Multiple state transition interface sampling of alanine dipeptide in explicit solvent

Du, W.; Marino, K.A.; Bolhuis, P.G.

Published in:
Journal of Chemical Physics

DOI:
10.1063/1.3644344

Multiple state transition interface sampling of alanine dipeptide in explicit solvent

Wei-Na Du, Kristen A. Marino, and Peter G. Bolhuis

Citation: J. Chem. Phys. **135**, 145102 (2011); doi: 10.1063/1.3644344
View online: http://dx.doi.org/10.1063/1.3644344
View Table of Contents: http://jcp.aip.org/resource/1/JCPSA6/v135/i14
Published by the American Institute of Physics.

Multiple state transition interface sampling of alanine dipeptide in explicit solvent

Wei-Na Du, Kristen A. Marino, and Peter G. Bolhuis[a)]
Van't Hoff Institute for Molecular Sciences, University of Amsterdam, PO Box 94157, 1090 GD Amsterdam, The Netherlands

(Received 13 July 2011; accepted 9 September 2011; published online 10 October 2011)

We have applied the recently developed multiple state transition interface sampling approach to alanine dipeptide in explicit water. We extract the rate constant matrix for configurational changes between each pair of metastable states. The results are comparable with values from previous literature and show that the method is applicable to biomolecular systems. © *2011 American Institute of Physics*. [doi:10.1063/1.3644344]

I. INTRODUCTION

Investigating the long time scale dynamics of rare events in complex biomolecular systems provides a challenge for molecular dynamics simulations. Typically, a biomolecular system spends most of the time in (meta)stable states and only occasionally jumps between these states. When using straightforward molecular dynamics simulation to investigate such rare events, a large amount of computer time will be wasted on sampling the metastable states. In the case of protein folding and unfolding, this waiting time can easily becomes microseconds to seconds, a time beyond even the capability of the most powerful modern super computer. Therefore, the investigation of the dynamics of such rare events necessitates the application of novel simulation algorithms. In the past decades, many algorithms have been developed to assess rare protein conformational transitions, for instance by umbrella sampling,[1] flooding,[2] local elevation,[3] adaptive bias force,[4] metadynamics,[5] replica exchange,[6] milestoning,[7] string method,[8] and many others. One such approach, transition path sampling (TPS) (Refs. 9–12) resolves the rare event problem by focusing only on the transition paths between stable states. TPS does this by generating an ensemble of unbiased dynamical trajectories connecting an initial with a final stable state. The advantage of TPS is the relative independence from an *a priori* choice of order parameters or reaction coordinate that describing the transition. Rather, the reaction coordinate can be extracted from the simulation results. The method has been successfully applied to various two-state systems[11,13,14] previously (see, e.g., Ref. 15 for a review). One drawback of the regular two state version of TPS is that, once there are more than two (meta)stable or intermediate states, trajectories will be attracted to those additional states and become trapped, which severely decreases the efficiency of the method. A remedy for this drawback is the recently developed multiple state transition path sampling (MSTPS),[16] which extends TPS by including all possible transition paths between any two stable or intermediate states in the path ensemble. In Ref. 16, it is shown that the MSTPS method is more efficient than the two state TPS method.

While TPS samples dynamical paths, information about rate constants is only available through a reversible work calculation of slowly constricting pathways leaving the initial stable to end at the final state.[9] Introduction of the transition interface sampling (TIS) method accelerated this computation considerably by measuring the effective positive flux through a series of interfaces between the initial and final states.[17] Reference 16 also introduced a multiple state version of TIS (MSTIS) and applied it to a simple 2D system.

The aim of this work is to show that the MSTPS/MSTIS framework can be applied to biomolecular systems and allows the evaluation of the full rate matrix. To that end, we apply the MSTIS method to a small prototypical biomolecule, the alanine dipeptide. Alanine dipeptide in aqueous solution has been widely used to test new simulation and sampling algorithms.[18–23] Despite its simplicity, it exhibits several multiple metastable states when solvated in explicit water, with transition rates between these states on the order of nanoseconds.

