steps, such as Kinfold [9] and Multistrand [28,29]. States of the CTMCs correspond to secondary structures and have exponentially distributed holding times, and transitions between states correspond to forming or breaking of a single base pair. Nucleic acid kinetic models [24,42], along with nucleic acid thermal stability models [2,17,38,39], specify the rate of transition between states and the holding time of states. These simulators can stochastically sample paths (sequences of states from an initial to a target state) and trajectories (sequences of states from an initial to a target state, along with the times to transition between successive states). The mean first passage time (MFPT) from an initial to a target state can be estimated from sampled trajectories. The first passage time of a trajectory is the first time that the trajectory occupies the target state. Kinetic rates, such as the rate constant of a reaction [28], can then be derived from such estimates.

Our work addresses two challenging tasks in accurately predicting the MFPT of a reaction's CTMC, in the full state space of all non-pseudoknotted[1] secondary structures. The first task is to estimate the MFPT of a reaction's CTMC, given a calibrated kinetic model. The second task is to calibrate parameters of kinetic

[1] A pseudoknot is a secondary structure that has at least two base pairs in which one nucleotide of a base pair is intercalated between the two nucleotides of the other base pair.

models; even though thermal stability models are well calibrated [3,23], parameters of kinetic models, which affect the rate of transition between states and consequently holding times of states, are not well calibrated [42]. These tasks are challenging, particularly for multistranded DNA kinetics, because when nucleic acid strands interact, they are prone to the formation of many metastable secondary structures due to stochastic formation and breakage of base pairs. The number of possible secondary structures nucleic acids can form may be exponentially large compared to the number of nucleotides the strands contain. Moreover, to make accurate estimations, many sampled trajectories might be required, which might be time-consuming to obtain (see Sect. 4). In this work, we make progress on these tasks, by focusing on the Multistrand kinetic simulator [28,29] (described in Sect. 2.1), that is used to analyze the folding kinetics of multiple interacting nucleic acid strands and models the kinetics as CTMCs with elementary steps. In the rest of this section, first, we describe related work for MFPT estimation and our contributions. Then, we describe related work for calibrating kinetic models based on MFPTs and our contributions.

1.1 Mean First Passage Time Estimation

Exact linear algebra methods [33,42] can provide an exact solution to the MFPT of a CTMC that can be explicitly represented. However, their accuracy could be compromised by numerical errors and it is infeasible to use these methods for CTMCs with large implicitly-represented state spaces. Our previous work [42] estimates MFPTs on heuristically obtained reduced state spaces of the CTMCs. Moreover, the state spaces are customized for each type of reaction. In contrast to exact linear algebra methods, the MFPT could be approximated in the full state space or reduced state space with Gillespie's stochastic simulation algorithm (SSA) [11]. SSA can be slow depending on the CTMC of the reaction. We could adapt sequential Monte Carlo and importance sampling techniques [8,13], but these methods require a proposal distribution that efficiently reduces the variance of the estimator. More recently, machine learning algorithms have been developed to successfully predict DNA hybridization rates [41] from sequence, without enumerating the state space of the reaction. However, these methods can not treat other reactions or kinetics.

Our Contribution. We show how to use a reduced variance stochastic simulation algorithm (RVSSA), a Rao-Blackwellised version [20] of SSA, to estimate the MFPT of a reaction's CTMC. In SSA, the variance of MFPT estimates arises for two reasons. First, the path to a target state affects the MFPT. Second, the holding time in each state affects the MFPT. RVSSA removes the stochasticity in the holding times by using expected holding times of states. We prove that RVSSA produces a lower variance estimator of the MFPT compared to SSA. Moreover, we show in our experiments that RVSSA has a lower variance than SSA in estimating the MFPT of a reaction's CTMC, when in the sampled paths there exist a small number of states that have large expected holding times compared to other states. One interesting example that we identify is the association of poly(A) and poly(T) sequences in low concentrations (see Fig. 2b).

1.2 Parameter Estimation

In order to make accurate MFPT estimations, the underlying parameters of the CTMCs should be calibrated. Models of nucleic acid thermal stability [2, 17,38,39] have been extensively calibrated to experimental data [3,23]. However, extensive calibration of nucleic acid kinetic models remains challenging [42]. Our previous work [42] uses a maximum a posteriori approach and a Markov chain Monte Carlo approach to calibrate DNA kinetic models on a wide range of reactions, such as strand displacement [40], but on reduced state spaces of the reactions. The reduced state spaces are manually designed and coded and the largest reduced state space contains less than 1.5×10^4 states. Moreover, related work [32,40] uses reaction specific models to calibrate a kinetic model. These models are not easily adapted to other kinetic models. There have been advances in calibrating CTMCs [1,21,30] based on transient probabilities, the likelihood that a process will be in a given state at a given time, and these advances have been used for calibration of nucleic acid kinetics [13] and chemical reaction networks [10,12,19,22].

During the optimization, for every new parameter set, we could use SSA or RVSSA to obtain an unbiased estimate of the MFPT of a reaction's CTMC. However, sampling new trajectories for every new parameter set could be computationally expensive for large CTMCs, slow reactions, or arbitrary parameter sets. One reason is that transitions might be repeatedly sampled. We could also use importance sampling techniques [8,13], but these methods would require a proposal distribution that efficiently reduces the variance of the estimator, which is challenging when the underlying transition probabilities of the CTMC are changing throughout parameter estimation.

Our Contribution. To estimate parameters for DNA kinetics modeled as CTMCs based on MFPTs, we show how to use a generalized method of moments (GMM) [14] estimator. More importantly, we show how to use a fixed path ensemble inference (FPEI) approach that speeds up parameter estimation compared to a reference method that uses SSA directly during inference (SSAI). The GMM method is widely used in econometrics and has also recently been used in chemical reaction networks [22]. The GMM method can be used when a maximum likelihood estimate or a maximum a posteriori estimate is infeasible, as is the case with CTMCs that have very large state spaces. The GMM method minimizes a weighted norm of moment conditions obtained from samples. The moment conditions are functions of model parameters and the dataset such that the expectation of the moment conditions is zero at the true value of the parameters. To minimize the squared norm of the moment conditions, we use the Nelder-Mead direct-search optimization algorithm [26], which has been frequently used in optimization problems that have small stochastic perturbations in function values [4].

To speed up parameter estimation, we introduce and use FPEI. In this method, we condense paths, where for every path, we compute the set of states and the number of times each state is visited. Rather than generating new trajectories with SSA for every parameter set variation (the SSAI method), in FPEI

we use fixed condensed paths to speed up parameter estimation. For example, in this work, the length of the longest path is more than 1×10^8, whereas the number of unique states and transitions of the path is approximately 3.8×10^5 and 1.4×10^6, respectively. In FPEI, we use RVSSA to estimate the MFPT of the fixed paths given a new parameter set. Since the MFPT estimates obtained with fixed paths are biased, we alternate between minimizing the error of prediction on fixed paths, and resampling new paths and restarting the optimization method.

