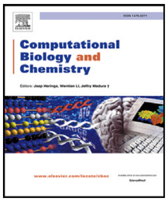
Computational Biology and Chemistry 104 (2023) 107837

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac



Research Article

Predicting DNA kinetics with a truncated continuous-time Markov chain method

Sedigheh Zolaktaf [a,*,1], Frits Dannenberg [b,1], Mark Schmidt [a,c], Anne Condon [a], Erik Winfree [b]

[a] University of British Columbia, Canada
[b] California Institute of Technology, United States of America
[c] Canada CIFAR AI Chair (Amii), Canada

ARTICLE INFO

Keywords:
DNA kinetics
Reaction rate constant estimation
Truncated continuous-time Markov chain

ABSTRACT

Predicting the kinetics of reactions involving nucleic acid strands is a fundamental task in biology and biotechnology. Reaction kinetics can be modeled as an elementary step continuous-time Markov chain, where states correspond to secondary structures and transitions correspond to base pair formation and breakage. Since the number of states in the Markov chain could be large, rates are determined by estimating the mean first passage time from sampled trajectories. As a result, the cost of kinetic predictions becomes prohibitively expensive for rare events with extremely long trajectories. Also problematic are scenarios where multiple predictions are needed for the same reaction, e.g., under different environmental conditions, or when calibrating model parameters, because a new set of trajectories is needed multiple times. We propose a new method, called pathway elaboration, to handle these scenarios. Pathway elaboration builds a truncated continuous-time Markov chain through both biased and unbiased sampling. The resulting Markov chain has moderate state space size, so matrix methods can efficiently compute reaction rates, even for rare events. Also the transition rates of the truncated Markov chain can easily be adapted when model or environmental parameters are perturbed, making model calibration feasible. We illustrate the utility of pathway elaboration on toehold-mediated strand displacement reactions, show that it well-approximates trajectory-based predictions of unbiased elementary step models on a wide range of reaction types for which such predictions are feasible, and demonstrate that it performs better than alternative truncation-based approaches that are applicable for mean first passage time estimation. Finally, in a small study, we use pathway elaboration to optimize the Metropolis kinetic model of Multistrand, an elementary step simulator, showing that the optimized parameters greatly improve reaction rate predictions. Our framework and dataset are available at https://github.com/DNA-and-Natural-Algorithms-Group/PathwayElaboration.

1. Introduction

Reactions involving nucleic acid strands play key roles in cellular processes, are valuable tools in synthetic biology, and are the basis for programming in the field of DNA computing. Examples of such reactions include the folding of tRNA strands, which aid in protein synthesis in the cell; RNA toehold switching, which can be used to detect the presence of small molecules or biomolecules (Angenent-Mari et al., 2020), and DNA three-way strand displacement, which is used to implement logic circuits and oscillators (Soloveichik et al., 2010; Srinivas et al., 2017). Each reactant is a single strand or a complex comprised of a small handful of short strands, plus its associated secondary structure (set of complementary base pairs) if any. The reaction's products involve the same strand(s), in a new structural configuration.

Good software simulators would be a boon to scientists and engineers who study and design such reactions. Simulators could flexibly model different environmental conditions such as temperature, could predict reaction rates, and could sample folding trajectories that provide detailed insight on mechanistic principles or unexpected behaviors. Some simulators use coarse-grained models that consider large conformational changes (Sun et al., 2018; Isambert and Siggia, 2000), while others use elementary step models that consider the formation or breaking of individual base pairs (Flamm et al., 2000; Schaeffer et al., 2015). Molecular dynamic models that follow the three-dimensional

[*] Corresponding author.
E-mail addresses: nasimzf@cs.ubc.ca (S. Zolaktaf), fdannenberg@live.nl (F. Dannenberg), schmidtm@cs.ubc.ca (M. Schmidt), condon@cs.ubc.ca (A. Condon), winfree@caltech.edu (E. Winfree).
[1] Joint first authorship for first two authors: Sedigheh Zolaktaf and Frits Dannenberg contributed equally to this work.

https://doi.org/10.1016/j.compbiolchem.2023.107837
Received 31 July 2022; Received in revised form 5 February 2023; Accepted 21 February 2023
Available online 25 February 2023

motion of the polymer chains are also well-established (Ouldridge et al., 2011; Šulc et al., 2012). In this work, we are interested in elementary step models because they are computationally more efficient than molecular dynamics, yet they can uncover unexpected secondary structures in intermediate states. The Kinfold unimolecular simulator (Flamm et al., 2000) proceeds in elementary steps, or transitions, in which a base pair forms or breaks. Multistrand (Schaeffer, 2013; Schaeffer et al., 2015) adopts the same principles, and can simulate unimolecular and bimolecular reactions involving multistranded DNA complexes. These simulators do not explicitly store all possible states, i.e., secondary structures of the reacting strand(s), since the number of states can be exponential in the total strand lengths. Rather, they stochastically generate successive secondary structures of a trajectory on the fly, along with the time for each transition, in a manner consistent with kinetic and thermodynamic models and detailed balance.

However, trajectory-based simulation of all but the simplest examples is computationally costly, particularly for reactions, such as rare events, with long folding trajectories. The cost is amplified when doing simulations at multiple temperatures, because new trajectories are needed for each temperature setting. Another problem is that, in contrast with thermodynamic "nearest neighbor" models which have been extensively trained using experimental data, current kinetic models are rather simplistic. Kinfold's Metropolis and Kawasaki kinetic models use a single parameter, and Multistrand's Metropolis model has two parameters, one for unimolecular and one for bimolecular reactions. Rate predictions with current kinetic parameters can be off by orders of magnitude. A 15-parameter kinetic model that is based on Arrhenius principles shows promise (Zolaktaf et al., 2019), but the model parameters have not yet been well calibrated, because of the cost of running multiple inference steps, each relying on a new set of trajectories for each of a large set of reactions.

What is needed is a way to *efficiently* and *adaptably* approximate the predictions of elementary step nucleic acid kinetic simulators. Efficiency makes simulation of rare events possible, and adaptability makes efficient updates possible when the parameters of the kinetic model or environment change. There is extensive literature, discussed further in Section 2, on simulation of molecular folding, but none is well suited to address the unique combination of technical challenges here– exponentially large state spaces, rare events, inference on trajectory space rather than state space, and changing model parameters. Briefly, methods that rely on sampled trajectories alone are not adaptable since (as noted already above) they require new trajectories when the model changes. Alternative approaches use coarse-graining or probabilistic roadmaps to build approximate Markov chain models with significantly fewer states than the full elementary step model, making it possible to use matrix methods to efficiently compute reaction rates. However, these methods are not adaptable because calculating transition rates between states typically involves costly estimation of energy barriers. The most promising methods build *truncated* Markov chains (Kuntz et al., 2021), where the state space and transitions are subsets of those in the elementary step model, and so the cost to update each transition rate is constant. However, current truncation-based approaches either require prohibitively many states, or simulation time, or choose states via biased sampling alone, thereby omitting deep energy basins that strongly influence reaction rates.

In this work we propose a new approach, called pathway elaboration, illustrated in Fig. 1. Pathway elaboration leverages the Multistrand simulator to select a subset of the states through biased path sampling (1a) as well as unbiased (1b) exploration. Biased sampling efficiently finds trajectories from initial to final states even for rare events, while unbiased elaboration from states of the biased samples can discover low-energy basins in which a reaction can get "trapped". Elementary step transitions are added (1c) between pairs of explored states that are adjacent, i.e., differ by one base pair. A pruning step (1d) then removes states and associated transitions, while keeping reaction rate estimates within predetermined upper bounds. The result is a continuous-time, truncated Markov chain representation of the reaction. With this truncated representation, matrix methods can efficiently estimate reaction rates, even for rare events, and trajectories can be sampled. The same representation can be used even when parameters of the kinetic model, or temperature, are slightly perturbed (1e), and can be updated in time proportional to the number of states for larger perturbations. As a result, it is possible to amortize the initial cost of running pathway elaboration to reduce the cost of kinetic parameter inference (1f) or rate estimation of the same reaction at different temperatures (1g).

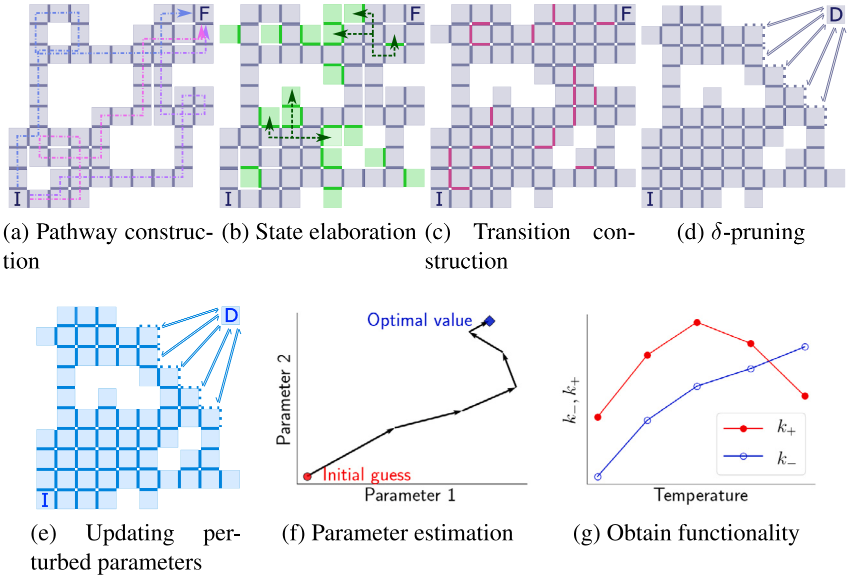
We evaluate pathway elaboration in several ways. We first use a case study to illustrate how pathway elaboration provides insight on the kinetics of two contrasting DNA reactions. Both reactions involve toehold-mediated strand displacement, with the second differing from the first by the introduction of a single mismatch between the invading strand and the substrate to which it binds. Pathway elaboration correctly predicts that the second is roughly three orders of magnitude slower than the first, and a visualization of the trajectories sampled from the truncated Markov chain shows how the energy barrier introduced by the mismatch slows down the reaction.

We then compare pathway elaboration's predictions with Multistrand's unbiased stochastic simulation (Gillespie, 1977; Doob, 1942). We compare with this unbiased "gold standard" simulation mode, rather than with experimentally determined reaction rates, because we want to understand the degree to which pathway elaboration's truncation changes the predicted rate. For this study, we use a diverse set of 237 unimolecular and bimolecular reactions for which unbiased simulation is feasible. While the reaction rate constants predicted by unbiased simulation on this dataset differ by over 7 orders of magnitude, pathway elaboration's predictions differ from unbiased simulation by a factor of just 13% on average, an encouraging finding. In our experiments, pathway elaboration is on average 5 times faster than stochastic simulation. The quality of pathway elaboration's predictions is better on average than an alternative truncation-based method that we implemented, based on transition path sampling (Bolhuis et al., 2002).

Finally, we use pathway elaboration to rapidly evaluate perturbed model parameters during optimization of Multistrand's two kinetic parameters. We use the experimentally determined rates of the same 237 reactions to train the optimizer and an additional 30 reactions as our testing set. On these 30 reactions, which involve rare events and have large state spaces, unbiased stochastic simulation is too costly to run (requiring more than two weeks per reaction on our system). Here we compare the *experimentally determined* reaction rates (rather than the rates predicted by unbiased simulation) with the rates produced by pathway elaboration before and after optimization, since the purpose is to see if pathway elaboration's truncated model shows promise as a practical approach for parameter optimization. On the training set, a 26.9-fold average error in the predicted reaction rate constant reduces to a 2.8-fold average error, and for the 30 test reactions, a 13.4-fold average error reduces to a 4.3-fold average error. The entire optimization and evaluation takes less than five days.

2. Background and related work

In Section 2.1 we provide background on the continuous-time Markov chain (CTMC) model to which our pathway elaboration method (Section 3) applies and also provide related definitions. In Section 2.2, we describe the most relevant concepts for interacting nucleic acid strands. We also describe how the Multistrand kinetic simulator models the kinetics of multiple interacting nucleic acid strands as CTMCs and how it estimates reaction rate constants from mean first passage time (MFPT) estimates for these reactions. Finally, in Section 2.3 we provide further related work on MFPT and reaction rate constant estimation and model calibration.



(a) Pathway construction (b) State elaboration (c) Transition construction (d) δ-pruning

(e) Updating perturbed parameters (f) Parameter estimation (g) Obtain functionality

Fig. 1. The pathway elaboration method and its applications. Pathway elaboration can be used for mean first passage time estimation of rare events and the rapid evaluation of perturbed parameters. Here, in the underlying detailed-balance continuous-time Markov chain, boxes in a square grid represent states of the continuous-time Markov chain, with transitions between adjacent boxes, initial state I at bottom left and target state F at top right. (**a**) From state I, sample paths that are biased towards the target state F. Three sampled paths are shown with blue, pink and purple dotted lines. (**b**) From each sampled state found in the previous step, run short unbiased simulations to fill in the neighborhood. Simulations from two states are shown with green dashed lines. The green states and transitions are sampled. (**c**) Include all missing transitions between the states that were sampled in steps **a** and **b**. The red transitions are included. (**d**) Prune states that are expected to reach the target state quickly by redirecting their transitions into a new target state. (**e**) For perturbed model parameters, keep the topology of the truncated continuous-time Markov chain, but update the transition rates. (**f**) We can use truncated continuous-time Markov chains for perturbed parameters, such as to estimate model parameters or (**g**) to predict forward (k_+) and reverse (k_-) reaction rate constants as temperature changes.

2.1. Continuous-time Markov chain

Continuous-time Markov chain (CTMC). We indicate a CTMC as a tuple $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$, where $\mathcal{S}$ is a countable set of states, $\mathbf{K} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is the rate matrix and $\mathbf{K}(s,s) = 0$ for $s \in \mathcal{S}$, $\pi_0 : \mathcal{S} \rightarrow [0,1]$ is the initial state distribution in which $\sum_{s \in \mathcal{S}} \pi_0(s) = 1$, and $\mathcal{S}_{\text{target}}$ is the set of target states. We define the set of initial states as $\mathcal{S}_{\text{init}} = \{ s \in \mathcal{S} \mid \pi_0(s) \neq 0 \}$. For CTMCs considered here, $\mathcal{S}_{\text{target}} \cap \mathcal{S}_{\text{init}} = \emptyset$. A transition between states $s, s' \in \mathcal{S}$ can occur only if $\mathbf{K}(s,s') > 0$. The probability of moving from state s to state s' is defined by the transition probability matrix $\mathbf{P} : \mathcal{S} \times \mathcal{S} \rightarrow [0,1]$ where

$$\mathbf{P}(s,s') = \frac{\mathbf{K}(s,s')}{\mathbf{E}(s,s)}. \tag{1}$$

Here $\mathbf{E} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is a diagonal matrix in which $\mathbf{E}(s,s) = \sum_{s' \in \mathcal{S}} \mathbf{K}(s,s')$ is the exit rate. The time spent in state s before a transition is triggered is exponentially distributed with exit rate $\mathbf{E}(s,s)$. The generating matrix $\mathbf{Q} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ is $\mathbf{Q} = \mathbf{K} - \mathbf{E}$.

Detailed-balance CTMC. In a detailed-balance CTMC $\mathcal{C}^R = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}}, \pi)$, also known as a reversible CTMC, a probability distribution $\pi : \mathcal{S} \rightarrow [0,1]$ over the states exists that satisfies the detailed balance condition $\pi(s)\mathbf{K}(s,s') = \pi(s')\mathbf{K}(s',s)$ for all $s, s' \in \mathcal{S}$. The detailed balance condition is a sufficient condition for ensuring that π is a stationary distribution ($\pi\mathbf{P} = \pi$). For a detailed-balance finite-state CTMC, π is the unique stationary distribution of the chain and is also the unique equilibrium distribution (Whitt, 2006).