Chodera *et al.*[18] computed the transition matrix of alanine dipeptide conformational changes directly using straightforward molecular dynamics. The authors found that the alanine peptide configuration space could best be divided into 6 metastable states. This division in states leads to a Markov state model (MSM) in which the molecular kinetics can be summarized as stochastic transitions between the states.[24] Using the same division into six states, we compute the rate matrix using MSTIS, and compare it with these previous results. As some transition are much more abundant than others, one can combine states, to reduce the complexity of the rate matrix.

The paper is organized as follows. We first describe the theoretical background in Sec. II. In Sec. III, we present and discuss our MSTPS/MSTIS results. We end with concluding remarks.

a)Author to whom correspondence should be addressed. Electronic mail: p.g.bolhuis@uva.nl.

II. METHODS

A. Multiple state transition path sampling

1. Two-state transition path sampling

In this section, we briefly review the TPS methodology. A TPS simulation performs a Monte Carlo random walk in trajectory space. A dynamical trajectory $\mathbf{x}(L) = \{x_0, x_1, \ldots, x_L\}$ is discretized in time by a time-step Δt into $L+1$ slices x_τ. Each time slice $x_\tau = \{\mathbf{r}_\tau, \mathbf{p}_\tau\}$ contains all positions r and momenta p of all N particles in the system at time $t = \tau \Delta t$. The probability $\mathcal{P}_{AB}[\mathbf{x}(L)]$ for a path of a fixed length L connecting the initial state A and the final state B is given by

$$\mathcal{P}_{AB}[\mathbf{x}(L)] = Z_{AB}^{-1} h_A(x_0) \mathcal{P}[\mathbf{x}(L)] h_B(x_L). \tag{1}$$

$\mathcal{P}[\mathbf{x}(L)]$ is the dynamical probability of the discretized path $\mathbf{x}(L)$, while $h_\Omega(x)$ denotes an indicator function that is unity when $x \in \Omega$, i.e., x is inside a stable state Ω and zero otherwise. The product $h_A(x_0)h_B(x_L)$ thus guarantees that the probability is non-zero only when a path starts from A and ends in B. The normalization factor Z_{AB} is akin to a partition function. The dynamical path probability $\mathcal{P}[\mathbf{x}(L)]$ is given by

$$\mathcal{P}[\mathbf{x}(L)] = \rho(x_0) \prod_{\tau=0}^{L-1} p(x_\tau \to x_{(\tau+1)}), \tag{2}$$

where $p(x_\tau \to x_{(\tau+1)})$ represents the Markovian probability for transitions from a phase point at time τ to one at time $\tau + 1$ (e.g., a delta function for molecular dynamics, or a Gaussian for Langevin dynamics[9]). All trajectories that connect the two defined stable states form the *path ensemble*, which plays a key role in TPS as the representation of the system's rare event. A definition of reaction coordinate is not necessary in TPS, but the reactant A and product B should be defined properly by order parameters. These order parameters should be chosen such that the two states can not only be distinguished from each other, but also that each state lies inside the basin of attraction of that stable state.[15]

The shooting algorithm[9] can efficiently sample the path ensemble by selecting a random slice, changing the momenta slightly and shooting off a new direction forward and backward in time. Each path that still connects A with B can be accepted in the Monte Carlo procedure.

Based on TPS, the TIS algorithm for calculating rate constants between two states requires the definition of a series of hyper-surfaces or "interfaces" $\{x : \lambda(x) = \lambda_s\}$ with $\lambda_s \in \mathbb{R}$.[17] TIS thus requires an order parameter λ that parametrizes the rare event. The definition of this order parameter here is not as strict as a reaction coordinate, but it has to vary monotonically from reactant to product. In TIS, the rate constant k_{AB} for the $A \to B$ transition is then expressed as

$$k_{AB} = \langle \phi_1 \rangle P_A(\lambda_B|\lambda_1) = \langle \phi_1 \rangle \prod_{s=1}^{n-1} P_A(\lambda_{s+1}|\lambda_s). \tag{3}$$

Here, $\langle \phi_1 \rangle$ is the average flux out of state A through interface λ_1, which, in practice, can be obtained from an MD simulation by establishing the number of times the system crosses λ_1, coming directly from A, per unit time. $P_A(\lambda_B|\lambda_1)$ is the conditional "crossing" probability that a path from A has crossed interface λ_1 and reaches state B before returning to A. This crossing probability, which is naturally very small as we are considering a rare event, can be replaced with the product of n crossing probabilities in Eq. (3). One can compute $P_A(\lambda_{s+1}|\lambda_s)$ for a particular TIS interface λ_s as the fraction of paths coming out of state A and crossing λ_s, that also cross the next interface λ_{s+1} before going back to λ_s. Note that the final interface $\lambda_n = \lambda_B$.