To implement RVSSA and FPEI, we augment the Multistrand kinetic simulator [28,29] where for each reaction the full state space of all non-pseudoknotted secondary structures is possible. We conduct computational experiments on experimental DNA reactions that have moderate or large state spaces or are slow, including hairpin closing, hairpin opening, helix dissociation with and without mismatches, and helix association. We compare the performance of RVSSA with SSA for MFPT estimation and FPEI with SSAI for parameter estimation. Results for our example data are encouraging, showing that FPEI speeds up parameter estimation compared to using SSAI, by more than a factor of three for slow reactions. Also, for reactions with large state spaces, it speeds up parameter estimation by more than a factor of two.

2 Preliminaries

2.1 The Multistrand Kinetic Simulator

The Multistrand kinetic simulator [28,29] models the kinetics of multiple interacting nucleic acid strands as a CTMC. A state of the CTMC represents a system microstate, in other words, the configuration of the strands in the fixed volume that we simulate. A system microstate is a collection of complex microstates. A complex microstate is a set of strands connected by base pairing (secondary structures). In Multistrand, all possible secondary structures are permitted except for pseudoknots. Multistrand defines the energy of a state as the sum of the standard free energy for each complex, which is determined with Multistrand's nucleic acid thermal stability model. Transitions between states correspond to the forming or breaking of a single base pair. For example, in Fig. 1, state t can transition to states s and u. The transition rate k_{ts} from state t to state s is determined by the energy of the states and a nucleic acid kinetic model.

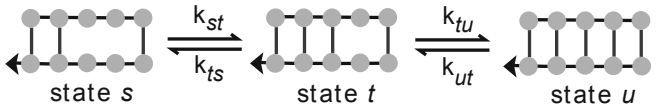


Fig. 1. State t can transition to states s and u by breaking or forming a single base pair, respectively. The reverse transitions are also possible.

We experiment with the Metropolis [24] and the Arrhenius [42] kinetic models that are implemented in the Multistrand software. The Metropolis kinetic model has two free parameters k_{uni} and k_{bi} that distinguish between unimolecular and bimolecular transitions, respectively. In the Arrhenius kinetic model, transition rates additionally depend on the local context of the base pair that is forming or breaking. The model differentiates between seven different half contexts $\mathcal{C} = \{\text{stack}, \text{loop}, \text{end}, \text{stack+loop}, \text{stack+end}, \text{loop+end}, \text{stack+stack}\}$. For example, in Fig. 1, in the transition from state t to state s, the half contexts of the base pair that is breaking are a stack and a loop. An Arrhenius rate constant A_l and an activation energy E_l are associated with each half context l. The model also has a bimolecular scaling constant α. In total, the model has 15 free parameters.

To sample paths and trajectories for a reaction, experimental conditions need to be determined, such as the sequence of the strands, the temperature, the concentration of Na^+ and Mg^{2+} cations, and the initial concentration of the strands. We adopt the trajectory mode of Multistrand for all reactions of our dataset. In this mode, SSA is implemented to simulate trajectories over the CTMC of a reaction, starting in an initial state and halting when the reaction is over, and to estimate the MFPT. For helix association and hairpin closing reactions, all trajectories start from the state where no base pairs are formed and end at the state where the duplex is fully formed. For hairpin opening and helix dissociation the start and end states are reversed. Given the estimated MFPT $\hat{\tau}^r$ of reaction r, as computed over several trajectories, the reaction rate constant of reaction r is computed as

$$\hat{k}^r = \begin{cases} \frac{1}{\hat{\tau}^r} & \text{first order reaction} \\ \frac{1}{\hat{\tau}^r u^r} & \text{second order reaction} \end{cases}, \tag{1}$$

where u^r is the initial concentration of the reactants of reaction r in the simulation [28]. Equation (1) also holds for the experimental reaction rate constant and the experimental MFPT, called timescale, of the reaction.

2.2 Gillespie's Stochastic Simulation Algorithm

Gillespie's stochastic simulation algorithm (SSA) [11] has been widely used to simulate stochastic trajectories in CTMCs [28,29]. It provides an unbiased and consistent estimate of the MFPT from an initial state to a target state. It estimates the MFPT as the mean of the first passage times of sampled trajectories. In brief, to sample a trajectory and its first passage time, SSA advances forward in two steps:

1. At a jump from the current state s_i, SSA samples the holding time T_i of the state from an exponential distribution with a rate equal to the sum of the transition rates from the state, in other words, $T_i \mid s_i \sim \text{Exp}(k_{s_i})$, where $k_{s_i} = \sum_{s \in \mathcal{S}} k_{s_i s}$, $\mathcal{S}$ is the state space of the CTMC, $k_{s_i s}$ is the transition rate from state s_i to state s, if s is not a neighbor of s_i then $k_{s_i s} = 0$, $\mathbb{E}[T_i \mid s_i] = k_{s_i}^{-1}$ and $\text{Var}(T_i \mid s_i) = k_{s_i}^{-2}$.

2. At a jump from the current state s_i, SSA samples the next state s_{i+1} from the outgoing transition probabilities of state s_i, in other words, $p(s_i, s) = \frac{k_{s_i s}}{k_{s_i}}, s_i \neq s$.

Let P be a trajectory of length Z from state s to state t, with holding times $T_1, ..., T_{Z-1}$, obtained by using SSA with initial state s, and ending the first time that state t is sampled. In SSA, the FPT of the trajectory is computed as

$$F^{\text{SSA}} = \sum_{i=1}^{Z-1} T_i. \tag{2}$$

By using N independently sampled trajectories, we obtain a Monte Carlo estimator for the MFPT of the CTMC as $\hat{\tau}_N^{\text{SSA}} = \frac{1}{N} \sum_{n=1}^{N} F_n^{\text{SSA}}$.

3 Methodology

3.1 Mean First Passage Time Estimation

In SSA, the variance of MFPT estimates arises for two reasons. First, the path to a target state affects the MFPT. Second, the holding time in each state affects the MFPT. Hordijk et al. [18] show how to obtain a reduced variance estimate of a steady-state measure of an irreducible and positive recurrent CTMC. Their constant holding-time method eliminates the variability in the random holding time of states and instead uses expected holding times. To estimate the MFPT of a reaction's CTMC, we formulate a Rao-Blackwellised version [20] of SSA, which similar to Hordijk et al. also eliminates the variability in the random holding times of states. However, the CTMC is not restricted to be irreducible or positive recurrent and the MFPT estimate is not necessarily a steady-state measure. We call this method the reduced variance stochastic simulation algorithm (RVSSA). Similar to SSA, RVSSA also produces a consistent and unbiased estimator of the MFPT, but has a smaller variance in predicting MFPTs compared to SSA[2].