Boltzmann distribution. In many Markov models of physical systems, eventually the population of states will stabilize and reach a Boltzmann distribution (Schaeffer et al., 2015; Flamm et al., 2000; Tang, 2010) at equilibrium. With this distribution, the probability that a system is in a state s is

$$\pi(s) = \frac{1}{Z} e^{-\frac{E(s)}{k_B T}}, \tag{2}$$

where $E(s)$ is the energy of the system at state s, T is the temperature, k_B is the Boltzmann constant, and $Z = \sum_{s \in \mathcal{S}} e^{-\frac{E(s)}{k_B T}}$ is the partition function. To ensure that at equilibrium states are Boltzmann distributed, the detailed balance conditions are

$$\frac{\mathbf{K}(s,s')}{\mathbf{K}(s',s)} = e^{-\frac{E(s')-E(s)}{k_B T}}. \tag{3}$$

Reversible transition. In this work, a reversible transition between states s and s' means $\mathbf{K}(s,s') > 0$ if and only if $\mathbf{K}(s',s) > 0$.

Trajectories and paths. A trajectory $(s_0, t_0), (s_1, t_1,), \ldots, (s_m, t_m,)$ with m transitions over a CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$ is a sequence of states s_i and holding times t_i for which $\mathbf{K}(s_i, s_{i+1}) > 0$ and $t_i \in \mathbb{R}_{>0}$ for $i \geq 0$. We define a path $s_0, s_1, \ldots, s_m$ with m transitions over a CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$ as a sequence of states s_i for which $\mathbf{K}(s_i, s_{i+1}) > 0$.

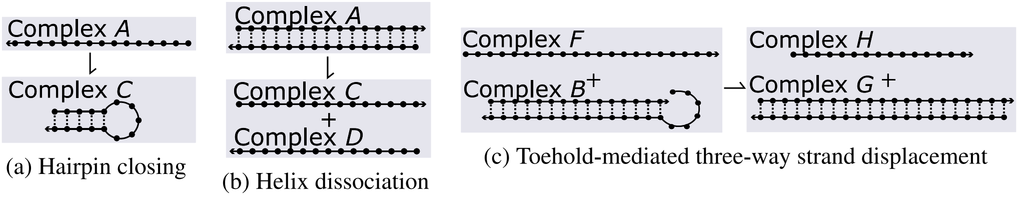
The stochastic simulation algorithm (SSA). SSA (Gillespie, 1977; Doob, 1942) simulates statistically correct trajectories over a CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$. At state s_i, the probability of sampling s_{i+1} is $\mathbf{P}(s_i, s_{i+1})$. At a jump from state s_i, it samples the holding time T_i from an exponential distribution with exit rate $\mathbf{E}(s,s) = \sum_{s' \in \mathcal{S}} \mathbf{K}(s,s')$.

Mean first passage time (MFPT). In a CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$, for a state $s \in \mathcal{S}$ and a target state $s_{\text{f}} \in \mathcal{S}_{\text{target}}$, the MFPT τ_s is the expected time to first reach s_{f} starting from state s. For state s, the MFPT from s to s_{f} equals the expected holding time in state s plus the MFPT to s_{f} from the next visited state (Suhov and Kelbert, 2008), so

$$\tau_s = \frac{1}{\mathbf{E}(s,s)} + \sum_{s' \in \mathcal{S}} \frac{\mathbf{K}(s,s')}{\mathbf{E}(s,s)} \tau_{s'}. \tag{4}$$

Multiplying the equation by the exit rate $\mathbf{E}(s,s) = \sum_{s' \in \mathcal{S}} \mathbf{K}(s,s')$ then yields

$$\sum_{s' \in \mathcal{S}} \mathbf{K}(s,s')(\tau_{s'} - \tau_s) = -1. \tag{5}$$



(a) Hairpin closing (b) Helix dissociation (c) Toehold-mediated three-way strand displacement

Fig. 2. Examples of unimolecular and bimolecular interacting nucleic acid strand reactions. (**a**) Hairpin closing is a unimolecular reaction. It has one reactant complex (A) and one product complex (C). The reverse reaction, hairpin opening, is also a unimolecular reaction. (**b**) Helix dissociation is a unimolecular reaction. It has one reactant complex (A) and two product complexes (C and D). The reverse reaction, helix association, is a bimolecular reaction. (**c**) Toehold-mediated three-way strand displacement is a bimolecular reaction. It has two reactant complexes (B and F) and two product complexes (G and H).

Now writing $\mathbf{t} : \mathcal{S} \setminus s_{\mathrm{f}} \to \mathbb{R}_{\geq 0}$ to be the vector of MFPTs for each state, such that $\mathbf{t}[s] = \tau_s$, we find a matrix equation as

$$\tilde{\mathbf{Q}}\mathbf{t} = -\mathbf{1}, \tag{6}$$

where $\tilde{\mathbf{Q}}$ is obtained from $\mathbf{Q}$ by eliminating the row and column corresponding to the target state, and $\mathbf{1}$ is a vector of ones. If there exists a path from every state to the final state s_{f}, then $\tilde{\mathbf{Q}}$ is a weakly chained diagonally dominant matrix and is non-singular (Azimzadeh and Forsyth, 2016). The MFPT from the initial states to the target state s_{f} is found as

$$\tau_{\pi_0} = \sum_{s \in \mathcal{S}} \pi_0(s)\tau_s. \tag{7}$$

If instead of a single target state s_{f} we have a set of target states $\mathcal{S}_{\mathrm{target}}$, then to compute the MFPT to $\mathcal{S}_{\mathrm{target}}$ we convert all target states into one state s_{f} so that $\mathcal{S}^* = \mathcal{S} \setminus \mathcal{S}_{\mathrm{target}} \cup \{s_{\mathrm{f}}\}$. For $s, s' \in \mathcal{S}^* \setminus \{s_{\mathrm{f}}\}$, we update the rate matrix $\mathbf{K}^* : \mathcal{S}^* \to \mathbb{R}_{\geq 0}$ by $\mathbf{K}^*(s, s_{\mathrm{f}}) = \sum_{s'' \in \mathcal{S}_{\mathrm{target}}} \mathbf{K}(s, s'')$, $\mathbf{K}^*(s, s') = \mathbf{K}(s, s')$, and $\mathbf{K}^*(s_{\mathrm{f}}, s)$ is not used in the computation of the MFPT (see Eq. (6)).

Truncated CTMC. Let $\hat{\mathcal{S}} \subseteq \mathcal{S}$ be a subset of the states over the CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\mathrm{target}})$ or detailed-balance CTMC $\mathcal{C}^R = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\mathrm{target}}, \pi)$ and let $\hat{\mathcal{S}}_{\mathrm{target}} \subseteq \hat{\mathcal{S}}$. We construct the rate matrix $\hat{\mathbf{K}} : \hat{\mathcal{S}} \times \hat{\mathcal{S}} \to \mathbb{R}_{\geq 0}$ as

$$\hat{\mathbf{K}}(s, s') = \mathbf{K}(s, s'). \tag{8}$$

We construct the initial probability distribution $\hat{\pi}_0 : \hat{\mathcal{S}} \to [0, 1]$ as

$$\hat{\pi}_0(s) = \frac{\pi_0(s)}{\sum_{s \in \hat{\mathcal{S}}} \pi_0(s)}. \tag{9}$$

We define the truncated CTMC as $\hat{\mathcal{C}} = (\hat{\mathcal{S}}, \hat{\mathbf{K}}, \hat{\pi}_0, \hat{\mathcal{S}}_{\mathrm{target}})$ and $\hat{\mathcal{C}}^R = (\hat{\mathcal{S}}, \hat{\mathbf{K}}, \hat{\pi}_0, \hat{\mathcal{S}}_{\mathrm{target}}, \hat{\pi})$ for $\mathcal{C}$ and $\mathcal{C}^R$, respectively. For a detailed-balance $\hat{\mathcal{C}}^R$, $\hat{\pi} : \hat{\mathcal{S}} \to [0, 1]$ defined as

$$\hat{\pi}(s) = \frac{\pi(s)}{\sum_{s \in \hat{\mathcal{S}}} \pi(s)}, \tag{10}$$

satisfies the detailed balance conditions in $\hat{\mathcal{C}}^R$ and is the unique equilibrium distribution of $\hat{\mathcal{S}}$ in $\hat{\mathcal{C}}^R$ (Whitt, 2006).

2.2. The multistrand kinetic model of interacting nucleic acid strands

Multistrand is a kinetic simulator that is based on SSA for analyzing the folding kinetics of multiple interacting nucleic acid strands. Multistrand can handle both a system of DNA strands and a system of RNA strands.[2]

Interacting nucleic acid strands (reactions). Following Multistrand, we are interested in modeling the interactions of nucleic acid strands in a stochastic regime. In this regime, we have a discrete number of nucleic acid strands (a set called Ψ^*) in a fixed volume V (the "box") and under fixed conditions, such as the temperature T and the concentration of Na^+ and Mg^{2+} cations. This regime can be found in systems that have a small volume with a fixed count of each molecule, and can also be applied to larger volumes when the system is well mixed. Moreover, it can be used to derive reaction rate constants of reactions in a chemical reaction network that follows mass-action kinetics (Schaeffer et al., 2015).

Following Multistrand, a complex is a subset of strands of Ψ^* that are connected through base pairing (see Fig. 2). A complex microstate is the complex base pairs, that is secondary structure. A system microstate is a set of complex microstates, such that each strand $\psi \in \Psi^*$ is part of exactly one complex. A unimolecular reaction with reaction rate constant k_1 (units s^{-1}) has the form

$$A \xrightarrow{k_1} C + D, \tag{11}$$

and a bimolecular reaction at low concentration with reaction rate constant k_2 (units $\mathrm{M}^{-1}\mathrm{s}^{-1}$) can be written in the form of

$$B + F \xrightarrow{k_2} G + H. \tag{12}$$

Each reactant and product is a complex; A, B, F, C and G are nonempty but D and H may be empty complexes. For example, hairpin closing (Fig. 2(a)) is a unimolecular reaction involving one strand, where complexes A and C are comprised of this one strand, while D is empty. Helix dissociation (Fig. 2(b)) is an example of a unimolecular reaction where complex A has two strands while C and D are each of one of these strands. An example of a bimolecular reaction with two reactants and two non-empty products is toehold-mediated three-way strand displacement (Fig. 2(c)). We discuss these type of reactions further in Section 4. We are interested in computing the reaction rate constants of such reactions.

The Multistrand kinetic model. The Multistrand kinetic model is a detailed-balance CTMC $\mathcal{C}^R = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\mathrm{target}}, \pi)$ for a set of interacting nucleic acid strands Ψ^* in a fixed volume V (the "box") and under fixed conditions, such as the temperature T and the concentration of Na^+ and Mg^{2+} cations. The state space $\mathcal{S}$ of the CTMC is the set of all non-pseudoknotted system microstates[3] of the set Ψ^* of interacting strands. The transition rate $\mathbf{K}(s, s')$ is non-zero if and only if s and s' differ by a single base pair.[4] Multistrand distinguishes between unimolecular

[2] Currently, Multistrand does not handle a system of mixed DNA and RNA strands, though it can be extended to handle such systems using good thermodynamic parameters.

[3] A pseudoknotted secondary structure 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. A non-pseudoknotted system microstates does not contain any pseudoknotted secondary structures. Currently, Multistrand excludes pseudoknotted secondary structures due to computationally difficult energy model calculations.

[4] Multistrand allows Watson–Crick base pairs to form, that is A-T and G-C in DNA and A-U and G-C in RNA. Additionally, it provides an option to allow G-T in DNA and G-U in RNA.

transitions, in which the number of strands in each complex remains constant, and bimolecular transitions where this is not the case. There are bimolecular join moves, where two complexes merge, and bimolecular break moves, where a complex falls apart and releases two separate complexes.

The transition rates in the Multistrand kinetic model obey detailed balance as

$$\frac{\mathbf{K}(s,s')}{\mathbf{K}(s',s)} = e^{-\frac{\Delta G^{\circ}_{\text{box}}(s')-\Delta G^{\circ}_{\text{box}}(s)}{RT}}, \tag{13}$$

where $\Delta G^{\circ}_{\text{box}}(s)$ is the free energy of state s (units: kcal mol^{-1}) and depends on the temperature T (units: K) as $\Delta G = \Delta H - T\Delta S$, and $R \approx 1.98\times10^{-3}$ kcal K^{-1} mol^{-1} is the gas constant. The enthalpy ΔH and entropy ΔS are fixed and calculated in the model using thermodynamic models that depend on the concentration of Na^{+} and Mg^{2+} cations and also on a volume-dependent entropy term. The detailed balance condition determines the ratio of rates for reversible transitions. A standard kinetic model that is used in Multistrand to determine the transition rates is the Metropolis kinetic model (Metropolis et al., 1953), where all energetically favorable transitions occur at the same fixed rate and energetically unfavorable transitions scale with the difference in free energy. Unimolecular transition rates are given as

$$\mathbf{K}(s,s') = \begin{cases} k_{\text{uni}} & \text{if}\Delta G^{\circ}_{\text{box}}(s) < \Delta G^{\circ}_{\text{box}}(s'), \\ k_{\text{uni}} e^{-\frac{\Delta G^{\circ}_{\text{box}}(s')-\Delta G^{\circ}_{\text{box}}(s)}{RT}} & \text{otherwise,} \end{cases} \tag{14}$$

and bimolecular transition rates are given as

$$\mathbf{K}(s,s') = \begin{cases} k_{\text{bi}} u & \text{join move,} \\ k_{\text{bi}} e^{-\frac{\Delta G^{\circ}_{\text{box}}(s')-\Delta G^{\circ}_{\text{box}}(s)+\Delta G^{\circ}_{\text{volume}}}{RT}} \times \text{M} & \text{break move,} \end{cases} \tag{15}$$

where u is the concentration of the strands (units: M), $\Delta G^{\circ}_{\text{volume}} = -RT\ln u$, $k_{\text{uni}} > 0$ is the unimolecular rate constant (units: s^{-1}), and $k_{\text{bi}} > 0$ is the bimolecular rate constant (units: M^{-1} s^{-1}). The kinetic parameters $\theta = \{k_{\text{uni}}, k_{\text{bi}}\}$ are calibrated from experimental measurements (Wetmur and Davidson, 1968; Morrison and Stols, 1993).

The distribution π_0 is an initial distribution over the microstates of the reactant complexes, and the set $\mathcal{S}_{\text{target}}$ is a subset of the microstates of the product complexes, which we determine based on the type of the reaction (see Section 4). To set π_0 for unimolecular reactions, we use particular complex microstates. One illustrative example is the (unimolecular) hairpin closing reaction, where we set $\pi_0(h) = 1$ for the system microstate that has no base pairs and $\pi_0(s) = 0$ for all other structures, and $\mathcal{S}_{\text{target}}$ is the system microstate where the strand has a fully formed duplex and a loop. For a bimolecular reaction, when the bimolecular transitions are slow enough between the two complexes, it is valid to assume the complexes each reach equilibrium before bimolecular transitions occur and therefore are Boltzmann distributed (Schaeffer, 2013). Let $\mathcal{CM}$ be the set of all possible complex microstates of a complex B in a volume. A distribution π_b is Boltzmann distributed with respect to complex B if and only if

$$\pi_b(c') = \frac{e^{-\Delta G(c')/RT}}{\sum_{c\in\mathcal{CM}} e^{-\Delta G(c)/RT}} \tag{16}$$

for all complex microstates $c' \in \mathcal{CM}$. In a bimolecular reaction of the form in Eq. (12), for a system microstate s that has complex microstates c and c' corresponding to complexes B and F, we define the initial distribution as $\pi_0(s) = \pi_b(c) \times \pi_b(c')$. For all other states, we define $\pi_0(s) = 0$.