2. Multiple state TPS/TIS

The standard TPS scheme assumes the presence of only two stable states. Many complex systems might have additional (meta)stable states that can trap the trial trajectories during the shooting move, thus making it difficult to sample paths between two specific stable states. The multiple state TPS/TIS method[16] resolves this problem by sampling the ensemble of pathways that connect any two of the stable states in the system instead of focusing on only two of the stables states. The path sampling algorithm simply accepts any path between two valid metastable states $i \neq j$. (For a detailed description see Ref. 16).

Reference 16 also introduces a scheme to compute the rate constants k_{ij} between any pair of stable states in the system. Each of the stable states i is assigned a set of m λ_{si}-interfaces, with $0 \leq s \leq m$. λ_{mi} is thus the "outermost" interface for stable state i. The multiple state TIS ensemble for paths between state i and any other state j is then

$$\mathcal{P}_{ij}^{\mathrm{TIS}}[\mathbf{x}(L)] \equiv Z_{\mathrm{TIS}}^{-1} \prod_k \bar{h}_k[\mathbf{x}(L)] h_i(x_0) \mathcal{P}[\mathbf{x}(L)] h_j(x_L) \hat{h}_i^m[\mathbf{x}(L)], \tag{4}$$

where Z_{TIS} is a normalization factor. The indicator functions $h_i(x_0)$ and $h_j(x_L)$ ensure again that the paths begin in state i and end in j, respectively. The product over $\bar{h}_k[\mathbf{x}(L)]$ is unity only when each configuration x along the path is outside each of the stable state definitions and zero otherwise, thus guaranteeing that except for the end points, the paths do not visit any of the stable states. This requirement is necessary, because we use the TIS flexible path length convention, i.e., the path length is determined by the first entering point of any stable state. The function $\hat{h}_i^m[\mathbf{x}(L)]$ is unity for paths that cross the outermost interface λ_{mi} and vanishes otherwise. Note that the initial and final state can also be the same, i.e. $i = j$ as long as λ_{mi} is crossed. Analogous to Eq. (3) the rate constant k_{ij} for transitions from a state i to a state j is

$$k_{ij} = \langle \phi_{mi} \rangle P_i(\lambda_{0j}|\lambda_{mi}). \tag{5}$$

Here, the flux through the outermost λ_{mi}-interface $\langle \phi_{mi} \rangle$ follows from a "regular" TIS simulation using the set of λ-interfaces for a given state i

$$\langle \phi_{mi} \rangle = \langle \phi_{1i} \rangle P_i(\lambda_{mi}|\lambda_{1i}) = \langle \phi_{1i} \rangle \prod_{s=1}^{m-1} P_i(\lambda_{(s+1)i}|\lambda_{si}). \tag{6}$$

$P_i(\lambda_{0j}|\lambda_{mi})$ is the crossing probability for a path that crosses λ_{mi} coming directly from i to reach state j before returning to i, and follows directly from the multiple state transition path sampling ensemble

$$P_i(\lambda_{0j}|\lambda_{mi}) = \frac{\int \mathcal{D}\mathbf{x}(L)\mathcal{P}_{ij}^{\mathrm{TIS}}[\mathbf{x}(L)]}{\int \mathcal{D}\mathbf{x}(L)\sum_j \mathcal{P}_{ij}^{\mathrm{TIS}}[\mathbf{x}(L)]} \approx \frac{n_{ij}}{\sum_j n_{ij}}, \qquad (7)$$

where the integrals are over all paths regardless of length. This probability can be estimated from the number of pathways n_{ij} starting in i, crossing λ_{mi} and ending in j, divided by all pathways starting in i and crossing λ_{mi}.