In brief, in RVSSA, instead of sampling a random holding time for each state, we use an estimator based on the expected holding time. The algorithm proceeds as follows.

1. At a jump from the current state s_i, compute the expected holding time $\overline{T}_i$ before jumping to the next state, in other words, $\overline{T}_i = k_{s_i}^{-1} = (\sum_{s \in \mathcal{S}} k_{s_i s})^{-1}$. Note that $\mathbb{E}[\overline{T}_i \mid s_i] = k_{s_i}^{-1}$ and $\text{Var}(\overline{T}_i \mid s_i) = 0$.
2. Step 2 of SSA.

[2] For our purpose here, we are only interested in the MFPT, so the smaller variance is good. In other contexts, the full distribution of FPTs will be of interest, and for that purpose only SSA, but not RVSSA, will be appropriate.

Let P be a path of length Z from state s to state t, with expected holding times $\overline{T}_1, ..., \overline{T}_{Z-1}$, obtained by using RVSSA with initial state s, and ending the first time that state t is sampled. In RVSSA, we compute the MFPT of the path as

$$Y^{\text{RVSSA}} = \sum_{i=1}^{Z-1} \overline{T}_i. \tag{3}$$

By using N independently sampled paths, we obtain a Monte Carlo estimator for the MFPT of the CTMC as $\hat{\tau}_N^{\text{RVSSA}} = \frac{1}{N} \sum_{n=1}^{N} Y_n^{\text{RVSSA}}$.

Theorem 1. *The estimator of the MFPT from state s to state t produced by RVSSA has a lower variance than the estimator produced by SSA.*

Proof. Let P denote a random path from state s to state t. We have $\mathbb{E}[F^{\text{SSA}} \mid P] = \mathbb{E}[Y^{\text{RVSSA}} \mid P]$, and consequently

$$\text{Var}(\mathbb{E}[F^{\text{SSA}} \mid P]) = \text{Var}(\mathbb{E}[Y^{\text{RVSSA}} \mid P]). \tag{4}$$

Also, $\mathbb{E}[\text{Var}(F^{\text{SSA}} \mid P)] > 0$, and $\mathbb{E}[\text{Var}(Y^{\text{RVSSA}} \mid P)] = \mathbb{E}[\text{Var}(\sum_{i=1}^{Z-1} \overline{T}_i \mid P)] = 0$ because $\overline{T}_i$ are constants and independent given P. Based on the law of total variance

$$\begin{aligned} &\text{Var}(Y^{\text{RVSSA}}) = \mathbb{E}[\text{Var}(Y^{\text{RVSSA}} \mid P)] + \text{Var}(\mathbb{E}[Y^{\text{RVSSA}} \mid P]) \overset{\text{by Eq. (4)}}{=} \\ &\mathbb{E}[\text{Var}(Y^{\text{RVSSA}} \mid P)] + \text{Var}(\mathbb{E}[F^{\text{SSA}} \mid P]) = \text{Var}(\mathbb{E}[F^{\text{SSA}} \mid P]) < \\ &\mathbb{E}[\text{Var}(F^{\text{SSA}} \mid P)] + \text{Var}(\mathbb{E}[F^{\text{SSA}} \mid P]) = \text{Var}(F^{\text{SSA}}). \end{aligned} \tag{5}$$

Therefore, it can be concluded that $\text{Var}(\hat{\tau}_N^{\text{RVSSA}}) = \text{Var}(\frac{1}{N} \sum_{n=1}^{N} Y_N^{\text{RVSSA}}) = \frac{1}{N}\text{Var}(Y^{\text{RVSSA}}) < \frac{1}{N}\text{Var}(F^{\text{SSA}}) = \text{Var}(\hat{\tau}_N^{\text{SSA}})$. □

For an unbiased estimator, the expected mean squared error (MSE) of the estimator is equal to the variance of the estimator [36]. Consequently, RVSSA has a smaller MSE than SSA and requires fewer sampled paths to estimate the MFPT,

$$\mathbb{E}[(\hat{\tau}_N^{\text{RVSSA}} - \tau)^2] = \frac{1}{N}\text{Var}(Y^{\text{RVSSA}}) < \frac{1}{N}\text{Var}(F^{\text{SSA}}) = \mathbb{E}[(\hat{\tau}_N^{\text{SSA}} - \tau)^2]. \tag{6}$$

3.2 Parameter Estimation

In Sect. 3.1, we assume that the underlying parameters of the CTMCs are known. Here, we focus on estimating the underlying parameters of the transition rates when they are not known a priori.

To estimate model parameters, we formulate a generalized method of moments (GMM) [14] objective function based on experimental and predicted MFPTs. The GMM estimators have desirable statistical properties under suitable conditions, such as consistency and asymptotic normality. The GMM method minimizes a weighted norm of moment conditions. The moment conditions are functions of model parameters and observed values such that the

expectation of the moment conditions is zero at the true value of the parameters. Given a column vector $\mathbf{g}$ of moment conditions and its transpose $\mathbf{g}^{\mathrm{T}}$, the GMM method seeks the true parameter set θ^* as

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathbf{g}(\theta)^{\mathrm{T}}\mathbf{W}\mathbf{g}(\theta), \tag{7}$$

where $\mathbf{W}$ is a positive-definite matrix that controls the variance of the estimator. For optimally chosen weights, which depend on the covariance of the moment conditions at the true parameter set θ^*, the estimator has the smallest possible variance for the parameters. Since the true parameter set is unknown, there exist several approaches to deal with this issue. For example, the two-step GMM estimator [15] uses the identity matrix in the first step to estimate a parameter set. In the second step, it uses the estimated parameters to produce the weighting matrix and reestimates the parameters. In our experiments, we only use the identity weighting matrix, which produces a consistent and asymptotic normal GMM estimator, and leave other options to future work.