Following the conventions of Multistrand, we estimate the reaction rate constant for a reaction from its MFPT τ_{π_0} (Eq. (7)). For a reaction in the form of Eq. (11),

$$k_1 = \frac{1}{\tau_{\pi_0}}. \tag{17}$$

In the limit of low concentrations for a reaction in the form of Eq. (12),

$$k_2 = \frac{1}{u}\frac{1}{\tau_{\pi_0}}. \tag{18}$$

2.3. Related work

Stochastic simulation approaches. There exist numerous Monte Carlo techniques (Rubino and Tuffin, 2009) for driving simulations towards the target states or to reduce the variance of estimators. For example, importance sampling techniques (Hajiaghayi et al., 2014; Doucet and Johansen, 2009) use an auxiliary sampler to bias simulations, after which estimates are corrected with importance weights. Moreover, many accelerated variants of SSA have been developed for CTMC models of chemically reacting systems (Gillespie, 2007; Cao et al., 2007; Gillespie, 2001; Turner et al., 2004; Sandmann, 2008), which can be adapted to simulate arbitrary CTMCs. There also exists a proliferation of rare event simulation methods for molecular dynamics (Zuckerman and Chong, 2017; Allen et al., 2009; Bolhuis et al., 2002). The ideas behind these methods can be adapted for CTMCs and can be used along with SSA for more efficient computations. For example, in transition path sampling (TPS) (Bolhuis et al., 2002) an ensemble of paths are generated using a Monte Carlo procedure. First, a single path is generated that connects the initial and target states. New paths are then generated by picking random states along the current paths and running time-limited simulations from the states. Sampled states along paths that do not reach the initial or target states are rejected. Even though we could use TPS along with SSA to simulate rare events for CTMCs (Eidelson and Peters, 2012), it is likely that many of the simulated paths require a long simulation time. For example, if the energy landscape has more than one local maximum between the initial and target states, then paths simulated from in between these local maxima could require a long simulation time to reach either the initial or the target states. Moreover, the simulated paths could be correlated and depend on the initial path, and therefore the estimations of different runs could have a high variance. The correlation of paths could be reduced by retaining a fraction of the paths but it would also reduce the computational efficiency.

Stochastic simulations are usually not reusable for the rapid evaluation of perturbed parameters and have to be adapted. This is because the holding times of simulated trajectories need to be updated, which requires that information about all transitions from each sampled state is also stored. Stochastic simulation methods have been to some extent adapted for the rapid evaluation of perturbed parameters. SSA has been adapted in the fixed path ensemble inference approach (Zolaktaf et al., 2019) for parameter estimation. In this approach, an ensemble of paths are generated using SSA and are then compacted and reused for mildly perturbed parameters. To estimate MFPTs, a Monte Carlo approach is used based on expected holding times of states. Despite being useful for parameter estimation in general, this method is not suitable for rare events, because the paths are generated according to SSA. In Section 5.3.3, we use SSA as a baseline method to build truncated CTMCs for MFPT estimation.

Truncation-based approaches. An alternative to sampling methods is to develop a smaller CTMC, whose MFPT well approximates that of the original large CTMC model. As is the case with sampling methods, techniques that have been developed to approximate the continuous state spaces of molecular dynamics simulations can be adapted for this purpose. In the context of predicting protein folding kinetics, the collection of paths produced by TPS has been used to build a so-called Markovian state model (MSM) (Singhal et al., 2004). The MSM is the CTMC obtained by including all states and transitions along the sampled paths; since each state appears once, the MSM is more compact then the underlying set of paths. The MSM approach can easily be adapted to build approximations to large CTMC models, for the purpose of estimating MFPTs and other properties of the CTMC.

The resulting MSM is a truncated CTMC. That is, it contains a subset of the states of the original CTMC, with transitions between states that are adjacent in the original CTMC. In Section 5.3.3, we use TPS as a baseline method to build truncated CTMCs for MFPT estimation.

A probabilistic roadmap is another type of graph-based model, related to our work (Kavraki et al., 1996; Tang et al., 2005). States in a probabilistic roadmap can be selected by random sampling or according to relevant properties, such as having low free energy. Then edges are added to connect nearby (though not necessarily adjacent) states. However, there are some challenges with this method that make it unsuitable for our purposes. First, it is not clear that sampling methods based on state (as opposed to path) properties will include important states on the most likely folding trajectories from initial to target states. Another challenge is determining appropriate transition rates between states that are not adjacent in the CTMC model.

Error estimation. Another important problem in CTMCs is computing transient probabilities, that is the probability distribution of the states over time. Transient probabilities can be computed exactly with the master equation (Van Kampen, 1992) for CTMCs that have a feasible state space size. An important tool that has been developed to quantify the error of transient probability estimations for truncated CTMCs is the finite state projection (FSP) method (Munsky and Khammash, 2006). The FSP method tells us that as the size of the state space of the truncated CTMC grows, the approximation monotonically improves and provides upper and lower bounds on the true transient probabilities. As the authors of the FSP method mention, there are many ways to grow the state space, for example by iteratively adding states that are reachable from the already-included states within a fixed number of steps. There have been many attempts to enumerate a suitable set of states that provides good approximations while being small enough that transient probabilities can be computed efficiently (Dinh and Sidje, 2016). In the Krylov-FSP-SSA approach (Sidje and Vo, 2015) an SSA approach is used to drive the FSP and adaptive Krylov methods are used to efficiently evaluate the matrix exponential for transient probability estimation. In brief, the method starts from an initial state space and proceeds iteratively in three steps. First, it drops states that have become improbable. Second, it runs SSA from each state of the remaining state space to incorporate probable states. Third, it adds states that are reachable within a fixed number of steps. Despite its great potential, this way of building the state space may not be suitable for estimating MFPTs of rare events.

The Krylov-FSP-SSA method has also been used to build truncated CTMCs for the purpose of optimizing parameter sets that are used for transient probability estimation (Dinh and Sidje, 2017). Moreover, in related work (Georgoulas et al., 2017), an ensemble of truncated CTMCs is used to obtain an unbiased estimator of transient probabilities, which are further used for Bayesian inference.

3. The pathway elaboration method

We are interested in efficiently estimating MFPT of rare events in detailed-balance CTMCs and also the rapid evaluation of mildly perturbed parameters. Our approach is to create a reusable in-memory representation of CTMCs, which we call a truncated CTMC, and to compute the MFPTs through matrix equations (Eqs. (6) and (7)).

We propose the *pathway elaboration* method for building a truncated detailed-balance CTMC $\hat{\mathcal{C}}^R$ for a detailed-balance CTMC $\mathcal{C}^R$. We call this approach the pathway elaboration method as we build a truncated CTMC by elaborating an ensemble of prominent paths in the system. The method has three main steps to build a truncated CTMC, and an additional step for the rapid evaluation of perturbed parameters.

1. The "pathway construction" step uses biased simulations to find an ensemble of short paths from the initial states to the target states. This step is inspired by importance sampling (Madras, 2002; Rubino and Tuffin, 2009; Andrieu et al., 2003; Hajiaghayi et al., 2014) and exploration–exploitation trade-offs (Sutton and Barto, 2018).

Algorithm 1: The pathway elaboration method.

```
Function PathwayElaboration($\mathcal{C}^R$, $N$, $\beta$, $K$, $\kappa$, $\pi'$)
    $(\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}}, \pi) = \mathcal{C}^R$
    $\mathcal{S}_0 \leftarrow$ ConstructPathway($\mathcal{C}^R$, $N$, $\beta$, $\pi'$)
    $\hat{\mathcal{S}} \leftarrow \mathcal{S}_0$
    for $s \in \mathcal{S}_0$ do
        $\mathcal{S}' \leftarrow$ ElaborateState($s, \mathcal{C}^R$,K, $\kappa$)  // Run SSA $K$
         times from $s$ with a time limit of $\kappa$ and return
         the visited states.
        $\hat{\mathcal{S}} \leftarrow \hat{\mathcal{S}} \cup \mathcal{S}'$
    $\hat{\mathbf{K}} \leftarrow$ Construct rate matrix from $\hat{\mathcal{S}}$ and $\mathbf{K}$  // Eq. (8).
    return $\hat{\mathcal{C}}^R = (\hat{\mathcal{S}}, \hat{\mathbf{K}}, \hat{\pi}_0, \hat{\mathcal{S}}_{\text{target}}, \hat{\pi})$
     // For $\hat{\pi}_0$ and $\hat{\pi}$, see Eq. (9) and Eq. (10),
     respectively.
Function ConstructPathway($\mathcal{C}$, $N$, $\beta$, $\pi'$)
    $(\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}}) = \mathcal{C}$
    $\mathcal{S}_0 \leftarrow \emptyset$
    for n = 1 to N do
        Sample $s \sim \pi_0$
        $\mathcal{S}_0 \leftarrow \mathcal{S}_0 \cup \{s\}$
        Sample $s_b \sim \pi'$
        for t =1,2, ... do
            if $s = s_b$ then break
            Sample $z \sim$ Uniform$(0,1)$
            if $z < \beta$ then // Bias simulations towards $s_b$
             using Eq. (19).
                Sample $s'|s \sim \mathbf{P}(\cdot|X_{t-1} = s)$
            else
                Sample $s'|s \sim \check{\mathbf{P}}_{s_b}(\cdot|X_{t-1} = s)$
            $\mathcal{S}_0 \leftarrow \mathcal{S}_0 \cup s'$
            $s \leftarrow s'$
    return $\mathcal{S}_0$
```

2. The "state elaboration" step uses SSA from every state in the pathway to add additional states to the pathway, with the intention of increasing accuracy. This step is inspired by the string method (Weinan et al., 2002).
3. The "transition construction" step creates a matrix of transitions between every pair of states obtained from the first and second steps.
4. The "δ-pruning" step prunes the CTMC obtained from the previous steps to facilitate the rapid evaluation of perturbed parameters.

These steps result in a truncated detailed-balance CTMC $\hat{\mathcal{C}}^R = (\hat{\mathcal{S}}, \hat{\mathbf{K}}, \hat{\pi}_0, \hat{\mathcal{S}}_{\text{target}}, \hat{\pi})$. Fig. 1, parts (**a**) to (**d**), illustrates the key steps of the pathway elaboration method, and Algorithm 1 provides high-level pseudocode. We next describe these steps in detail.

Pathway construction. We construct a pathway by biasing N SSA simulations towards the target states. We bias a simulation by using the shortest-path distance function $d : \mathcal{S} \times \mathcal{S}_{\text{target}} \to \mathbb{R}_{\geq 0}$ from every state $s \in \mathcal{S}$ to a fixed target state $s_b \in \mathcal{S}_{\text{target}}$ (Kuehlmann et al., 1999; Hajiaghayi et al., 2014). For every biased path, we can use a different s_b. Therefore, in general, we can sample s_b from a probability distribution π' over the target states. Given s_b, we use an exploitation–exploration trade-off approach. At each transition, the process randomly based on a threshold β chooses to either decrease the distance to s_b or to explore the region based on the actual probability matrix of the transitions.

Let $\mathcal{D}_{s_b}(s)$ be the set of all neighbors of s whose distance with s_b is one less than the distance of s with s_b, and let $\mathbf{P}(s, s')$ be as in Eq. (1).

Instead of sampling states according to $\mathbf{P}$, we use $\tilde{\mathbf{P}} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ where

$$\tilde{\mathbf{P}}(s, s') = \begin{cases} \mathbf{P}(s, s') = \frac{\mathbf{K}(s,s')}{\sum_{s'' \in \mathcal{S}} \mathbf{K}(s,s'')} & 0 \leq z \leq \beta, \\ \check{\mathbf{P}}_{s_b}(s, s') = \frac{\mathbf{K}(s,s')\mathbf{1}\{s' \in \mathcal{D}_{s_b}(s)\}}{\sum_{s'' \in \mathcal{S}} \mathbf{K}(s,s'')\mathbf{1}\{s'' \in \mathcal{D}_{s_b}(s)\}} & \beta < z \leq 1. \end{cases} \tag{19}$$

Here z is chosen uniformly at random from $[0, 1]$, β is a threshold, and $\mathbf{1}\{.\}$ is an indicator function that is equal to 1 if the condition is met and 0 otherwise. When $\beta = 1$, then $\tilde{\mathbf{P}}(s, s') = \mathbf{P}(s, s')$.

In Proposition 3.1, we show that if $\beta < 1/2$, then biased paths will reach target states in an expected number of steps that is linear in the distance from initial to target states. Lower values of β help the process reach the target states more quickly, but larger values of β help the process explore the state space. Using values of $\beta \geq 1/2$ is also useful in practice, as we have done in Section 5. However, in this work, for $\beta \geq 1/2$ we have not established a bound on the expected number of steps to reach the target states. When $\beta \rightarrow 1$, the pathway construction step will perform as SSA.

Proposition 3.1. *Let d_{max} be the maximum distance from a state in a CTMC to target state s_b. Then when $0 \leq \beta < 1/2$, the expected length of a pathway that is sampled according to Eq. (19) is at most $\frac{d_{max}}{1-2\beta}$.*

Proof. Based on the distance of states with s_b, we can project a biased path that is generated with Eq. (19) to a 1-dimensional random walk R, where coordinate $x = 0$ corresponds to s_b and coordinate $x > 0$ corresponds to all states $s \neq s_b$ with $d(s, s_b) = x$. From the definition of $\tilde{\mathbf{P}}$ and since all states have a path to s_b by a transition to a neighbor state that decreases the distance by one, at each step, the random walk either takes one step closer to $x = 0$ with probability at least $1 - \beta$ or one step further from $x = 0$ with probability at most β. If we let $E(R, k)$ denote the expected time for random walk R to reach 0 from k, then we have that when $0 \leq \beta < 1/2$,

$$E(R, k) \leq \frac{k}{1 - 2\beta}, \tag{20}$$

which follows from classical results on biased random walks—see Feller XIV.2 (Feller, 1968). Therefore, if $0 \leq \beta < 1/2$, the proposition holds, and the state space built with N biased paths from the initial state s_0 to a target state s_b has expected size

$$\mathbb{E}[|\hat{\mathcal{S}}|] \leq \frac{N \cdot d(s_0, s_b)}{1 - 2\beta} \leq \frac{N \cdot d_{\max}}{1 - 2\beta}. \tag{21}$$

If for each biased path, the initial state is sampled from π_0 and the target state is sampled from π', then we sum over the N sampled (initial state, target state) pairs, and the total expected state space size is still bounded by $\frac{N \cdot d_{\max}}{1-2\beta}$. □

For efficient computations, we should compute the shortest-path distance efficiently. For elementary step models of interacting nucleic acid strands, we can compute $d(s, s_b)$ by computing the minimum number of base pairs that need to be deleted or formed to convert s to s_b. Multistrand provides a list of base pairings for every complex microstate in a system microstate (state) and we can calculate the distance between two states in a running time of O(b), where b is the number of bases in the strands.

State elaboration. By using Eq. (19), a biased path could have a low probability of reaching a state that has a high probability of being visited with SSA. For example, in some helix association reactions (Zhang et al., 2018), intra-strand base pairs are likely to form before completing hybridization. However, the corresponding states do not lie on the shortest paths from the initial states to the target states. Let c be the minimum number of transitions from s_0 that are required to reach s but which increase the distance to s_b. Let the random walk R be defined as the previous step. Let P_1 denote the probability of reaching s_b before reaching s for this random walk. Following classical results on biased random walks (Feller, 1968), for $\beta \neq 1/2$, $P_1 \geq \frac{(\frac{\beta}{1-\beta})^c - 1}{(\frac{\beta}{1-\beta})^{d_{s_b}(s_0)+c} - 1}$. In the extreme case if $\beta = 0$, then $P_1 = 1$ and the probability of reaching s will be 0.