From the expression for the rate constants for transitions between all stable states i and j in Eq. (5) it follows that the flux $\langle\phi_{mi}\rangle$ in Eq. (6) only has to be calculated once for each stable state i. All remaining crossing probabilities can be obtained simultaneously within one multiple state TIS simulation.

The rate constant matrix hence consists of 3 factors,

$$k_{ij} = \langle\phi_{1i}\rangle P_i(\lambda_{mi}|\lambda_{1i})P_i(\lambda_{0j}|\lambda_{mi}), \qquad (8)$$

that are determined, respectively, by a straightforward MD simulation, a TIS for each state, and a multiple state TIS simulation.

The shooting move for MSTIS is done in the regular way, by accepting any path that connects two stable states i and j, provided it crosses the interface λ_{si}.

3. Combining stable states

A potential problem with a multiple state approach is that the free energy barriers between pairs i and j might be different in height. This will favor paths between the pairs i and j that have the lowest free energy barrier. When the barriers are roughly of the same height, this is not a problem as then all possible paths are equally represented in the ensemble. However, for widely varying barriers, one transition might dominate the ensemble. This dominance might be avoided by applying the Wang-Landau scheme advocated in Ref. 16 and explored in Ref. 25 (Recently this Wang-Landau scheme was also applied in the TPS framework[26]). Another possibility is to exclude the most dominant transition, i.e., those with very low barriers. Ideally, this should be done automatically based on the frequency of sampling. Alternatively, we can view the sets of states separated by low barriers as a single combined state, as interconversion between these states will be fast compared to the higher barriers. For instance, state α' and α'' can be combined into one state α with an indicator function $h_\alpha[x]$ which is unity if $x \in \alpha = \alpha' \cup \alpha''$ and zero otherwise. Special care needs to be taken for defining the TIS interfaces for such combined states. One possible combined interface definition is shown in Fig. 1. The order parameters λ_s', λ_s'' denote the distance to the center of respectively α' and α'' in some metric. These distances define interfaces $\{x : \lambda'(x) = \lambda_s'\}$ and $\{x : \lambda''(x) = \lambda_s''\}$ for both states. We can combine these interfaces through the envelope of the hypersurface, i.e., $\{x : \lambda(x) = \min[\lambda'(x), \lambda''(x)] = \lambda_s\}$, where the min function returns the smaller of its argument. The combined interface is thus defined by the distance λ_s to the center of the nearest sub-state in the combined state.

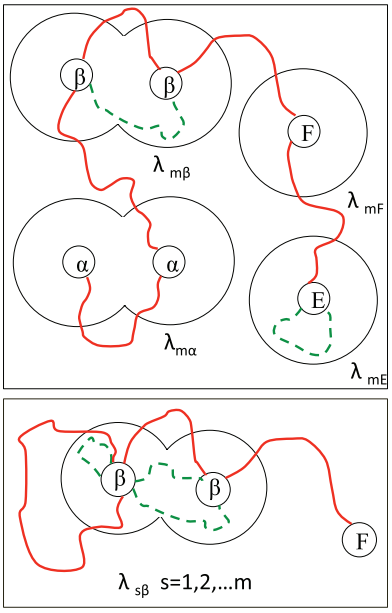

FIG. 1. Top: Schematic picture of acceptable (solid lines) and unacceptable (dashed lines) trajectories in the MSTIS path ensemble, using the combined states α and β. Bottom: Schematic picture for regular TIS with combined states. Again, acceptable paths (solid) cross the combined interface $\lambda_{m\beta}$ (black solid dumbbell), while unacceptable paths (dashed lines) do not.

The shooting move using the combined state interface and state definitions is straightforward. Each path has to start in $\alpha = \alpha' \cup \alpha''$, end in any state, and cross the combined interface, i.e., at least one time slice x should obey $\min[\lambda'(x), \lambda''(x)] > \lambda_s$.

Note the combination rule can be easily generalized to a set of l sub-states and corresponding distance order parameters $\lambda(x) = \{\lambda_1(x), \lambda_2(x), ..., \lambda_l(x)\}$ leading to the interface definition $\{x : \min[\lambda(x)] = \lambda_s\}$.