Let θ be a parameter set for a kinetic model that parameterizes the CTMC of reactions, and let θ^* be the true parameter set. For reaction r, based on the experimental MFPT τ^r and an unbiased estimator of the MFPT $\hat{\tau}^r$, we can define a moment condition as $g^r(\theta) = \hat{\tau}^r(\theta) - \tau^r$. However, since reactions occur at timescales that cover many orders of magnitude, from slow reactions, such as helix dissociation, to faster reactions, such as hairpin closing, and since we are using an identity matrix, we use $\log_{10}$ differences instead; we define a moment condition as

$$g^r(\theta) = \log_{10}\hat{\tau}^r(\theta) - \log_{10}\tau^r, \tag{8}$$

where we approximate $\mathbb{E}[g^r(\theta^*)] = \mathbb{E}[\log_{10}\hat{\tau}^r(\theta^*)] - \log_{10}\tau^r \approx 0$ for the true parameter set θ^* (if one exists). This approximation is reasonable for an unbiased and low variance estimator of the experimental MFPT τ^r. The Taylor expansion of $\mathbb{E}[\log_{10}\hat{\tau}^r(\theta^*)]$ around $\log_{10}\mathbb{E}[\hat{\tau}^r(\theta^*)] = \log_{10}\tau^r$ is $\mathbb{E}[\log_{10}\hat{\tau}^r(\theta^*)] \approx \mathbb{E}\left[\log_{10}\tau^r + \frac{1}{\tau^r}(\hat{\tau}^r - \tau^r) - \frac{1}{2(\tau^r)^2}(\hat{\tau}^r - \tau^r)^2\right] = \log_{10}\tau^r - \frac{\mathrm{Var}(\hat{\tau}^r(\theta^*))}{2(\tau^r)^2}$, where the second term disappears. Also, note that based on Eq. (1), instead of Eq. (8) we equivalently use $g^r(\theta) = \log_{10}\hat{\tau}^r(\theta) - \log_{10}\tau^r = \log_{10}k^r - \log_{10}\hat{k}^r(\theta)$, which is commonly used in related work [41,42]. Based on the entire reactions of the dataset $\mathcal{D}$, we define the GMM estimator as

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{r\in\mathcal{D}} \left(\log_{10}k^r - \log_{10}\hat{k}^r(\theta)\right)^2. \tag{9}$$

This can be recognized as the least mean squared error (MSE) parameter set.

In our experiments (described in Sect. 4.3), we seek a parameter set that minimizes the GMM estimator. However, we also considered using the negative of Eq. (14) from our previous work [42], where $g^r(\theta)$ is defined to be normally distributed with an unbiased mean and variance σ^2, and a small $L2$ regularization term is also defined. With this objective function, the predictive quality of the fixed path ensemble inference (FPEI) approach, which we describe later on, only slightly changes for our dataset.

To minimize the objective function, we use the Nelder-Mead direct-search optimization algorithm [26]. To approximate a local optimum parameter set θ with size $|\theta|$, the algorithm maintains a simplex of $|\theta| + 1$ parameter sets. The algorithm evaluates the function value at every parameter set of the simplex. It proceeds by attempting to replace a parameter set that has the worst function value with a parameter set reflected through the centroid of the remaining $|\theta|$ parameter sets in the simplex with expansion and contraction as needed. The algorithm uses only the ranks of the function values to determine the next parameter set, and therefore has been frequently used in optimization problems that have small stochastic perturbations in function values [4]. This robustness is essential for its use in SSAI.

In SSAI, during the optimization, to obtain an unbiased estimate of τ^r for every parameter set variation, we use SSA. However, obtaining new trajectories for every parameter set is computationally expensive. One reason is that transitions might be repeatedly sampled. Therefore the length of a trajectory could be much larger than the number of unique states and transitions of the trajectory (see Sect. 4.3). We propose to use FPEI which uses an ensemble of fixed paths, with an efficient data structure, to speed up parameter estimation. In FPEI, for every reaction, we build a fixed set of paths with an initial parameter set θ_0. For a new parameter set θ_m, we use the fixed paths to estimate the MFPT. To speed up computations, we condense paths; for every path, we compute the set of states and the number of times each state is visited. We compute the holding time of a state in a path as if the path is regenerated in the full state space. To compute the holding time of a state under a new parameter set, we need to compute the total outgoing transition rate from the state under the new parameter set. Therefore, we also store information about the outgoing neighbors of the states that affect the outgoing transition rate. Alternatively, depending on memory and storage limitations, similar to SSA and RVSSA, we could repeatedly compute the outgoing neighbors of the states on the fly. Given this data, as the parameter set is updated to θ_m, we compute the MFPT of path P according to RVSSA as

$$Y^{\mathrm{FPEI}}(\theta_m) = \sum_{i=1}^{Z-1} \overline{T}_i(\theta_m), \text{ where } \overline{T}_i(\theta_m) = \frac{1}{k_{s_i}(\theta_m)}, \tag{10}$$

where the transition rates of the CTMC depend on the parameter set θ_m and the path is obtained with θ_0. Because of the condensed representation, this formula is not literally computed, but rather a mathematically equivalent one with fewer terms is computed. Given N fixed paths obtained with θ_0, we estimate the MFPT of the CTMC that is parameterized with θ_m as $\hat{\tau}_N^{\mathrm{FPEI}}(\theta_m) = \frac{1}{N}\sum_{n=1}^{N} Y_n^{\mathrm{FPEI}}(\theta_m)$.

With fixed paths, the MFPT estimates are biased and the learned parameter set might not perform well in the full state space where other paths are possible. Therefore, to reduce the bias and to ensure that the ensemble of paths is a fair sample with respect to the optimized parameters, we alternate between minimizing the error of prediction on fixed paths, and resampling new paths and

Algorithm 1. SSAI

```
θ ← θ_0                                          // Choose initial parameter set θ_0
Initialize the simplex in the Nelder-Mead algorithm using θ and its
perturbations
while stopping criteria not met do
                                  // See Section 4.3 for our stopping criteria
    θ ← Retrieve a parameter set from the Nelder-Mead algorithm
    Update the free parameters of the kinetic model with θ
    foreach reaction r ∈ dataset D do
        foreach n=1,2,...,N do
            Sample a trajectory P_n using SSA and calculate its FPT using
            Eq. 2
        Calculate the MFPT of the reaction using the FPTs of the trajectories
    Calculate the GMM function in Eq. 9 using the MFPT of the reactions
    Update the simplex in the Nelder-Mead algorithm based on the GMM
    function
```

Algorithm 2. FPEI

```
θ ← θ_0                                          // Choose initial parameter set θ_0
while stopping criteria not met do
                                  // See Section 4.3 for our stopping criteria
    Update the free parameters of the kinetic model with θ
    foreach reaction r ∈ dataset D do
        foreach n=1,2,...,N do
            Sample a path P_n using RVSSA
            Condense path P_n for the reaction
    Initialize the simplex in the Nelder-Mead algorithm using θ and its
    perturbations
    while stopping criteria not met do
        θ ← Retrieve a parameter set from the Nelder-Mead algorithm
        Update the free parameters of the kinetic model with θ
        foreach reaction r ∈ dataset D do
            foreach n=1,2,...,N do
                Calculate the MFPT of path P_n using Eq. 10
            Calculate the MFPT of the reaction using the MFPTs of the paths
        Calculate the GMM function in Eq. 9 using the MFPT of the reactions
        Update the simplex in the Nelder-Mead algorithm based on the GMM
        function
```

restarting the optimization method. An overview of our parameter estimation framework using SSAI and FPEI, with a GMM estimator and the Nelder-Mead algorithm, is given in Algorithms 1 and 2, respectively.