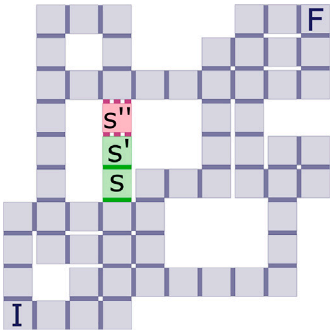


Fig. 3. Example used in the main text for the state elaboration step of pathway elaboration. If in the elaboration step, the simulation finds s and s' but not s'', then without detailed balance, a slow transition from s' to s could result in an overestimation of the MFPT from the initial state I to the target state F. However, in the full state space, s' might quickly reach F via a fast transition to s''.

Therefore, for detailed-balance CTMCs, we elaborate the pathway to possibly include states that have a high probability of being visited with SSA but were not included with our biased sampling. State elaboration with SSA has also been used in related work (Sidje and Vo, 2015) for transient probability estimation. Here, we use SSA to elaborate the pathway constructed from our previous step; we run K simulations from each state of the pathway for a maximum simulation time of κ. A simulation stops as soon as the simulation time becomes greater than κ or reaches a target state. By simulation time we mean the time of a SSA trajectory, not the wall-clock time. The worst-case average running time of elaborating the states in the pathway with this approach is O($|\mathcal{S}_0| K \kappa \mathbf{E}_{\max}(s, s)$), where $\mathcal{S}_0$ is the state space of the pathway constructed from the previous step and $\mathbf{E}_{\max}(s, s)$ is the largest exit rate in the CTMC for which the pathway is being constructed.

“state elaboration” must only be applied with the next step “transition construction” to include both possible forward and backward transitions between neighbor states otherwise only including the forward transitions found with state elaboration will lead to spurious sink states. Sink states that are not a target state make the MFPT to the target states infinite.

Note: We recommend the state elaboration step only for reversible and detailed-balance CTMCs. This is because a trajectory that stops while visiting a non-target state might introduce a spurious sink into the enumerated state space. Specifically, if in this trajectory the last transition is irreversible and the last state was never previously visited, then this last state may become a spurious sink state. For example in Fig. 3, assume that the CTMC has rates $\mathbf{K}(s, s')$ and $\mathbf{K}(s', s'')$ but does not have $\mathbf{K}(s', s)$, and assume that s'' can reach F. Assume also that in the state elaboration step, the simulation finds s and s', but not s'' or any other neighbor of s'. Then without the reverse transition $\mathbf{K}(s', s)$, s' will become a spurious sink state and the MFPT to the target state F will become infinite. Moreover, having reversible transitions that do not obey the detailed balance condition may lead to an overestimate of the MFPT. For example, in Fig. 3 assume that all transitions are reversible, but assume that the reversible transitions between s and s' do not obey detailed balance. Also, assume that $\pi(s)$ and $\pi(s')$ are both high, and that $\mathbf{K}(s, s')$ is large whereas $\mathbf{K}(s', s)$ is small. Then, if the elaboration stops at s' and no other neighbors of s' besides s are discovered, the small value of $\mathbf{K}(s', s)$ will make the MFPT large. However, in the full state space, s' might quickly reach F through a fast transition to s''. Thus, the state elaboration step may not be suitable for non-detailed-balance CTMCs.

Transition construction. The previous two steps produce a state space $\hat{\mathcal{S}}$. Now, for all pairs of states (s, s') in $\hat{\mathcal{S}}$, we set $\hat{\mathbf{K}}(s, s') = \mathbf{K}(s, s')$.

Note that for detailed-balanced CTMC's both forward and backward transitions will be included. This ensures that no spurious sink states are introduced and makes computations more accurate. In the related roadmap planning work (Kavraki et al., 1996; Tang et al., 2005), states are connected to their nearest neighbors as identified by a distance metric. We include all missing transitions by checking for every state in $\hat{S}$ whether its neighbors are also in $\hat{S}$ in $O(|\hat{S}|m\})$ time, where m is the maximum number of neighbors of the states in the original CTMC.

δ-pruning. Given a (truncated) CTMC in which we can compute the MFPT from every state to the target state, one question is: which states and transitions can be removed from the Markov chain without changing the MFPT from the initial states significantly? This question is especially relevant for the rapid evaluation of perturbed parameters, where MFPTs need to be recomputed often.

Given a CTMC $\mathcal{C} = (\mathcal{S}, \mathbf{K}, \pi_0, \mathcal{S}_{\text{target}})$ and a pruning bound δ, let the MFPT from any state s to $\mathcal{S}_{\text{target}}$ be τ_s and let the MFPT from the initial states to $\mathcal{S}_{\text{target}}$ be τ_{π_0}. Let $\mathcal{S}_{\delta p} = \left\{ s \in \mathcal{S} \mid \tau_s < \delta\tau_{\pi_0} \text{ and } \pi_0(s) = 0 \right\}$ be the set of states that are δ-close to $\mathcal{S}_{\text{target}}$ and that are not an initial state. We construct the δ-pruned CTMC $\mathcal{C}_\delta = (\mathcal{S}_\delta, \pi_0, \mathbf{K}_\delta, \{s_d\})$ over the pruned set of states $\mathcal{S}_\delta = \mathcal{S} \setminus \mathcal{S}_{\delta p} \cup \{s_d\}$, where s_d is the new target state. For $s, s' \in \mathcal{S}_\delta \setminus \{s_d\}$, we update the rate matrix $\mathbf{K}_\delta : \mathcal{S}_\delta \rightarrow \mathbb{R}_{\geq 0}$ by $\mathbf{K}_\delta(s, s_d) = \sum_{s' \in \mathcal{S}_{\delta_p}} \mathbf{K}(s, s')$ and $\mathbf{K}_\delta(s, s') = \mathbf{K}(s, s')$. Note that $\mathbf{K}_\delta(s_d, s)$ is not used in the computation of the MFPT (Eq. (6)), so we can simply assume $\mathbf{K}_\delta(s_d, s) = 0$. Alternatively, to retain detailed-balance conditions, we can define the energy of s_d as $E(s_d) = -RT \log \sum_{s'' \in \mathcal{S}_{\delta p}} e^{-\frac{E(s'')}{RT}}$ (see Eqs.7.1 and 7.2 from Schaeffer (2013)) and define $\mathbf{K}_\delta(s_d, s) = e^{-\frac{E(s)-E(s_d)}{RT}} \mathbf{K}_\delta(s, s_d)$. For the pruned CTMC $\mathcal{C}_\delta = (\mathcal{S}_\delta, \pi_0, \mathbf{K}_\delta, \{s_d\})$, let the MFPT $\tau^\delta_{\pi_0}$ be given as usual (Eq. (7)). Then by construction

$$\tau^\delta_{\pi_0} \leq \frac{\tau_{\pi_0}}{1+\delta}. \tag{22}$$

We can calculate the MFPT from every state to the target states by solving Eq. (6) once for CTMC $\mathcal{C}$. Therefore, the running time of δ-pruning depends on the running time of the matrix equation solver that is used. For a CTMC with state space $\mathcal{S}$, the running time of a direct solver is at most $O(|\mathcal{S}|^3)$. For iterative solvers the running time is generally less than $O(|\mathcal{S}|^3)$. After the equation is solved, the CTMC can be pruned in $O(|\mathcal{S}|)$ for any δ.

Updating perturbed parameters. We are interested in rapidly estimating the MFPT to target states given mildly perturbed parameters, for applications such as parameter estimation and functionality estimation as temperature changes. Our approach is to reuse truncated CTMCs for mild parameter perturbations, similar to related work that builds MSM models using TPS sampling (Singhal et al., 2004). In parameter estimation, to minimize bias in the optimized parameters, we can periodically rebuild the truncated CTMCs from scratch, similar to related-work that reuses SSA-generated paths (Zolaktaf et al., 2019). Even though the MFPT estimates may be biased in this way, we could have significant savings in running time by avoiding the cost of sampling and building truncated CTMCs from scratch for every parameter set. In this approach, we would still have to solve Eq. (6), but it could be negligible compared to the other costs. For example in Table 2, on average, solving the matrix equation is faster than SSA by a factor of 47 and is faster than building the truncated CTMC by a factor of 10.

A perturbed thermodynamic model parameter affects the energy of the states. Therefore, to update the transition rates, we would also have to recompute the energy of the states. A perturbed kinetic model only affects the transition rates. A perturbed experimental condition could affect both the energy of the states and the transition rates. Therefore, assuming the energy of a state can be updated in a constant time, the truncated CTMC can be updated in $O(|\hat{\mathcal{S}}| + |\hat{\mathcal{E}}|)$, where $\hat{\mathcal{E}}$ is the set of transitions of the truncated CTMC. For nucleic acid kinetics with elementary steps, the energy of a state can be computed from scratch in $O(b)$ time, or in $O(1)$ time using the energy calculations of a neighbor state (Schaeffer, 2013).

Tuning parameters. In the pathway construction and state elaboration steps, N, β, K, and κ are tuning parameters that affect the quality of prediction. An efficient method to quantify the error of MFPT estimates would be beneficial to set the parameters as we discuss in Section 6. But in its absence, to set these parameters one could use similar values tuned on similar reactions. Alternatively, one could proceed as follows. Initially, we set β by starting with a small number of N biased simulations with $\beta = 0$, then incrementing β up to 1 until the simulations becomes unfeasible. As shown in Proposition 3.1, if we set β to less than $1/2$, then biased paths will reach target states in expected time that is linear in the distance from initial to target states. Similar to SSA, for a fixed β and when $K = 0$ and $\kappa = 0$, we could increase N until the estimated MFPT stops changing significantly (based on the law of large numbers it will converge). Note that for $K = 0$ and $\kappa = 0$ we could compute the MFPT by computing the average of the biased paths without solving matrix equations. For setting K, one possibility is to consider the number of neighbors of each state. A reaction where states have a lot of neighbors requires a larger K compared to a reaction where states have a smaller number of neighbors. κ should be set with respect to K. As stated in Section 5, a large value of κ along with a small value of K could result in excursions that do not reach any target state and lead to overestimates of the MFPT. One could set κ to a small value and then increase K until the MFPT estimate stops changing, and could repeat this process while feasible.

In δ-pruning, for a given bound δ, the running time for solving Eq. (6) for the pruned CTMC $\mathcal{C}_\delta$ might still be high. In that case, a larger value of δ is required. To set δ in practice, it could be useful to consider the number of states that will be pruned for a given δ, that is $|\mathcal{S}_{\delta p}|$.

4. Dataset of interacting DNA strands

Here we describe our dataset of DNA kinetics in which we use in our computational experiments.

The speed at which nucleic acid strands interact is difficult to predict and depends on reaction topology, strands' sequences, and experimental conditions. The number of secondary structures interacting nucleic strands may form is exponentially large in the length of the strands. Typical to these reactions are high energy barriers that prevent the reaction from completing, meaning that long periods of simulation time are required before successful reactions occur. Consider reactions that occur with rates lower than $10000\ \text{M}^{-1}\ \text{s}^{-1}$ such as three-way strand displacement at room temperature (see Table 1). These types of reactions are slow to simulate not because the simulator takes longer to generate trajectories for larger molecules, but the slowness is instead a result of the energy landscape: at low temperatures, duplexes simply are more stable, and require longer simulated time until their dissociation is observed.

We curate a dataset of 267 interacting DNA strands from the published literature, summarized in Table 1. The reactions are annotated with the temperature, the buffer condition, and the experimentally determined reaction rate constant. The dataset covers a wide range of slow and fast unimolecular and bimolecular reactions where the reaction rate constants vary over 8.6 orders of magnitude. For unimolecular reactions, we consider hairpin opening (Bonnet et al., 1998), hairpin closing (Bonnet et al., 1998), and helix dissociation (Cisse et al., 2012). For bimolecular reactions, we consider helix association (Hata et al., 2018; Zhang et al., 2018) and toehold-mediated three-way strand displacement (Machinek et al., 2014). The reactions from Cisse et al. (2012) and Machinek et al. (2014) may have mismatches between the bases of the strands. The type of reactions in Table 1 are widely used in nanotechnology, such as in molecular beacon probes (Chen et al., 2015).

For bimolecular reactions, we Boltzmann sample initial reacting complexes. For reactions in which we define only one target state, in the pathway construction step, we bias the paths towards that state. In this work, for reactions in which we define a set of target states, we

Table 1
Summary of the dataset of 267 nucleic acid kinetics. The initial concentration of the reactants is denoted as u and k is the experimental reaction rate constant.

	Dataset No.	Reaction type & source[a]	# of reactions	Mean # of bases	[Na$^+$] (M)	T (°C)	u (M)	$\log_{10}k$
$\mathcal{D}_{\text{train}}$	1	Hairpin opening (Bonnet et al., 1998)	63	25	[0.15–0.5]	[10–49]	1×10^{-8}	[1.4–4.6]
	2	Hairpin closing (Bonnet et al., 1998)	62	25	[0.15–0.5]	[10–49]	1×10^{-8}	[3.2–4.8]
	3	Helix dissociation (with mismatches) (Cisse et al., 2012)	39	18	[0.01–0.2]	[23–37]	1×10^{-8}	[−1.2–0.9]
	4	Helix association (Hata et al., 2018)	43	46	0.195	25	5×10^{-8}	[4.0–6.7]
	5	Helix association (Zhang et al., 2018)	20	72	0.75	[37–55]	1×10^{-5}	[4.4–7.4]
	6	Toehold-mediated three-way strand displacement (with mismatches) (Machinek et al., 2014)	10	102	0.05[b]	23	[5×10^{-9}–1×10^{-8}]	[5.3–6.8]
$\mathcal{D}_{\text{test}}$	7	Helix association (Hata et al., 2018)	4	46	0.195	25	5×10^{-8}	[4.0–5.0]
	8	Toehold-mediated three-way strand displacement (Machinek et al., 2014)	26	100	0.05[b]	23	[5×10^{-9}–1×10^{-8}]	[2.7–6.3]

[a] See Fig. 2 for example figures of these reactions.
[b] The experiment was performed without Na$^+$ in the buffer.

Table 2
Pathway elaboration ($N = 128$, $\beta = 0.6$, $K = 256$, $\kappa = 16$ ns) versus SSA. The *mean* statistics are averaged over the '# of reactions'. Also, the pathway elaboration experiments are repeated three times and their mean is calculated. MAE refers to the mean absolute error of pathway elaboration with SSA (Eq. (23)). $|\hat{S}|$ is the size of the truncated state space. See Fig. 6 for an illustration of individual reaction predictions.

Dataset No.	# of reactions	MAE	Mean $\lvert\hat{S}\rvert$ for pathway elaboration	Mean matrix computation time (s) for pathway elaboration	Mean computation time (s) for pathway elaboration	Mean computation time (s) for SSA
1	63	0.04	5.7×10^{2}	4.5×10^{-3}	1.0×10^{3}	2.7×10^{1}
2	62	0.03	1.8×10^{3}	1.5×10^{-2}	1.0×10^{3}	1.2×10^{1}
3	39	0.04	5.3×10^{2}	6.8×10^{-3}	1.6×10^{3}	3.8×10^{3}
4	43	0.29	8.1×10^{4}	3.0×10^{1}	2.1×10^{4}	4.9×10^{5}
5	20	0.51	3.8×10^{5}	2.3×10^{4}	1.6×10^{5}	3.7×10^{4}
6	10	0.31	3.0×10^{5}	1.3×10^{3}	1.3×10^{5}	3.8×10^{5}
All datasets	237	0.13	6.0×10^{4}	2.0×10^{3}	2.4×10^{4}	1.1×10^{5}

bias paths towards only one target state, so that $\pi'(s_b) = 1$ for one state and $\pi'(s) = 0$ for all other states. Next we describe these states.

Hairpin closing and hairpin opening. For a hairpin opening reaction, we define the initial state to be the system microstate in which a strand has fully formed a duplex and a loop (see Fig. 2(a)). We define the target state to be the system microstate in which the strand has no base pairs. Hairpin closing is the reverse reaction, where a strand with no base pair forms a fully formed duplex and a loop.