B. Simulation details

1. System setup and equilibration

All energy minimization and molecular dynamics simulations were performed with the GROMACS package (version 4.0.5) (Refs. 27–30) using the AMBER96 force field.[31] A

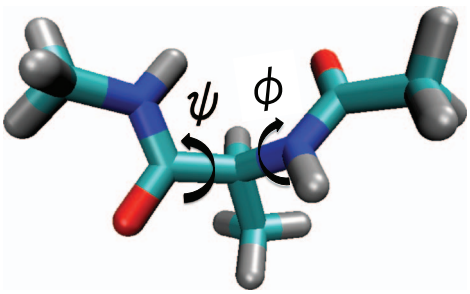

FIG. 2. The molecular structure of alanine dipeptide rendered in licorice representation. Carbons in cyan, oxygen in red, nitrogen in blue, and hydrogen white. The two order parameters describing the metastable states are the dihedral angles ϕ and ψ. Figure made with VMD (Ref. 38).

single alanine dipeptide molecule (Ace-Ale-NME, see Fig. 2) was solvated by 620 TIP3P water molecules[32] in a truncated octahedral periodic box. The non-bonded van der Waals cutoff radius was 1.1 nm. The LINCS algorithm handled bond constraints,[33] and fast Particle-Mesh Ewald method treated long-distance electrostatic interactions.[34,35] The system was energy minimized using steepest descent energy minimization followed by conjugate gradient optimization. A 10 ps position restricted MD simulation was performed, followed by a 1 ns equilibration run at constant temperature and constant pressure of respectively, 300 K and 1 atm. The time step was 2 fs. The temperature was kept constant with the Nose-Hoover thermostat,[36,37] and the pressure was maintained by a Parrinello-Rahman barostat. The average box size from this constant pressure simulation provided the proper box size (d = 2.92160 Å) to run the constant volume simulations.

2. Replica exchange molecular dynamics

To initialize MSTPS, we need definitions of the various metastable states of the alanine dipeptide. One way to obtain these beforehand, is by performing a very long MD simulation, or by accelerating methods such as replica exchange MD. Replica exchange molecular dynamics (REMD) (Ref. 39) can enhance the sampling of biomolecules with rough free energy surfaces by simulating n replicas at different temperatures and occasionally exchanging the replicas. In each replica, the velocity rescaling thermostat with a stochastic term is employed for temperature coupling. Every 1 ps, $n(n-1)$ exchanges between any random pair of replicas were attempted. The acceptance rule for exchanging each pair of selected replicas (1 and 2) is given by

$$P_{acc}[1 \to 2] = \min\left[1, e^{(E_1 - E_2)\left(\frac{1}{k_B T_1} - \frac{1}{k_B T_2}\right)}\right],$$

where k_B represents Boltzmann constant, T_1 and T_2 are temperatures of the two selected replicas, while E_1 and E_2 are their potential energies. The configurations of alanine dipeptide and system potential energy were written to disk every 0.1 ps.

To explore the free energy surface we ran a 20 ns REMD simulation with 24 replicas. The temperatures in these 24 replicas were chosen from a exponential distribution from 300 K to 500 K. The exchange move had an average acceptance ratio of 46%, based on first neighbor exchanges only (the total number of exchange trials is of course much larger). The free energy (in units of $k_B T$) is the negative logarithm of the probability histogram obtained by projecting the REMD trajectory data to suitable order parameters. We use the virtual move Monte Carlo method for better statistics.[40]

3. Path sampling simulation

The path sampling is performed by a perl-script wrapper around the GROMACS package. We employed the *two-way* shooting algorithm with flexible path length. Shooting points for generating trial paths are picked randomly from the previous accepted path with length $L^{(o)}$ and a new velocity will be assigned randomly for each atom at that point. The trial trajectories are generated with the Gromacs MD engine by integrating both forward and backward in time. The integration is stopped when a path reaches one of the stable states. When crossing the current interface, the trial path with a length $L^{(n)}$ can be accepted with the Metropolis rule

$$P_{acc}[o \to n] = \min\left[1, \frac{L^{(o)}}{L^{(n)}}\right] \tag{9}$$

in order to obey detailed balance.[17]