We also considered a normalized importance sampling approach [8], to obtain consistent estimators of the MFPTs using fixed paths. In this approach, we also compute the set of traversed transitions and how often each of those transitions occur in the path. We weigh the estimated MFPT of each path P by

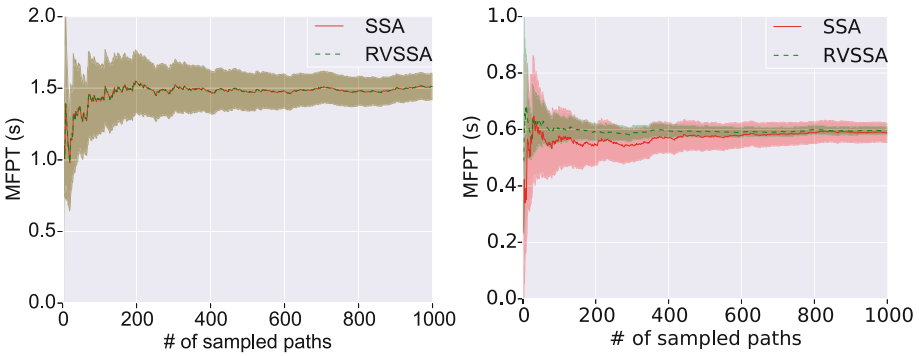
the relative likelihood of the path given the new and the initial parameter sets $\tilde{L}(\theta_m) = \frac{L(\theta_m)}{L(\theta_0)}$, where $L(\theta_m)$ is the likelihood of P under parameter assignment θ_m. For RVSSA, $L(\theta_m) = \prod_{i=1}^{Z-1} \frac{k_{s_i s_{i+1}}(\theta_m)}{\sum_{s \in \mathcal{S}} k_{s_i s}(\theta_m)} e^{-\sum_{s \in \mathcal{S}} k_{s_i s}(\theta_m) \overline{T}_i(\theta_m)}$, and we estimate the MFPT as $\hat{\tau}_N^{\text{FPEI}}(\theta_m) = \frac{1}{\sum_{n=1}^{N} \tilde{L}_n(\theta_m)} \sum_{n=1}^{N} \tilde{L}_n(\theta_m) Y_n^{\text{FPEI}}(\theta_m)$. In our experiments, this approach performed poorly, since the effective sample size of the relative likelihoods was small.

4 Experiments

To evaluate the performance of RVSSA for MFPT estimation and FPEI for parameter estimation, in the full state space of all non-pseudoknotted secondary structures of each reaction, we augment the Multistrand kinetic simulator [28,29] and we conduct computational experiments. Our dataset and framework are available at https://github.com/DNA-and-Natural-Algorithms-Group/FPEI.

4.1 Dataset

To evaluate the performance of RVSSA and FPEI, we use 21 experimentally determined reaction rate constants published in the literature, including hairpin closing [5], hairpin opening [5], helix association [16,37], and helix dissociation with and without mismatches [7]. The dataset is summarized in Table 1. Each reaction of the dataset is annotated with a temperature and the concentration of Na^+, which affect the transition rates in the kinetic models that we use.



(a) Reaction no. 16 (b) Reaction no. 20

Fig. 2. The MFPT and 95% confidence interval of SSA and RVSSA, where the kinetic model is parameterized with θ_0. In both (**a**) and (**b**), RVSSA and SSA are using the same sampled paths. In (**a**), RVSSA and SSA have similar variance. The average computation time per sampled path, defined as the total computation time divided by the total number of sampled paths, is 3×10^2 s. In (**b**), RVSSA has a lower variance than SSA. The average computation time per sampled path is 0.5 s.

Table 1. Dataset of experimentally determined reaction rate constants. The concentration of the strands is set to 1×10^{-8} M, 5×10^{-8} M, and 1×10^{-8} M, for reactions no. 1–15, 16–19, and 20–21, respectively.

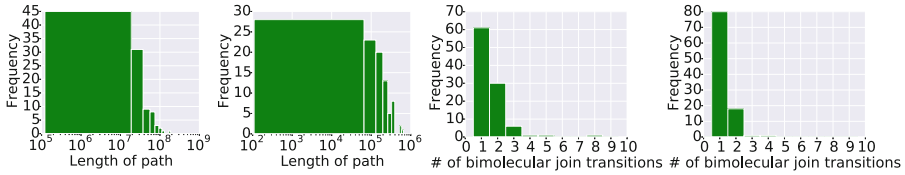
Reaction type & source	No.	Sequences	T/°C	$[Na]^+$/M	$\log_{10} k^r$
Hairpin closing Fig. 4 from [5]	1–5	CCCAA-$(T)_{30}$-TTGGG	14.4–29.8	0.1	3.53–3.69
Hairpin opening Fig. 4 from [5]	6–10	CCCAA-$(T)_{30}$-TTGGG	14.4–29.8	0.1	2.14–3.30
Helix dissociation (with a mismatch) Fig. S4 from [7]	11–15	AGGACTTGT + ACAAGACCT AGGACTTGT + ACAAGTGCT AGGACTTGT + ACAAGTCGT AGGACTTGT + ACAAGTCCA AGGACTTGT†	37	0.2	0.19–0.92
Helix association Table 1 from [16]	16–19	GCCCACACTCTTACTTATCGACT† GCACCTCCAAATAAAAACTCCGC† CGTCTATTGCTTGTCACTTCCCC† ACCCTTTATCCTGTAACTTCCGC†	25	0.195	5.71–6.68
Helix association Table 1 from [37]	20–21	25-mer Poly (dA)† 25-mer Poly (dG)†	48–78	0.4	-

†The complement of the demonstrated sequence is also a reactant.

4.2 Mean First Passage Time Estimation

Figures 2a and b show the performance of RVSSA compared with SSA for helix association reactions no. 16 and 20, respectively. To sample paths and trajectories, we parameterize the kinetic model with the Metropolis initial parameter set [32,42], in other words, $\theta_0 = \{k_{\text{uni}} = 8.2{\times}10^6\,\text{s}^{-1}, k_{\text{bi}} = 3.3{\times}10^5\,\text{M}^{-1}\text{s}^{-1}\}$. In both Figs. 2a and b, RVSSA and SSA have the same paths, but the algorithms generate different holding times for the states of the paths. In Multistrand's implementation of SSA, the effort needed to sample the holding time in the current state is small when compared to the task of computing outgoing transition rates. In Fig. 2a, RVSSA and SSA perform the same, whereas in Fig. 2b, RVSSA has a lower variance than SSA, consistently. To understand the discrepancy between the two figures, we analyze the experiments, described below.