Helix dissociation and helix association. For a helix dissociation reaction, we specify the initial state to be the system microstate in which two strands have fully formed a helix (see Fig. 2(b)). We define the set of target states to be the set of system microstates in which the strands have detached and there are no base pairs within one of the strands. We bias paths towards the target state in which there are no base pairs formed within any of the strands. Helix association is the reverse reaction. We Boltzmann sample the initial reacting complexes in which the strands have not formed base pairs with each other. We define the target state to be the system microstate in which the duplex has fully formed.

Toehold-mediated three-way strand displacement. In this reaction, an invader strand displaces an incumbent strand in a duplex, where a toehold domain facilitates the reaction (see Figs. 4 and 2(c)). We Boltzmann sample initial reacting complexes in which the incumbent and substrate form a complex through base pairing and the invader forms another complex. We define the set of target states to be the set of microstates where the incumbent is detached from the substrate and there are no base pairs within the incumbent. We bias paths towards the target state in which the substrate and invader have fully formed base pairs and there are no base pairs within the incumbent.

In datasets No. 1–6 from Table 1, we consider reactions that are feasible with SSA with our parameterization of Multistrand, given two weeks computation time, since we compare SSA results with pathway elaboration results. We indicate these reactions as $\mathcal{D}_{\text{train}}$ since we also use them as training set in Section 5.4. We indicate datasets No. 7–8 as $\mathcal{D}_{\text{test}}$ since we use them as testing set in Section 5.4.

5. Experiments for interacting DNA strands

We implement pathway elaboration for interacting DNA strands on top of the Multistrand kinetic simulator. In Section 5.1, we describe our experimental setup that is common in our experiments. In Section 5.2, we use pathway elaboration in a case study to gain insight on the kinetics of two contrasting DNA reactions. In Section 5.3, first we evaluate estimations of pathway elaboration by comparing them with estimations of SSA. Then we build truncated CTMCs using SSA and TPS on a subset of our dataset and compare their performance with pathway elaboration. After that, we show the effectiveness of the δ-pruning step. Finally, in Section 5.4, we use pathway elaboration for the rapid evaluation of perturbed parameters in parameter estimation.

5.1. Experimental setup

Experiments are performed on a system with 64 2.13 GHz Intel Xeon processors and 128 GB RAM in total, running openSUSE Leap 15.1. An experiment for a reaction is conducted on one processor. Our framework is implemented in Python, on top of the Multistrand kinetic simulator. To solve the matrix equations in Eq. (6), we use the sparse direct solver from SciPy (Virtanen et al., 2020) when possible.[5] Otherwise we use the sparse iterative biconjugate gradient algorithm (Fletcher, 1976) from SciPy.

In all of our experiments, the thermodynamic parameters for predicting the energy of the states are fixed and the energies are calculated with Multistrand. Each reaction uses its own experimental condition as provided in the dataset. In all our experiments, we use the Metropolis kinetic model from Multistrand. For all experiments except for Section 5.4, we fix the kinetic parameters to the Metropolis Mode parameter set (Zolaktaf et al., 2017), that is $\theta_1 = \{k_{\text{uni}} \approx 2.41 \times 10^6\ \text{s}^{-1}, k_{\text{bi}} \approx 8.01 \times 10^5\ \text{M}^{-1}\text{s}^{-1}\}$. To obtain MFPTs with SSA, we use

[5] The implementation we used allowed the sparse direct solver to use only up to 2 GB of RAM.

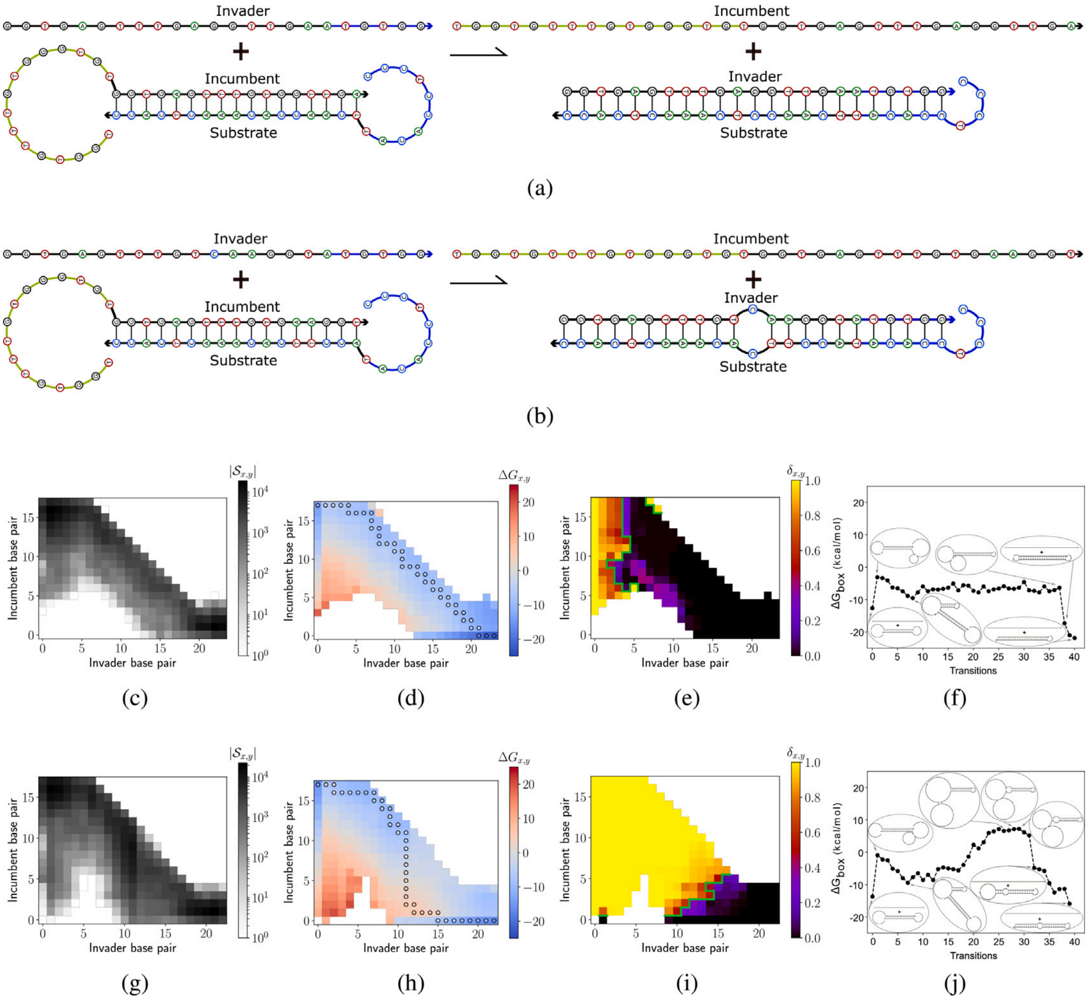


Fig. 4. Results of truncated CTMCs built with pathway elaboration (N = 128, β = 0.6, K = 1024, κ = 16 ns) for two toehold-mediated three-way strand displacement reactions from Machinek et al. (2014). (**a**) A toehold-mediated three-way strand displacement reaction that has a 6-nt toehold and a 17-nt displacement domain (Machinek et al., 2014). (**b**) A toehold-mediated three-way strand displacement reaction that has a 6-nt toehold, a 17-nt displacement domain, and a mismatch exists between the invader and the substrate at position 6 of the displacement domain (Machinek et al., 2014). Figs. 4(c), 4(d), 4(e), and 4(f) correspond to Fig. 4(a). Figs. 4(g),4(h),4(i), and 4(j) correspond to Fig. 4(b). In Figs. 4(c), 4(d), 4(e), 4(g), 4(h), and 4(i), the x-axis corresponds to the number of base pairs between the invader and the substrate, and the y-axis corresponds to the number of base pairs between the incumbent and the substrate. (**c, g**) At coordinate (x, y), $|S_{x,y}|$ is shown, where $S_{x,y}$ is a system macrostate (a nonempty set of system microstates) equal to the set of states with coordinate (x, y). (**d, h**) At coordinate (x, y), the free energy $\Delta G_{x,y}$ is shown, which is defined as $\Delta G_{x,y} = -RT \ln \sum_{s\in S_{x,y}} e^{\frac{-\Delta G(s)}{RT}}$ (Schaeffer, 2013). The free energy of the paths in Figs. 4(f) and 4(j) are also shown with the ∘ marker in Figs. 4(d) and 4(h), respectively. (**e, i**) At coordinate (x, y), the value of $\delta_{x,y} = \sum_{s\in S_{x,y}} \frac{w_s \delta(s)}{\sum_{s\in S_{x,y}} w_s}$ is shown, where $\delta(s) = \tau_s/\tau_{\pi_0}$ and $w_s = e^{\frac{-\Delta G(s)}{RT}}$. For ease of understanding, the green "halfway line" separates coordinates where $\delta_{x,y}$ is greater than 0.5 from coordinates where $\delta_{x,y}$ is less than 0.5. (**f, j**) The free energy landscape of a random path built with pathway elaboration (N = 1, β = 0, K = 0, κ = 0 ns) and the initial and the final states and some states near the local extrema are illustrated.

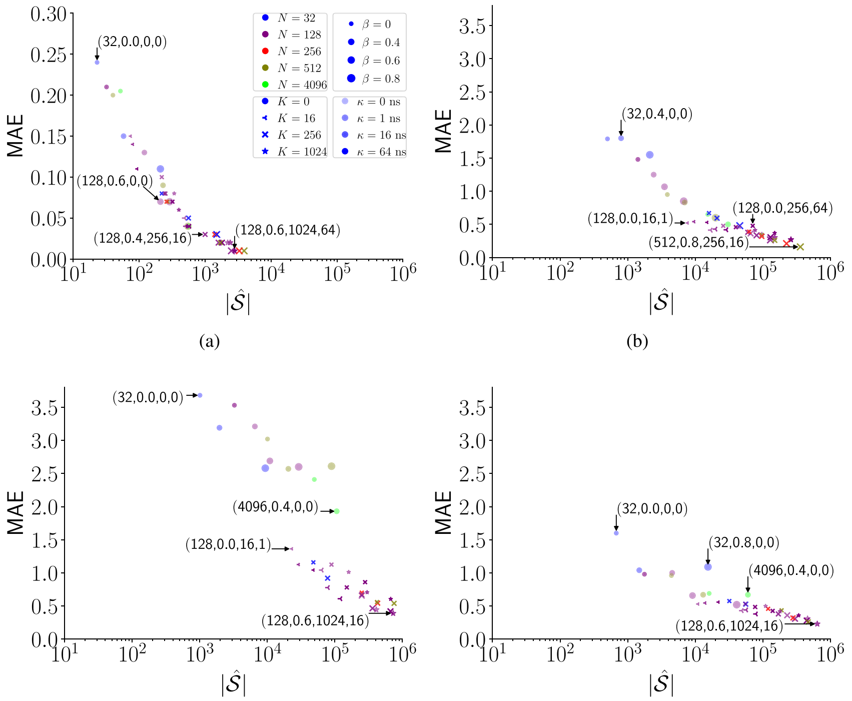
1000 samples, except for three-way strand displacement reactions in which we use 100 samples, since the simulations take a longer time to complete.

5.2. Case study

Here we illustrate the use of pathway elaboration to gain insight on the kinetics of two contrasting reactions from Machinek et al. (2014), one being a rare event.

Figs. 4(a) and 4(b) show the two toehold-mediated three-way strand displacement reactions that we consider (Machinek et al., 2014). In the reaction in Fig. 4(a), the invader and substrate are complementary strands in the displacement domain. In the reaction in Fig. 4(b), there is a mismatch between the invader and the substrate in the displacement domain. The rate of toehold-mediated strand displacement is usually determined by the time to complete the first bimolecular transition, in which the invader forms a base pair with the substrate for the first time. However, the rate could be controlled by several orders of magnitude by altering positions across the strand, such as using mismatch bases (Machinek et al., 2014). The reaction in Fig. 4(b) is approximately 3 orders of magnitude slower than the reaction in Fig. 4(a). For the reaction in Fig. 4(a), $\log_{10} k$ = 6.43, $\log_{10} \hat{k}_{\text{PE}}$ = 6.62, $\log_{10} \hat{k}_{\text{SSA}}$ = 6.75, $|\hat{S}|$ = 4.3 × 10^5, the computation time of pathway elaboration is 1.4×10^5 s, and the computation time of SSA is 3.9×10^5 s. For the reaction in Fig. 4(b), $\log_{10} k$ = 3.17, $\log_{10} \hat{k}_{\text{PE}}$ = 3.59, $|\hat{S}|$ = 7×10^5, the computation time of pathway elaboration is 2.7 × 10^5 s, and SSA is not feasible within 1 × 10^6 s.

In Figs. 4(c)–4(e) and 4(g)–4(i), we illustrate different properties of the truncated CTMCs for the reactions in Figs. 4(a) and 4(b), respectively. Comparing Fig. 4(c) with Fig. 4(g), we see that many states are sampled midway in Fig. 4(g) due to the mismatch. In Figs. 4(d) and 4(h), we compare the energy barrier (increase in free energy) while moving from the beginning of the x-axis towards the end of the x-axis. In Fig. 4(d), we can see a noticeable energy barrier in the beginning. However, in Fig. 4(h), we can see two noticeable energy barriers, one in the beginning and one midway. Figs. 4(e) and 4(i) show states that



(a)

(b)

(c)

(d)

Fig. 5. The MAE of pathway elaboration with SSA versus $|\hat{\mathcal{S}}|$ for different values of N, β, K and κ. (**a**) datasets No. 1,2, and 3, (**b**) dataset No. 4, (**c**) dataset No. 5, and (**d**) dataset No. 6. The annotated values on the figures correspond to N, β, K, and κ, respectively.

are δ-close to the target states. These figures show that with δ-pruning, states that are further from the initial states and closer to the target states will be pruned with smaller values of δ, compared to states that are closer to the initial states and further from the target states. Comparing Fig. 4(e) with Fig. 4(i), the states quickly reach the target states after the first several transitions in Fig. 4(e) (after the energy barrier). However, in Fig. 4(i), the states do not quickly reach the target states until after the second energy barrier. Fig. 4(f) and 4(j) show the free energy landscape and some of the secondary structures for a random path from an initial state to a target state for the reactions in Figs. 4(a) and 4(b), respectively. For the reaction in Fig. 4(a), the barrier is near the first transition. For the reaction in Fig. 4(b), there is a noticeable barrier after several base pairs form between the invader and the substrate, presumably near the mismatch.

5.3. Mean first passage time and reaction rate constant estimation

To evaluate the estimations of pathway elaboration, we compare its estimations with estimations obtained from SSA for the reactions in $\mathcal{D}_{\text{train}}$. Note that for many of these reactions the size of the state space is exponentially large in the length of the strands. Therefore, exact matrix equations is not possible for them. Instead we use SSA since it can generate statistically correct trajectories. We also compare the wall-clock computation time of pathway elaboration with SSA.

We evaluate the estimations of pathway elaboration based on the mean absolute error (MAE) with SSA, which is defined over a dataset $\mathcal{D}$ as

$$\text{MAE} = \frac{1}{|\mathcal{D}|}\sum_{r\in\mathcal{D}} |\log_{10}\hat{\tau}^r_{\text{SSA}} - \log_{10}\hat{\tau}^r_{\text{PE}}| = \frac{1}{|\mathcal{D}|}\sum_{r\in\mathcal{D}} |\log_{10}\hat{k}^r_{\text{SSA}} - \log_{10}\hat{k}^r_{\text{PE}}|, \tag{23}$$

where $\hat{\tau}^r_{\text{PE}}$ and $\hat{\tau}^r_{\text{SSA}}$ are the estimated MFPTs of SSA and pathway elaboration for reaction r, respectively, and $\hat{k}^r_{\text{SSA}}$ and $\hat{k}^r_{\text{PE}}$ are the estimated reaction rate constants of SSA and pathway elaboration for reaction r, respectively. The equality follows from Eqs. (17) and (18). We use $\log_{10}$ differences since the reactions rate constants cover many orders of magnitude. We use the MAE as our evaluation metric since it is conceptually easy to understand. For example, here, an MAE of 1 means on average the predictions are off by a factor of 10. In the rest of this subsection, we first look at the trade-off between the MAEs and the size of the truncated state space set $\hat{\mathcal{S}}$, with regards to different parameter settings of the pathway elaboration method. Then we look at the trade-off between the MAE and the computation time.