In Figs. 3a and b, the average length of the paths for both reaction no. 16 and reaction no. 20 is large. Also, in Figs. 3c and d, both reactions have a small number of bimolecular transitions on average. In Fig. 3e, for reaction no. 16, the state where two strands are disconnected has a small holding time, because the state has many fast unimolecular transitions between complementary bases within a strand in addition to the slow bimolecular transitions. However, in Fig. 3f, for reaction no. 20, the state where the two strands are disconnected has a large holding time, since there are no complementary bases within a poly(A) or poly(T) strand and the only transitions are slow bimolecular transitions. RVSSA has a significantly lower variance for reaction no. 20 compared to SSA, because in the sampled paths, there exists a small number of states that have large expected holding times compared to other states. SSA has a large variance in generating holding times for these states. Overall, in our experiments with parameter set θ_0,



(a) Reaction no. 16 (b) Reaction no. 20 (c) Reaction no. 16 (d) Reaction no. 20

```
1:        ..........................  .......................  t=0.000000000 s
2:        ..........................  ..(....)...............  t=0.000000045 s
.
19166996: ...........(......)....  .......................  t=0.751824960 s
19166997: ..........................  .......................  t=0.751825005 s
19166998: ...................(...+.....).................  t=0.751825051 s
19166999: ...................(...+.....).......(.....)...  t=0.751825095 s
.
19167442: ((((((((((((((((((((((((+))))))))))))))))))))))))  t=0.751841627 s
```

(e) Reaction no. 16

```
1:       ..........................  .......................  t=0.000000000 s
2:       ............(............+....).....................  t=0.219236868 s
3:       ............(.(..........+..).).....................  t=0.219236888 s
4:       ............(............+....).....................  t=0.219236898 s
.
173450: ((((((((((((((((((((((((+))))))))))))))))))))))))  t=0.222540278 s
```

(f) Reaction no. 20

Fig. 3. Histogram of the length of 100 random paths obtained with RVSSA for (**a**) reaction no. 16 and (**b**) reaction no. 20. Histogram of the number of bimolecular join transitions of the random paths for (**c**) reaction no. 16 and (**d**) reaction no. 20. Snapshot of the i-th state visited, dot-parentheses notation and jump times for a random path obtained with RVSSA for (**e**) reaction no. 16 and (**f**) reaction no. 20. The jump time at state i is equal to the jump time at state $i-1$ plus the holding time of state $i-1$. The green highlighting indicates where a bimolecular step occurs. (Color figure online)

RVSSA has a lower variance than SSA for reactions no. 20 and 21, but performs similar to SSA for other reactions in Table 1.

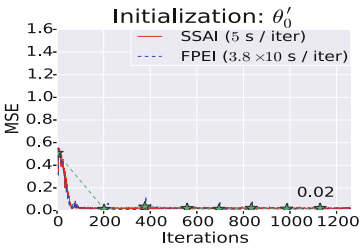
4.3 Parameter Estimation

Figure 4 shows the MSE, defined as the mean of $|\log_{10} k^r - \log_{10} \hat{k}^r(\theta)|^2$ on different reactions, of FPEI and SSAI over various iterations, where the methods are learning parameters for the Arrhenius kinetic model [42]. Also, it shows the average computation time per iteration, defined as the total computation time

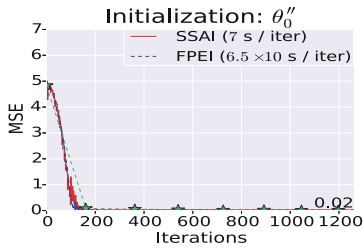
divided by the total number of iterations. Figure 5 shows the MSE and average computation time per iteration when the entire dataset is used. Reactions no. 20–21 are excluded in parameter estimation because of our uncertainty in our interpretation of the reported measurements. For reactions no. 1–15, FPEI and SSAI use 200 paths and 200 trajectories, respectively. For reactions no. 16–19, where simulations are more time-consuming, FPEI and SSAI use 20 paths and 20 trajectories, respectively.

We conduct distinct experiments by starting with two sets of initial parameters, where paths and trajectories are generated in a reasonable time. In one group of experiments (Figs. 4a, c, e, g, and 5a), we initialize the simplex in the Nelder-Mead algorithm with the Arrhenius initial parameter set [32,42], in other words, $\theta_0' = \{A_l = 468832.1058\,\mathrm{s}^{-1/2}, E_l = 3\,\mathrm{kcal\,mol}^{-1} \mid \forall l \in \mathcal{C}\} \cup \{\alpha = 0.0402\,\mathrm{M}^{-1}\}$ and its perturbations (in each perturbation, a parameter is multiplied by 1.05). In FPEI, we also use θ_0' to generate fixed paths. In another set of experiments (Figs. 4b, d, f, h, and 5b), we adapt parameter set $\theta_0'' = \{A_l = 468832.1058\,\mathrm{s}^{-1/2}, E_l = 2\,\mathrm{kcal\,mol}^{-1} \mid \forall l \in \mathcal{C}\} \cup \{\alpha = 0.0402\,\mathrm{M}^{-1}\}$ from θ_0' to increase the initial MSE in all experiments. We initialize the simplex in the Nelder-Mead algorithm with θ_0'' and its perturbations (in each perturbation, a parameter is multiplied by 1.05). In FPEI, we also generate fixed paths with θ_0''. In SSAI, we run the optimization until a limit on the number of iterations is reached or until a time limit is reached, which ever comes first. We also use this as the first stopping criteria in FPEI. In FPEI, to reduce the bias and to ensure that the ensemble of paths is a fair sample with respect to the optimized parameters, occasionally, the fixed paths are rebuilt from scratch and the optimization restarts. To this end, we set the second stopping criteria in FPEI to 200 iterations or 200 function evaluations of the Nelder-Mead algorithm, whichever comes first. Note that this empirical value is subject to change for different experiments. We could improve the method, by investigating a more robust way of when to update the paths. For example, we could compare the performance of SSA with the fixed paths in shorter intervals and update the fixed paths when their predictive quality diverges from SSA. During the optimization, we use an infinite value for parameter sets that have rates that are too slow or too fast; we bound downhill unimolecular and bimolecular rates (Eq. (7) and Eq. (8) of [42]) in $[1 \times 10^4, 1 \times 10^9]$ s^{-1} and in $[1 \times 10^4, 1 \times 10^9]\,\mathrm{M}^{-1}\mathrm{s}^{-1}$, respectively.

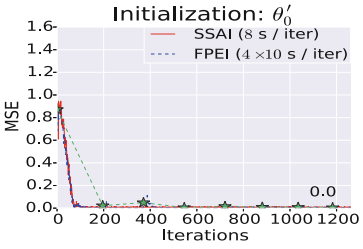
In Figs. 4d–h, FPEI reaches a minimal MSE more quickly than SSAI; consider the average computation time per iteration multiplied by the number of iterations to reach a minimal MSE. However, in Figs. 4a–c, SSAI reaches a minimal MSE more quickly than FPEI. This is because in Figs. 4d–h, the number of unique states is significantly smaller than the length of the paths. For example, in Fig. 4h, in the first set of fixed paths, the average length of a path is more than 2.3×10^7, whereas the average number of unique states and transitions is less than 1.5×10^5 and 5.6×10^5, respectively. In Fig. 4a, the average length of a path is 4.6×10^2, whereas the average number of unique states and transitions is 1.3×10^2 and 2.4×10^2, respectively. In Figs. 4e and f, which are slow dissociation reactions, compared to SSAI, FPEI speeds up parameter estimation by



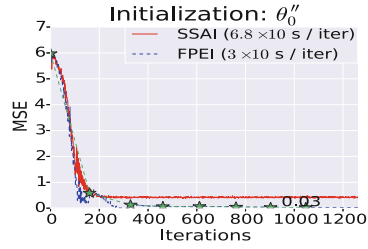
(a) Reactions no. 1-5 (hairpin closing)



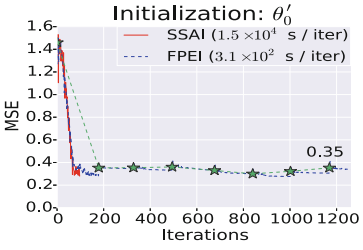
(b) Reactions no. 1-5 (hairpin closing)



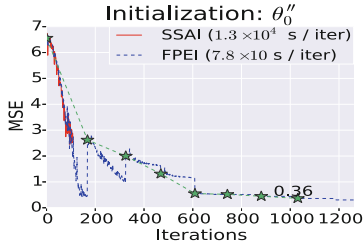
(c) Reactions no. 6-10 (hairpin opening)



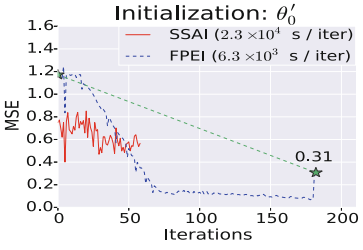
(d) Reactions no. 6-10 (hairpin opening)



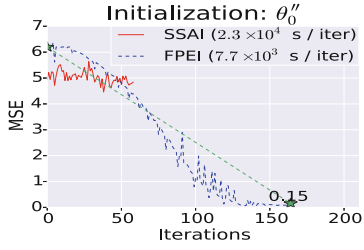
(e) Reactions no. 11-15 (helix dissociation)



(f) Reactions no. 11-15 (helix dissociation)



(g) Reactions no. 16-19 (helix association)

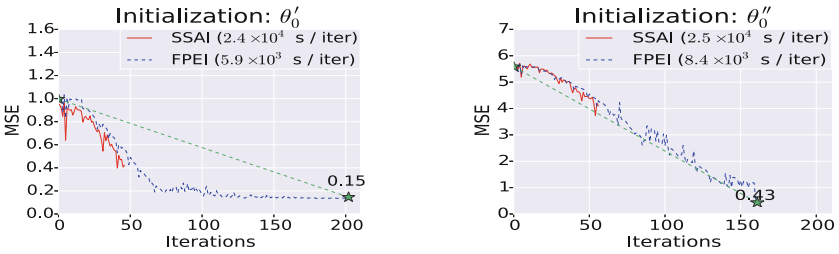


(h) Reactions no. 16-19 (helix association)

Fig. 4. The MSE of SSAI and FPEI on different types of reactions from Table 1. The average computation time per iteration is shown in the label of each method. The ★ markers show the MSE when trajectories are rebuilt from scratch using the learned parameter set from FPEI. In Figs. 4e–h, the SSAI traces stop at earlier iterations than the FPEI traces, even though SSAI was allocated more time than FPEI.

more than a factor of three. In Figs. 4g–h, compared to SSAI, FPEI speeds up parameter estimation by more than a factor of two. Also, the speed of FPEI in all the figures could be improved with a better implementation of the method; in our implementation, in the first iteration, computing neighbor states of all states in a fixed condensed path is slow, whereas the later iterations which reuse the fixed condensed paths are much faster than SSAI.

In Figs. 5a and b, where reactions no. 1–19 are all used in the optimization, FPEI speeds up parameter estimation, by more than a factor of two compared to SSAI. In Fig. 5a, FPEI reaches an MSE of 0.15 in 1.2×10^6 s, whereas SSAI reaches an MSE of 0.39 in the same time. In Fig. 5b, FPEI reaches an MSE of 0.43 in 1.3×10^6 s, whereas SSAI reaches an MSE of 3.72 in the same time.



(a) Reactions no. 1-19 (b) Reactions no. 1-19

Fig. 5. As in Fig. 4, but reactions no. 1–19 are all used as the dataset.

5 Discussion

In this work, we show how to use RVSSA to estimate the MFPT of a reaction's CTMC. In our experiments, RVSSA has a lower variance than SSA in estimating the MFPT of a reaction's CTMC, when in the sampled paths there exists a small number of states that have large expected holding times compared to other states. Furthermore, we show how to use FPEI along with a GMM estimator and the Nelder-Mead algorithm to estimate parameters for DNA kinetics modeled as CTMCs. In FPEI, we use RVSSA instead of SSA, since the MFPT estimator produced by RVSSA has a lower variance. In FPEI, we use fixed condensed paths because sampling new paths for every parameter set is computationally expensive. Since using fixed paths leads to biased estimates, we alternate between minimizing the error of prediction on fixed paths, and resampling new paths and restarting the optimization method. FPEI speeds up computations when the number of unique states is significantly smaller than the length of sampled paths. In our experiments on a dataset of DNA reactions, FPEI speeds up parameter estimation compared to using SSAI, by more than a factor of three for slow reactions. Also, in our experiments, for reactions with large state spaces, it speeds up parameter estimation by more than a factor of two.

FPEI can be applied to reactions modeled as CTMCs, when the fixed paths can be produced in a timely manner. Generating paths for FPEI could be computationally expensive for slow reactions, such as helix dissociation from Morrison and Stols [25]. The runtime also depends on the parameter set that is used. It would be helpful to make FPEI applicable for such reactions, by speeding up the generation of the fixed paths, for example, with importance sampling approaches [13].

Finally, in this work, we evaluated FPEI in the context of DNA reactions. It would be useful to adopt and evaluate FPEI in other CTMC models [9,34,35], and other domains that require estimating MFPTs in CTMCs, such as protein folding.