5.3.1. MAE of pathway elaboration with SSA versus $|\hat{\mathcal{S}}|$

Fig. 5 shows the MAE of pathway elaboration with SSA versus $|\hat{\mathcal{S}}|$ of pathway elaboration for different configurations of the N, β, K, and κ parameters. Figs. A.1 and A.2 from the Appendix represent Fig. 5 by varying only two parameters at a time. Each subfigure in these figures represent different datasets. The main differences between the systems in Figs. 5(a), 5(b), 5(c), and 5(d) are the average number of bases (as shown in Table 1), the state space size (which depends on the number of bases and sequence of the strands), the sequence of the strands which may lead to the formation of structures that can slow down the reaction, and also the presence of mismatches which also affect the reactions. Fig. 5(a) depicts unimolecular reactions involving relatively short strands, including hairpin closing (Dataset No. 1) and opening (Dataset No. 2) reactions and helix dissociation with mismatches (Dataset No. 3) reactions. These reactions have small state spaces and are relatively easy to simulate with pathway elaboration, which is why we have grouped them together in Fig. 5(a). In contrast, Fig. 5(b) (Dataset No. 4) and Fig. 5(c) (Dataset No. 5) illustrate bimolecular helix association reactions with larger state spaces. These strands may form intermediate structures which hinder the completion of the reaction. The reactions in Figs. 5(b) and 5(c) differ in their design of sequences and also the number of bases per reaction (Dataset No. 5 has 46 bases per reaction and Dataset No. 6 contains 72 bases per reaction). Fig. 5(d) contains three-way strand displacement reactions

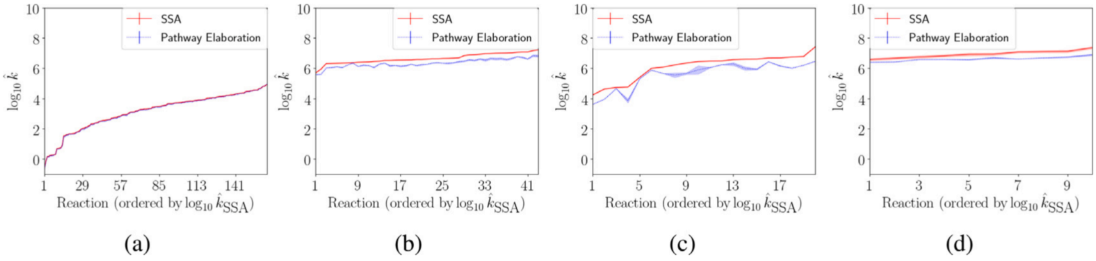


Fig. 6. The $\log_{10}\hat{k}_{\text{SSA}}$ and $\log_{10}\hat{k}_{\text{PE}}$ ($N = 128$, $\beta = 0.6$, $K = 256$, $\kappa = 16$ ns) for (**a**) datasets No. 1,2, and 3, and (**b**) dataset No. 4, (**c**) dataset No. 5, and (**d**) dataset No. 6. The reactions are ordered along the x-axis by their predicted $\log_{10}\hat{k}_{\text{SSA}}$. The pathway elaboration experiments are repeated three times. For each reaction, $\log_{10}\hat{k}_{\text{PE}}$ is calculated by the average of the three experiments. The shaded area for pathway elaboration indicates the range (minimum to maximum) of the three experiments. The shaded area for SSA indicates the 95% percentile bootstrap of the $\log_{10}\hat{k}_{\text{SSA}}$.

with mismatches (Dataset No. 6). The average number of nucleotides is 100 per reaction and the reactions have large state spaces.

The figures show that generally as N and β increase, the MAE decreases. This is because for a fixed N as $\beta \to 1$ the ensemble of paths will be generated by SSA. As $N \to \infty$, the truncated state space becomes larger and is more likely to contain the most probable paths from the initial states to the target states.

Comparing the MAE of configurations where $K = 0$ and $\kappa = 0$ with other settings where $K > 0$ and $\kappa > 0$, shows that the elaboration step helps reduce the MAE (in the Appendix, compare Figs. A.1(a)–A.1(d) with Figs. A.1(i)–A.1(l)). Particularly, the elaboration step is useful for Dataset No. 5, helix association from Zhang et al. (2018) where intra-strand base pairs can form before completing hybridization in large state spaces. The plots show that the elaboration step is more useful when β is small (in the Appendix, compare Figs. A.2(a)–A.2(d) with Figs. A.2(i)–A.2(l)). This could be because elaboration helps find rate determining states that were not explored due to the biased sampling. When $\beta \to 1$ the pathway elaboration method will perform as SSA and rate determining states can be found without elaboration.

Furthermore, the figures show that as K increases, the MAE decreases. However, with a large value for κ and a small value of K the performance could be diminished (such as in Fig. A.2(c) of the Appendix). In particular, consider that K and κ might involve simulations that go on excursions outside the ‘main’ densely-visited parts of the enumerated state space, and they might even terminate out there. Such excursions might very well introduce significant local minima into the enumerated state space — even when no significant local minima exist in the original full state space. For example, consider an excursion that goes off-path down a wide slope, perhaps towards the target state. If it terminates before reaching a target state, then a hypothetical simulation in the enumerated state space could get stuck, needing to climb back up the slope to the point where the excursion began. The expected hitting time in the enumerated state space will account for such wasted time, thus leading to an over estimation of the MFPT. Therefore, κ should be tuned with respect to K.

5.3.2. MAE of pathway elaboration with SSA versus computation time

Table 2 illustrates the MAE and the computation time of pathway elaboration for when $N = 128$, $\beta = 0.6$, $K = 256$, and $\kappa = 16$ ns compared with SSA. We illustrate this parameter setting because it provides a good trade-off between accuracy and computational time for the larger reactions. For the smaller reactions, we could achieve the same MAE with less computational time (by using smaller values for the parameter setting). Fig. 6 further shows the prediction of pathway elaboration for this parameter setting compared to the prediction of SSA for individual reactions. In Table 2, the MAE for unimolecular reactions is smaller than 0.05, whereas for bimolecular reactions it is larger than 0.29. This is because the CTMCs for the bimolecular reactions in our dataset are naturally bigger than the CTMCs for the unimolecular reactions in our dataset, and require larger truncated CTMCs. The MAE can be further reduced by changing the parameters (as shown in Fig. 5). With our implementation of pathway elaboration, the computation time of pathway elaboration for datasets No. 3, No. 4, and No. 6 are 2 times, 20 times, 3 times smaller than SSA, respectively. The computation time of SSA for datasets No. 1, No. 2, and No. 5 is smaller than the computation time of pathway elaboration. This is because pathway elaboration has some overhead, and in cases where SSA is already fast it can be slow. However, as we show in Section 5.4, even for these reactions, pathway elaboration could still be useful for the rapid evaluation of perturbed parameters. Also, the computation time for pathway elaboration could be significantly improved with more efficient implementations of the method.

5.3.3. Pathway elaboration versus other truncation-based approaches

In Section 5.3.1, we showed that the state elaboration step of pathway elaboration improves predictions compared to only using biased sampled from initial to target states, because it helps find deep energy basins that strongly influence reaction rates. Here we compare pathway elaboration with two other truncation-based approaches that are applicable for MFPT estimation (explained in Section 2.3). (We do not compare with the probabilistic roadmap method, because for MFPT estimation the target states should be reachable from the initial states and because of the difficulty of determining appropriate transition rates between non-adjacent states as noted in our related work section.) The first truncated CTMC model that we include in our comparison uses SSA to sample paths from initial to target states, and builds a CTMC from these states. We call this method SSA-T, where the “T” stands for truncated. Since the sampled paths from SSA are statistically correct, we want to see whether the estimate obtained by pathway elaboration compares well with the unbiased SSA-T estimates. We compare SSA-T with pathway elaboration only on our first two datasets, since SSA-T, being unsuitable for rare events, is too slow to run on our other datasets with the implementation that we used. Our second truncated CTMC model uses transition path sampling, and so we call it TPS-T. Our implementation of TPS-T first generates a single path that connects the initial and target states, using SSA. Then a new path is generated by choosing a random state in the most recently generated path, and finding a path to the initial or target states from this randomly-chosen state. In our experiments, we generate 128 paths in total for both SSA-T and TSP-T. As in Table 2, for pathway elaboration, we use $N = 128$, $\beta = 0.6$, $K = 256$, and $\kappa = 16$ ns.

Table 3 compares the MAE and computation time of CTMCs that are built with pathway elaboration versus CTMCs that are built with SSA and TPS. Fig. 7 further shows the prediction of these methods compared for individual reactions. Datasets No. 1 and 2 are used in this table which are hairpin opening and closing, respectively. The MAE of

Table 3
Building truncated CTMCs with pathway elaboration versus building truncated CTMCs with SSA (which we call SSA-T) and TPS (which we call TPS-T). For pathway elaboration, $N = 128$, $\beta = 0.6$, $K = 256$, and $\kappa = 16$ ns. For SSA-T and TPS-T, 128 successful simulations are used. The *mean* statistics are averaged over the '# of reactions'. Also, the experiments for each truncation-based approach is repeated three times and their mean is calculated. MAE refers to the mean absolute error of a method with SSA. $|\hat{S}|$ is the size of the truncated state space. The mean matrix computation time for all methods is less than 0.1 (s). See Fig. 7 for an illustration of individual reaction predictions.

Dataset No.	# of reactions	Method	MAE	Mean $\lvert\hat{S}\rvert$	Mean computation time (s)
1	63	Pathway elaboration	0.04	5.7×10^2	1.0×10^3
		SSA-T	0.03	4.0×10^2	1.4×10^5
		TPS-T	0.18	1.6×10^2	2.0×10^4
2	62	Pathway elaboration	0.03	1.8×10^3	1.0×10^3
		SSA-T	0.03	1.7×10^3	1.3×10^4
		TPS-T	0.34	3.2×10^2	1.7×10^3

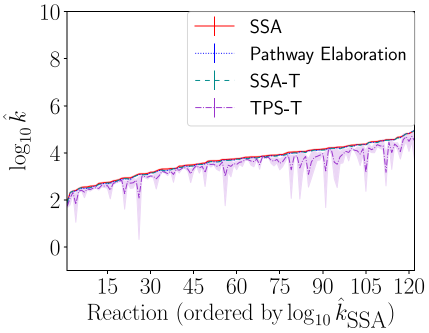


Fig. 7. The $\log_{10}\hat{k}$ of SSA, pathway elaboration, SSA-T, and TPS-T for datasets No. 1 and 2. The reactions are ordered along the x-axis by their predicted $\log_{10}\hat{k}_{\mathrm{SSA}}$. The experiments for each truncation-based approach is repeated three times, where for each reaction, $\log_{10}\hat{k}$ is calculated by the average of the three experiments. The shaded area for each truncation-based approach indicates the range (minimum to maximum) of its three experiments. The shaded area for SSA indicates the 95% percentile bootstrap of the $\log_{10}\hat{k}_{\mathrm{SSA}}$. See Table 3 for parameter settings and mean statistics.

pathway elaboration with SSA (0.04 and 0.03) compares well with the MAE of SSA-T with SSA (0.03 and 0.03). However, the MAE and the variance of TPS-T is high because the paths are correlated and depend on the initial path (Singhal et al., 2004). Increasing the number of simulations would reduce the variance of the predictions.

For a comparison of these methods with pathway elaboration regarding computation time, we have adapted our code for pathway elaboration to implement these methods. As shown in Table 3, in our experiments, the computation time of pathway elaboration is smaller than both SSA-T and TPS-T and the computation time of TPS-T is smaller than the computation time of SSA-T.

5.3.4. δ-Pruning

Fig. 8 shows how δ-pruning affects the quality of the $\log_{10}$ reaction rate constant estimates, the size of the state spaces, and the computation time of solving the matrix equations, for dataset No. 6. The MFPT estimates satisfy the bound given by Eq. (22) whilst δ-pruning reduces the computation time for solving the matrix equations by an order of magnitude for $\delta = 0.6$. Using larger values of δ we can further decrease the computation time. If we reuse the CTMCs many times, such as in parameter estimation, δ-pruning could help reduce computation time significantly.

5.4. Parameter estimation

In the previous subsections the underlying parameters of the CTMCs were fixed. Here we assume the parameters of the kinetic model of the CTMCs are not calibrated and we use pathway elaboration to build truncated CTMCs to rapidly evaluate perturbed parameter sets during parameter estimation. We use the 237 reactions indicated as $\mathcal{D}_{\mathrm{train}}$ in Table 1 as our training set. We use the 30 rare event reactions indicated as $\mathcal{D}_{\mathrm{test}}$ in Table 1 to show that given a well-calibrated parameter set for the CTMC model, the pathway elaboration method can estimate MFPTs and reaction rate constants of reactions close to their experimental measurement.

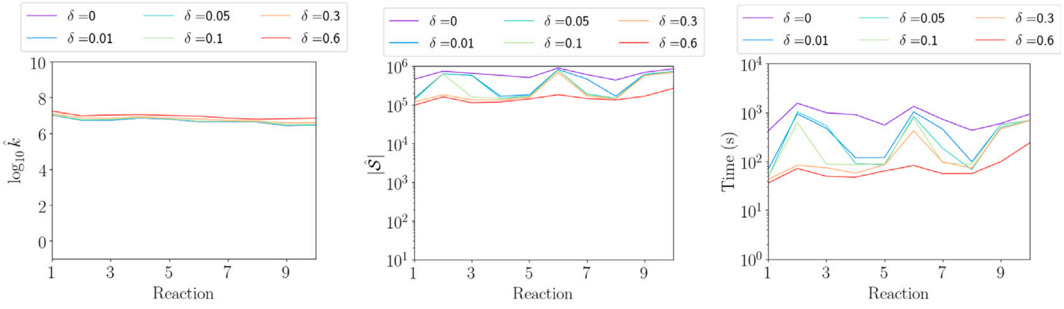
We seek the parameter set that minimizes the mean squared error (MSE) as

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_{\mathrm{train}}|} \sum_{r \in \mathcal{D}_{\mathrm{train}}} (\log_{10}\tau^r - \log_{10}\hat{\tau}^r_{\mathrm{PE}}(\theta))^2 = \\ \underset{\theta}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_{\mathrm{train}}|} \sum_{r \in \mathcal{D}_{\mathrm{train}}} (\log_{10}k^r - \log_{10}\hat{k}^r_{\mathrm{PE}}(\theta))^2, \tag{24}$$

which is a common cost function for regression problems. The equality follows from Eqs. (17) and (18). We use the Nelder–Mead optimization algorithm (Nelder and Mead, 1965; Virtanen et al., 2020) to minimize the MSE. We initialize the simplex in the algorithm with $\theta_2 = \{k_{\mathrm{uni}} = 5 \times 10^4\ \mathrm{s}^{-1}, k_{\mathrm{bi}} = 5 \times 10^4\ \mathrm{M}^{-1}\mathrm{s}^{-1}\}$ in which we choose arbitrarily and two perturbed parameter sets. Each perturbed parameter set is obtained from θ_2 by multiplying one of the parameters by 1.05, which is the default implementation of the optimization software (Virtanen et al., 2020). For every reaction, we also initialize the Multistrand kinetic model with θ_2. We build truncated CTMCs with pathway elaboration ($N = 128$, $\beta = 0.4$, $K = 256$, $\kappa = 16$ ns). Whenever the matrix equation solving time is large (here we consider a time of 120 s large), we use δ-pruning (here we use δ values of $0.01 - 0.6$) to reduce the time. During the optimization, for a new parameter set we update the parameters in the kinetic model of the truncated CTMCs and we reuse the truncated CTMC to evaluate the parameter set. To reduce the bias and to ensure that the truncated CTMCs are fair with respect to the optimized parameters, we can occasionally rebuild truncated CTMCs from scratch.