References

1. Andrieu, C., Roberts, G.O.: The pseudo-marginal approach for efficient Monte Carlo computations. Ann. Stat. **37**, 697–725 (2009)
2. Andronescu, M., Aguirre-Hernandez, R., Condon, A., Hoos, H.H.: RNAsoft: a suite of RNA secondary structure prediction and design software tools. Nucleic Acids Res. **31**, 3416–3422 (2003)
3. Andronescu, M., Condon, A., Hoos, H.H., Mathews, D.H., Murphy, K.P.: Computational approaches for RNA energy parameter estimation. RNA **16**(12), 2304–2318 (2010)
4. Barton, R.R., Ivey Jr., J.S.: Nelder-Mead simplex modifications for simulation optimization. Manag. Sci. **42**(7), 954–973 (1996)
5. Bonnet, G., Krichevsky, O., Libchaber, A.: Kinetics of conformational fluctuations in DNA hairpin-loops. Proc. Natl. Acad. Sci. **95**(15), 8602–8606 (1998)
6. Cherry, K.M., Qian, L.: Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. Nature **559**(7714), 370 (2018)
7. Cisse, I.I., Kim, H., Ha, T.: A rule of seven in Watson-Crick base-pairing of mismatched sequences. Nat. Struct. Mol. Biol. **19**(6), 623 (2012)
8. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. Handb. Nonlinear Filter. **12**(656–704), 3 (2009)
9. Flamm, C., Fontana, W., Hofacker, I.L., Schuster, P.: RNA folding at elementary step resolution. RNA **6**, 325–338 (2000)
10. Georgoulas, A., Hillston, J., Sanguinetti, G.: Unbiased Bayesian inference for population Markov jump processes via random truncations. Stat. Comput. **27**(4), 991–1002 (2017)
11. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**(25), 2340–2361 (1977)
12. Golightly, A., Wilkinson, D.J.: Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. Interface Focus **1**(6), 807–820 (2011)
13. Hajiaghayi, M., Kirkpatrick, B., Wang, L., Bouchard-Côté, A.: Efficient continuous-time Markov chain estimation. In: International Conference on Machine Learning, pp. 638–646 (2014)
14. Hansen, L.P.: Large sample properties of generalized method of moments estimators. Econ. J. Econ. Soc. **50**, 1029–1054 (1982)
15. Hansen, L.P., Heaton, J., Yaron, A.: Finite-sample properties of some alternative GMM estimators. J. Bus. Econ. Stat. **14**(3), 262–280 (1996)

16. Hata, H., Kitajima, T., Suyama, A.: Influence of thermodynamically unfavorable secondary structures on DNA hybridization kinetics. Nucleic Acids Res. **46**(2), 782–791 (2017)
17. Hofacker, I.L.: Vienna RNA secondary structure server. Nucleic Acids Res. **31**(13), 3429–3431 (2003)
18. Hordijk, A., Iglehart, D.L., Schassberger, R.: Discrete time methods for simulating continuous time Markov chains. Adv. Appl. Probab. **8**(4), 772–788 (1976)
19. Horváth, A., Manini, D.: Parameter estimation of kinetic rates in stochastic reaction networks by the EM method. In: 2008 International Conference on BioMedical Engineering and Informatics, vol. 1, pp. 713–717. IEEE (2008)
20. Lehmann, E.L., Casella, G.: Theory of Point Estimation. Springer, New York (2006)
21. Loskot, P., Atitey, K., Mihaylova, L.: Comprehensive review of models and methods for inferences in bio-chemical reaction networks. arXiv preprint arXiv:1902.05828 (2019)
22. Lück, A., Wolf, V.: Generalized method of moments for estimating parameters of stochastic reaction networks. BMC Syst. Biol. **10**(1), 98 (2016)
23. Mathews, D.H., Sabina, J., Zuker, M., Turner, D.H.: Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. J. Mol. Biol. **288**(5), 911–940 (1999)
24. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
25. Morrison, L.E., Stols, L.M.: Sensitive fluorescence-based thermodynamic and kinetic measurements of DNA hybridization in solution. Biochemistry **32**, 3095–3104 (1993)
26. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)
27. Ouldridge, T.E., Louis, A.A., Doye, J.P.: Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. J. Chem. Phys. **134**(8), 02B627 (2011)
28. Schaeffer, J.M.: Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. Ph.D. thesis, California Institute of Technology (2012)
29. Schaeffer, J.M., Thachuk, C., Winfree, E.: Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In: Phillips, A., Yin, P. (eds.) DNA 2015. LNCS, vol. 9211, pp. 194–211. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21999-8_13
30. Schnoerr, D., Sanguinetti, G., Grima, R.: Approximation and inference methods for stochastic biochemical kinetics–a tutorial review. J. Phys. Math. Theor. **50**(9), 093001 (2017)
31. Schreck, J.S., et al.: DNA hairpins destabilize duplexes primarily by promoting melting rather than by inhibiting hybridization. Nucleic Acids Res. **43**(13), 6181–6190 (2015)
32. Srinivas, N., et al.: On the biophysics and kinetics of toehold-mediated DNA strand displacement. Nucleic Acids Res. **41**, 10641–10658 (2013)
33. Suhov, Y., Kelbert, M.: Probability and Statistics by Example: Volume 2, Markov Chains: A Primer in Random Processes and Their Applications, vol. 2. Cambridge University Press, Cambridge (2008)
34. Šulc, P., Romano, F., Ouldridge, T.E., Rovigatti, L., Doye, J.P., Louis, A.A.: Sequence-dependent thermodynamics of a coarse-grained DNA model. J. Chem. Phys. **137**(13), 135101 (2012)

35. Tang, X., Kirkpatrick, B., Thomas, S., Song, G., Amato, N.M.: Using motion planning to study RNA folding kinetics. J. Comput. Biol. **12**(6), 862–881 (2005)
36. Wackerly, D., Mendenhall, W., Scheaffer, R.L.: Mathematical Statistics with Applications. Cengage Learning, Boston (2014)
37. Wetmur, J.G.: Hybridization and renaturation kinetics of nucleic acids. Annu. Rev. Biophys. Bioeng. **5**(1), 337–361 (1976)
38. Xu, Z.Z., Mathews, D.H.: Experiment-assisted secondary structure prediction with RNAstructure. In: Turner, D., Mathews, D. (eds.) RNA Structure Determination: Methods and Protocols, pp. 163–176. Humana Press, New York (2016)
39. Zadeh, J.N., et al.: NUPACK: analysis and design of nucleic acid systems. J. Comput. Chem. **32**, 170–173 (2011)
40. Zhang, D.Y., Winfree, E.: Control of DNA strand displacement kinetics using toehold exchange. J. Am. Chem. Soc. **131**, 17303–17314 (2009)
41. Zhang, J.X., et al.: Predicting DNA hybridization kinetics from sequence. Nat. Chem. **10**(1), 91 (2018)
42. Zolaktaf, S., et al.: Inferring parameters for an elementary step model of DNA structure kinetics with locally context-dependent arrhenius rates. In: Brijder, R., Qian, L. (eds.) DNA 2017. LNCS, vol. 10467, pp. 172–187. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66799-7_12