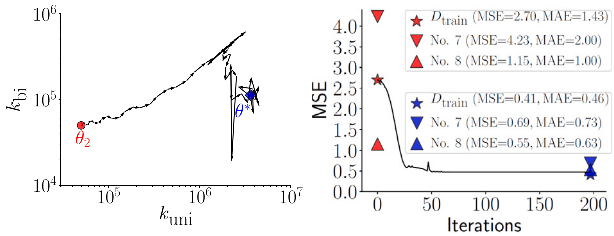
Although we use the MSE of pathway elaboration with experimental measurements as our cost function in the optimization procedure, the MAE of pathway elaboration with experimental measurements also decreases. Fig. 9 shows how the parameters, the MSE, and the MAE change during optimization. The markers are annotated with the MSE and the MAE of $\mathcal{D}_{\mathrm{train}}$ and datasets No. 7–8 when truncated CTMCs are built from scratch. The MAE of $\mathcal{D}_{\mathrm{train}}$ with the initial parameter set θ_2 is 1.43. The optimization finds $\theta^* = \{k_{\mathrm{uni}} \approx 3.61 \times 10^6\ \mathrm{s}^{-1}, k_{\mathrm{bi}} \approx 1.12 \times 10^5\ \mathrm{M}^{-1}\ \mathrm{s}^{-1}\}$ and reduces the MAE of $\mathcal{D}_{\mathrm{train}}$ to 0.46. The MAE of dataset No. 7 and dataset No. 8, which are not used in the optimization, reduce from 2.00 to 0.73 and from 1.00 to 0.63, respectively.

Overall, the experiment in this subsection shows that pathway elaboration enables MFPT estimation of rare events. It predicts their MFPTs close to their experimental measurements given an accurately calibrated model for their CTMCs. Moreover, it shows that pathway elaboration enables the rapid evaluation of perturbed parameters and makes feasible tasks such as parameter estimation which benefit from such methods. On average for the 30 reactions in the testing set, pathway elaboration takes less than two days, whereas SSA is not feasible within two weeks. The entire experiment in Fig. 9 takes less than five days parallelized on 40 processors. Note that clearly our optimization procedure could be improved, for example by using a larger dataset or a more flexible kinetic model. However, this experiment is a preliminary study; we leave a rigorous study on calibrating nucleic acid kinetic models with pathway elaboration and possible improvements to future studies.



(a) (b) (c)

Fig. 8. The effect of δ-pruning with different values of δ on truncated CTMCs that are built with pathway elaboration ($N = 128$, $\beta = 0.6$, $K = 1024$, $\kappa = 16$ ns) for dataset No. 6. $\delta = 0$ indicates δ-pruning is not used. **(a)** The $\log_{10}\hat{k}$. **(b)** The size of the truncated state space $|\hat{S}|$. **(c)** The computation time for solving Eq. (6).



(a) (b)

Fig. 9. Results of parameter estimation using pathway elaboration ($N = 128$, $\beta = 0.4$, $K = 256$, $\kappa = 16$ ns). **(a)** The parameters are optimized from an initial simplex of θ_2 and its perturbations to $\theta^* = \{k_{\text{uni}} \approx 3.61 \times 10^6\ \text{s}^{-1}, k_{\text{bi}} \approx 1.12 \times 10^5\ \text{M}^{-1}\ \text{s}^{-1}\}$. **(b)** The parameters are optimized using $\mathcal{D}_{\text{train}}$, shown with a line graph, and evaluated on dataset No. 7 and No. 8. The red markers at iteration 0 are annotated with the MSE and MAE of the datasets when the truncated CTMCs are built from scratch using the initial parameter set θ_2. The same truncated CTMCs are used in the optimization until iteration 200. At iteration 200, we build new CTMCs with the optimized parameter set θ^*. The blue markers at iteration 200 are annotated with the MSE and MAE of the datasets when the truncated CTMCs are built from scratch using the optimized parameter set θ^*.

6. Discussion

In this work, we address the problem of estimating MFPTs of rare events in CTMC models of nucleic acid reactions and also the rapid evaluation of perturbed parameters. To this end, we propose the pathway elaboration method, a time-efficient probabilistic truncation-based CTMC approach. We conduct computational experiments on a wide range of experimental measurements to show pathway elaboration could provide reasonable estimates of rare-events' MFPTs in small runtimes, and shows promise as a practical approach for parameter optimization. In summary, our results are promising, but there is still room for improvement.

Using pathway elaboration, in the best possible case, the sampled region of states and transitions is obtained faster than SSA, but without significant bias in the collected states and transitions. The sampled region may however qualitatively differ from what would be obtained from SSA, which may compromise the MFPT estimates. Moreover, reusing truncated CTMCs for significantly perturbed parameters could lead to inaccurate estimation of the MFPT in the original CTMC. Hence, a method to quantify the error of MFPT estimates when experimental measurements are not available would be beneficial. It would help us set values for N, β, K and κ for fixed model parameters, and also evaluate when a truncated CTMC has a high error for perturbed model parameters. One possible approach is to adapt the FSP method that is developed to quantify the error of truncated CTMCs for transient probabilities. We adapt it as follows. We combine all target states into one single absorbing state s_{f}. We project all states that are not in the truncated CTMC into an absorbing state s_o and we redirect all transitions from the truncated CTMC to states out of the CTMC into s_o. Then we use the standard matrix exponential equations to compute the full distribution on the state space at a given time. However, we only care about the probabilities that s_{f} and s_o are occupied. We search to compute the half-completion time $t_{1/2}$ with bounds by

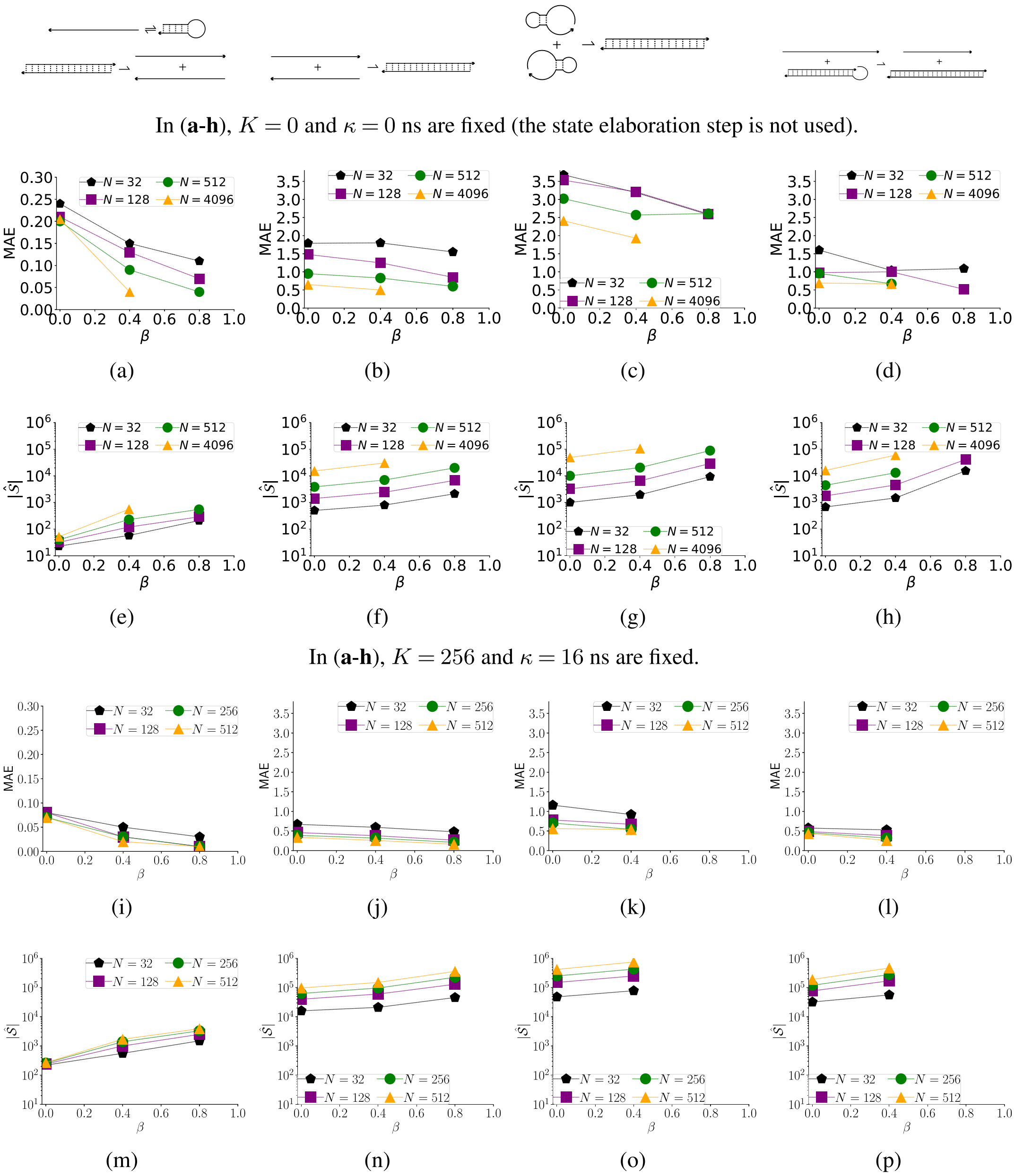
$$\begin{cases} t_{\min} & \text{s.t. } p(s_{\text{f}}\ ;\ t_{\min}) + p(s_o\ ;\ t_{\min}) = \frac{1}{2}, \\ t_{\max} & \text{s.t. } p(s_{\text{f}}\ ;\ t_{\max}) = \frac{1}{2}, \end{cases} \tag{25}$$

where $p(s\ ;\ t)$ is the probability that the process will be at state s at time t starting from the set of initial states. Since s_{f} and s_o are the only absorbing states, then $t_{\min}$ exists and clearly $t_{\min} \leq t_{1/2}$. Based on FSP, $p(s_{\text{f}}\ ;\ t_{\max})$ is an underestimate of the actual probability at time $t_{\max}$, if it exists. A possible way to determine if a solutions exists is to determine the probability of reaching state s_{f} compared to state s_o from the initial states, which can be calculated by solving a system of linear equations (see Eq. 2.13 from Metzner et al. (2009)). If the probability is greater or equal to $\frac{1}{2}$ then a solutions exists. If a solution does not exist for the given statespace, then based on FSP the error is guaranteed to decrease by adding more states and we can eventually find a solution to Eq. 6. The search for $t_{\max}$ can be completed with binary search. Thus, the true $t_{1/2}$ is guaranteed to satisfy $t_{\min} \leq t_{1/2} \leq t_{\max}$. For exponential decay processes, the relation between the half-completion time and the MFPT is (Cohen-Tannoudji et al., 1977; Simmons, 1972)

$$t_{1/2} = \frac{\ln 2}{\lambda} \text{ and } \tau = \frac{1}{\lambda} \rightarrow \tau = \frac{t_{1/2}}{\ln 2}, \tag{26}$$

where λ is the rate of the process. Thus, $\frac{t_{\min}}{\ln 2} \leq \tau \leq \frac{t_{\max}}{\ln 2}$. A drawback of this approach is that we might need a large number of states to find a solution to Eq. 6, which might make the master equation or the linear system solver infeasible in practice. Efficiently quantifying the error of MFPT estimates in truncated CTMCs for exponential and non-exponential decay processes is beyond the scope of this paper. It might be possible to use some other existing work (Kuntz et al., 2019; Backenköhler et al., 2019).

In the pathway elaboration method, we estimate MFPTs by solving matrix equations. Thus, its performance depends on the accuracy and speed of matrix equation solvers. For example, applying matrix equation solvers may not be suitable if the initial states lie very far from the target states, since the size of the truncated CTMCs depends on the shortest-path distance between these states. Although solving matrix equations through direct and iterative methods has progressed, both theoretically and practically (Fletcher, 1976; Virtanen et al., 2020; Cohen et al., 2018), solving stiff (multiple time scales) or very large



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. A.1. The effect of pathway construction with different values of N and β and fixed values of K and κ on the MAE of pathway elaboration with SSA and the $|\hat{S}|$ of pathway elaboration. In (**a–h**), $K = 0$ and $\kappa = 0$ ns are fixed. $K = 0$ indicates that the states of the pathway are not elaborated. In (**i–p**), $K = 256$ and $\kappa = 16$ ns are fixed. (**a**), (**e**), (**i**), and (**m**) correspond to datasets No. 1,2, and 3. (**b**), (**f**), (**j**), and (**n**) correspond to dataset No. 4. (**c**) (**g**), (**k**), and (**o**) correspond to dataset No. 5. (**d**), (**h**), (**l**), and (**p**) correspond to dataset No. 6. For the missing settings, pathway elaboration did not finish within two weeks computation time.

equations could still be problematic in practice. More stable and faster solvers would allow us to estimate MFPTs for stiffer and larger truncated CTMCs. Moreover, it might be possible to use fast updates for solving the matrix equations (Brand, 2006; Parks et al., 2006). Therefore, if we require to compute MFPT estimates with matrix equations as we monotonically grow the size of the state space or for a perturbed parameter set, the total cost for solving all the linear systems would be the same cost as solving the final linear system from scratch.

We might be able to improve the pathway elaboration method to relieve the limitations discussed above. For example, it might be possible to use an ensemble of truncated CTMCs to obtain an unbiased estimate of the MFPT (Georgoulas et al., 2017). To avoid excursions that lead to overestimation of the MFPT in the state elaboration step, we could run the pathway construction step from the last states visited in the state elaboration step. This would also relax the constraint of having reversible or detailed balance transitions. Presumably, an alternating approach of the two steps would make the approach more flexible. Moreover, currently we run the state elaboration step from every state of the pathway with the same setting. Efficiently running the state elaboration step as necessary, could reduce the time to construct the truncated CTMC in addition to the matrix computation time.

Finally, we used pathway elaboration in a small study to show it is promising for parameter optimization of DNA kinetic models. However, the parameter set obtained in this study requires further calibration on a wider range of reactions and potentially on more flexible kinetic models (Zolaktaf et al., 2017) to improve generalizability. Also, we evaluated the pathway elaboration method in the context of DNA reactions. However, the method is also applicable to RNA kinetics.

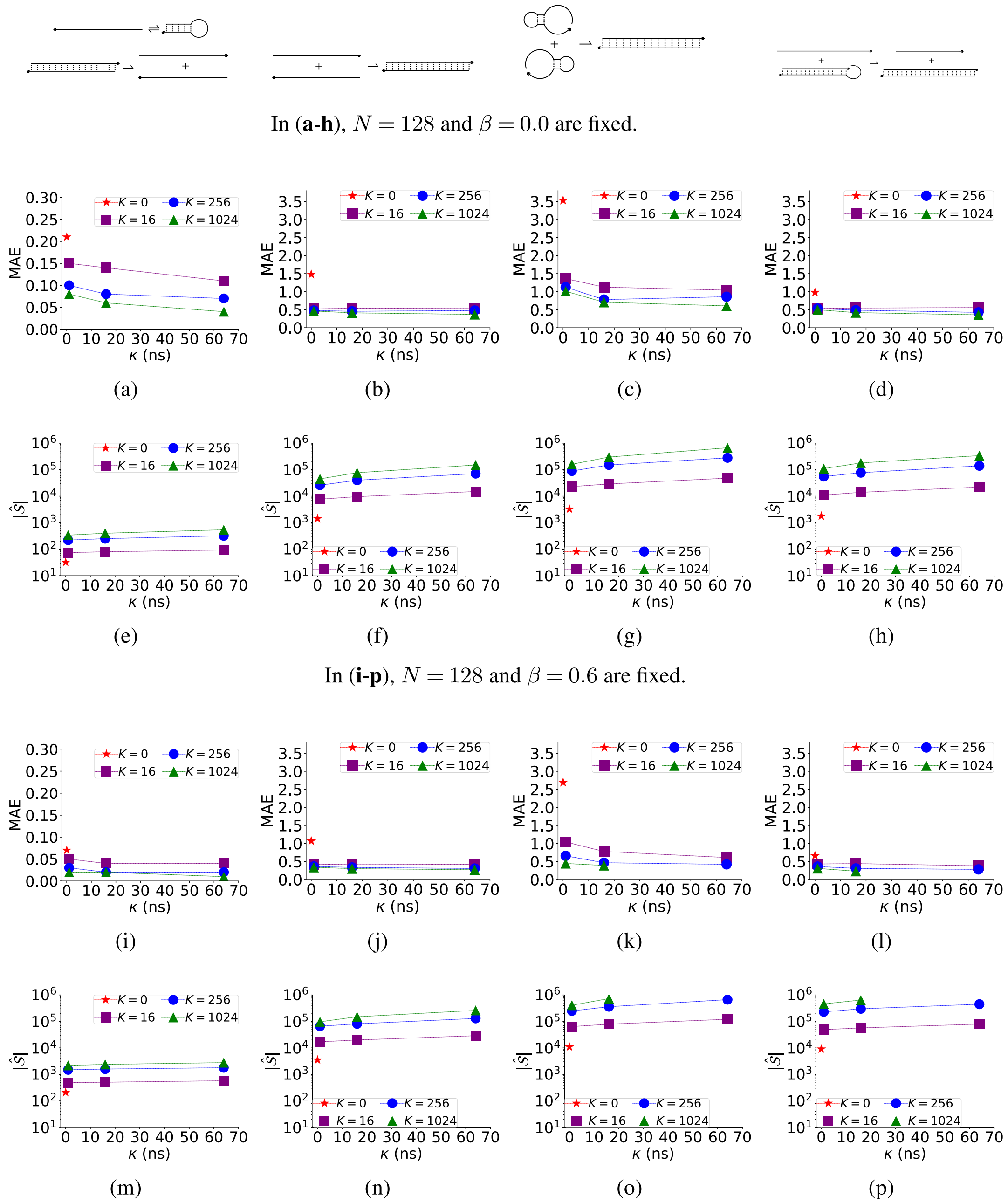


Fig. A.2. The effect of state elaboration, with different values of K and κ and fixed values of N and β on the MAE of pathway elaboration with SSA and the $|\hat{S}|$ of pathway elaboration. $K = 0$ indicates that the states of the pathway are not elaborated. In (**a–h**), $N = 128$ and $\beta = 0.0$ are fixed. In (**i–p**), $N = 128$ and $\beta = 0.6$ are fixed. (**a**), (**e**), (**i**), and (**m**) correspond to datasets No. 1,2, and 3. (**b**), (**f**), (**j**), and (**n**) correspond to dataset No. 4. (**c**) (**g**), (**k**), and (**o**) correspond to dataset No. 5. (**d**), (**h**), (**l**), and (**p**) correspond to dataset No. 6. For the missing settings, pathway elaboration did not finish within two weeks computation time.

Moreover, it is applicable to other detailed-balance CTMC models, such as chemical reaction networks (Anderson and Kurtz, 2011) and protein folding (McGibbon and Pande, 2015).

CRediT authorship contribution statement

Sedigheh Zolaktaf: Formulating the solution, Incorporating the data, Conducting experimental studies, Writing the paper. **Frits Dannenberg:** Formulating the solution, Incorporating the data, Conducting and guiding experimental studies, Writing the paper. **Mark Schmidt:** Guiding the experimental studies, Formulating the solution, Incorporating the data, Writing the paper. **Anne Condon:** Guiding the experimental studies, Formulating the solution, Incorporating the data, Writing the paper. **Erik Winfree:** Guiding the experimental studies, Formulating the solution, Incorporating the data, Writing the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the annonymous reviewers for their helpful feedback.

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada RGPIN-2016-04240, the United States National Science Foundation CHE/CCF 1643606, and the Canada CIFAR AI Chair Program.

Appendix. The mean absolute error of the pathway elaboration method for nucleic acid kinetics

Figs. A.1 and A.2 represent Fig. 5 by varying only two parameters at a time.

References

Allen, R.J., Valeriani, C., ten Wolde, P.R., 2009. Forward flux sampling for rare event simulations. J. Phys.: Condens. Matter 21 (46), 463102.

Anderson, D.F., Kurtz, T.G., 2011. Continuous time Markov chain models for chemical reaction networks. In: Design and Analysis of Biomolecular Circuits. Springer, pp. 3–42.

Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I., 2003. An introduction to MCMC for machine learning. Mach. Learn. 50 (1–2), 5–43.

Angenent-Mari, N.M., Garruss, A.S., Soenksen, L.R., Church, G., Collins, J.J., 2020. A deep learning approach to programmable RNA switches. Nature Commun. 11 (1), 1–12.

Azimzadeh, P., Forsyth, P.A., 2016. Weakly chained matrices, policy iteration, and impulse control. SIAM J. Numer. Anal. 54 (3), 1341–1364.

Backenköhler, M., Bortolussi, L., Wolf, V., 2019. Bounding mean first passage times in population continuous-time Markov chains. arXiv preprint arXiv:1910.12562.

Bolhuis, P.G., Chandler, D., Dellago, C., Geissler, P.L., 2002. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. Annu. Rev. Phys. Chem. 53 (1), 291–318.

Bonnet, G., Krichevsky, O., Libchaber, A., 1998. Kinetics of conformational fluctuations in DNA hairpin-loops. Proc. Natl. Acad. Sci. 95 (15), 8602–8606.

Brand, M., 2006. Fast low-rank modifications of the thin singular value decomposition. Linear Algebra Appl. 415 (1), 20–30.

Cao, Y., Gillespie, D.T., Petzold, L.R., 2007. Adaptive explicit-implicit tau-leaping method with automatic tau selection. J. Chem. Phys. 126 (22), 224101.

Chen, Y.-J., Groves, B., Muscat, R.A., Seelig, G., 2015. DNA nanotechnology from the test tube to the cell. Nature Nanotechnol. 10 (9), 748–760.

Cisse, I.I., Kim, H., Ha, T., 2012. A rule of seven in Watson-Crick base-pairing of mismatched sequences. Nat. Struct. Mol. Biol. 19 (6), 623.

Cohen, M.B., Kelner, J., Kyng, R., Peebles, J., Peng, R., Rao, A.B., Sidford, A., 2018. Solving directed laplacian systems in nearly-linear time through sparse LU factorizations. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science. FOCS, IEEE, pp. 898–909.

Cohen-Tannoudji, C., Davies, P.C., Diu, B., Laloe, F., Dui, B., et al., 1977. Quantum Mechanics. Vol. 1, John Wiley & Sons.

Dinh, K.N., Sidje, R.B., 2016. Understanding the finite state projection and related methods for solving the chemical master equation. Phys. Biol. 13 (3), 035003.

Dinh, K.N., Sidje, R.B., 2017. An application of the Krylov-FSP-SSA method to parameter fitting with maximum likelihood. Phys. Biol. 14 (6), 065001.

Doob, J.L., 1942. Topics in the theory of Markoff chains. Trans. Amer. Math. Soc. 52 (1), 37–64.

Doucet, A., Johansen, A.M., 2009. A tutorial on particle filtering and smoothing: Fifteen years later. In: Handbook of Nonlinear Filtering. Vol. 12, (656–704), p. 3.

Eidelson, N., Peters, B., 2012. Transition path sampling for discrete master equations with absorbing states. J. Chem. Phys. 137 (9), 094106.

Feller, W., 1968. An Introduction To Probability Theory and Its Applications, third ed. Vol. 1, Wiley, New York.

Flamm, C., Fontana, W., Hofacker, I.L., Schuster, P., 2000. RNA folding at elementary step resolution. RNA 6 (03), 325–338.

Fletcher, R., 1976. Conjugate gradient methods for indefinite systems. In: Numerical Analysis. Springer, pp. 73–89.

Georgoulas, A., Hillston, J., Sanguinetti, G., 2017. Unbiased Bayesian inference for population Markov jump processes via random truncations. Stat. Comput. 27 (4), 991–1002.

Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81 (25), 2340–2361.

Gillespie, D.T., 2001. Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115 (4), 1716–1733.

Gillespie, D.T., 2007. Stochastic simulation of chemical kinetics. Annu. Rev. Phys. Chem. 58, 35–55.

Hajiaghayi, M., Kirkpatrick, B., Wang, L., Bouchard-Côté, A., 2014. Efficient continuous-time Markov chain estimation. In: International Conference on Machine Learning. pp. 638–646.

Hata, H., Kitajima, T., Suyama, A., 2018. Influence of thermodynamically unfavorable secondary structures on DNA hybridization kinetics. Nucleic Acids Res. 46 (2), 782–791.

Isambert, H., Siggia, E.D., 2000. Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. Proc. Natl. Acad. Sci. 97 (12), 6515–6520.

Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.H., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. 12 (4), 566–580.

Kuehlmann, A., McMillan, K.L., Brayton, R.K., 1999. Probabilistic state space search. In: 1999 IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers (Cat. No. 99CH37051). IEEE, pp. 574–579.

Kuntz, J., Thomas, P., Stan, G.-B., Barahona, M., 2019. The exit time finite state projection scheme: bounding exit distributions and occupation measures of continuous-time Markov chains. SIAM J. Sci. Comput. 41 (2), A748–A769.

Kuntz, J., Thomas, P., Stan, G.-B., Barahona, M., 2021. Stationary distributions of continuous-time Markov chains: a review of theory and truncation-based approximations. SIAM Rev. 63 (1), 3–64.

Machinek, R.R., Ouldridge, T.E., Haley, N.E., Bath, J., Turberfield, A.J., 2014. Programmable energy landscapes for kinetic control of DNA strand displacement. Nature Commun. 5.

Madras, N.N., 2002. Lectures on Monte Carlo Methods. Vol. 16, American Mathematical Soc..

McGibbon, R.T., Pande, V.S., 2015. Efficient maximum likelihood parameterization of continuous-time Markov processes. J. Chem. Phys. 143 (3), 034109.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092.

Metzner, P., Schütte, C., Vanden-Eijnden, E., 2009. Transition path theory for Markov jump processes. Multiscale Model. Simul. 7 (3), 1192–1219.

Morrison, L.E., Stols, L.M., 1993. Sensitive fluorescence-based thermodynamic and kinetic measurements of DNA hybridization in solution. Biochemistry 32 (12), 3095–3104.

Munsky, B., Khammash, M., 2006. The finite state projection algorithm for the solution of the chemical master equation. J. Chem. Phys. 124 (4), 044104.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7 (4), 308–313.

Ouldridge, T.E., Louis, A.A., Doye, J.P., 2011. Structural, mechanical, and thermodynamic properties of a coarse-grained DNA model. J. Chem. Phys. 134 (8), 085101.

Parks, M.L., De Sturler, E., Mackey, G., Johnson, D.D., Maiti, S., 2006. Recycling Krylov subspaces for sequences of linear systems. SIAM J. Sci. Comput. 28 (5), 1651–1674.

Rubino, G., Tuffin, B., 2009. Rare Event Simulation using Monte Carlo Methods. John Wiley & Sons.

Sandmann, W., 2008. Discrete-time stochastic modeling and simulation of biochemical networks. Comput. Biol. Chem. 32 (4), 292–297.

Schaeffer, J.M., 2013. Stochastic simulation of the kinetics of multiple interacting nucleic acid strands (Ph.D. thesis). California Institute of Technology.

Schaeffer, J.M., Thachuk, C., Winfree, E., 2015. Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In: DNA Computing and Molecular Programming. In: Lecture Notes in Computer Science, Vol. 9211, pp. 194–211.

Sidje, R.B., Vo, H.D., 2015. Solving the chemical master equation by a fast adaptive finite state projection based on the stochastic simulation algorithm. Math. Biosci. 269, 10–16.

Simmons, G.F., 1972. Differential Equations with Applications and Historical Notes. CRC Press.

Singhal, N., Snow, C.D., Pande, V.S., 2004. Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. J. Chem. Phys. 121 (1), 415–425.

Soloveichik, D., Seelig, G., Winfree, E., 2010. DNA as a universal substrate for chemical kinetics. Proc. Nat. Acad. Sci. USA 107 (12), 5393–5398.

Srinivas, N., Parkin, J., Seelig, G., Winfree, E., Soloveichik, D., 2017. Enzyme-free nucleic acid dynamical systems. 138420, bioRxiv.

Suhov, Y., Kelbert, M., 2008. Probability and Statistics By Example: Volume 2, Markov Chains: A Primer in Random Processes and their Applications. Vol. 2, Cambridge University Press.

Šulc, P., Romano, F., Ouldridge, T.E., Rovigatti, L., Doye, J.P., Louis, A.A., 2012. Sequence-dependent thermodynamics of a coarse-grained DNA model. J. Chem. Phys. 137 (13), 135101.

Sun, T.-t., Zhao, C., Chen, S.-J., 2018. Predicting cotranscriptional folding kinetics for riboswitch. J. Phys. Chem. B 122 (30), 7484–7496.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tang, X., 2010. Techniques for modeling and analyzing RNA and protein folding energy landscapes (Ph.D. thesis). Texas A & M University.

Tang, X., Kirkpatrick, B., Thomas, S., Song, G., Amato, N.M., 2005. Using motion planning to study RNA folding kinetics. J. Comput. Biol. 12 (6), 862–881.

Turner, T.E., Schnell, S., Burrage, K., 2004. Stochastic approaches for modelling in vivo reactions. Comput. Biol. Chem. 28 (3), 165–178.

Van Kampen, N.G., 1992. Stochastic Processes in Physics and Chemistry. Vol. 1, Elsevier.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al., 2020. Scipy 1.0: fundamental algorithms for scientific computing in python. Nature Methods 1–12.

Weinan, E., Ren, W., Vanden-Eijnden, E., 2002. String method for the study of rare events. Phys. Rev. B 66, 052301.

Wetmur, J.G., Davidson, N., 1968. Kinetics of renaturation of DNA. J. Mol. Biol. 31 (3), 349–370.

Whitt, W., 2006. Continuous-Time Markov Chains. Dept. of Industrial Engineering and Operations Research, Columbia University, New York.

Zhang, J.X., Fang, J.Z., Duan, W., Wu, L.R., Zhang, A.W., Dalchau, N., Yordanov, B., Petersen, R., Phillips, A., Zhang, D.Y., 2018. Predicting DNA hybridization kinetics from sequence. Nature Chem. 10 (1), 91.

Zolaktaf, S., Dannenberg, F., Rudelis, X., Condon, A., Schaeffer, J.M., Schmidt, M., Thachuk, C., Winfree, E., 2017. Inferring parameters for an elementary step model of DNA structure kinetics with locally context-dependent arrhenius rates. In: DNA Computing and Molecular Programming. In: Lecture Notes in Computer Science, Vol. 10467, pp. 172–187.

Zolaktaf, S., Dannenberg, F., Winfree, E., Bouchard-Côté, A., Schmidt, M., Condon, A., 2019. Efficient parameter estimation for DNA kinetics modeled as continuous-time Markov chains. In: DNA Computing and Molecular Programming. In: Lecture Notes in Computer Science, Vol. 11648, pp. 80–99.

Zuckerman, D.M., Chong, L.T., 2017. Weighted ensemble simulation: review of methodology, applications, and software. Annu. Rev. Biophys. 46, 43–57.