III. RESULTS AND DISCUSSION

A. Stable states and interfaces from REMD simulation

The free energy was obtained by projecting the REMD trajectory data onto two slow degrees of freedom, the dihedral angles ϕ and ψ, which have been widely used to investigate the free energy surface of the alanine dipeptide system.[19–23] The resulting free energy landscape is shown in Fig. 3 and shows six distinct energy minima. The landscape agrees well with a previous investigation employing the same force field and water type.[18] In Ref. 18, six metastable states were identified by defining boxes in $\phi - \psi$ phase space, where each box contains one of the six minima. These six distinct energy minima will be used to define the metastable states in this work but now are based on the "core" definition. This definition is based on the distance in ϕ, ψ space to the free energy minimum. The order parameter for the stable states is thus $\lambda_i = \sqrt{(\phi - \phi_i^{\text{ref}})^2 + (\psi - \psi_i^{\text{ref}})^2}$, where $(\phi_i^{\text{ref}}, \psi_i^{\text{ref}})$ denotes the location of the minima in the free energy landscape. Based on the REMD free energy landscape, we locate the six minima A, B, C, D, E, and F at, respectively, $(-150, 150)$, $(-70, 135)$, $(-150, -65)$, $(-70, -50)$, $(50, -100)$, and $(40, 65)$ (in degrees), see Fig. 3. The stables state i is defined by the set $\{x : \lambda_i < \lambda_{0i}\}$. For simplicity, we set all stable state boundaries identical ($\lambda_{0A} = \lambda_{0B} = \lambda_{0C} = \lambda_{0D} = \lambda_{0E} = \lambda_{0F} = 10$), i.e., a circle with a 10-degree radius. At first sight, this might seem unfit for a proper stable state definition. However, the most important criterion for a stable state definition is that trajectories should have a high probability of hitting the stable state definition when in the basin of attraction, not that they always have to stay within this definition.[12,15] For each state, we

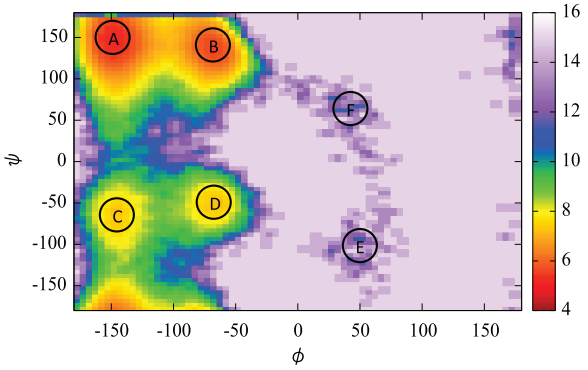

FIG. 3. The free energy surface from the REMD simulation obtained by projecting the logarithm of the probability to find a (ϕ, ψ) pair.

TABLE I. Definition of interfaces (in degrees) for the six states A-F. Also included are the combined β and α states for the 4-state computation.

	A	B	C	D	E	F	β	α
λ_0	10	10	10	10	10	10	10	10
λ_1	20	20	20	20	20	20	20	20
λ_2	45	45	45	45	45	45	45	40
λ_3	65	65	60	60	65	65	55	45
λ_4	80	75			80	80	65	50
λ_5							75	60

can use this order parameter to define a series of interfaces, from the zeroth interface (the stable state boundaries λ_{0i}) up to the outermost one (λ_{mi}). These interfaces are summarized in Table I. While the order parameter for each state is here defined in the same way, we stress that, in principle, one is free to a use different order parameter set for each state and its interfaces, provided that they separate and distinguish the stable states.

Note that we assume, similar to previous studies, that the minima are sufficiently deep to cause a separation of time scales, and hence a meaningful rate constant for each transition. However, the barriers between states A and B as well as between states C and D are so low that crossings are very frequent, and the assumption of exponential kinetics might be invalidated. We therefore also consider a 4-state description with the combined states $\alpha = C \cup D$ and $\beta = A \cup B$. The interfaces for the combined states are constructed as described in Sec. II, and are located at positions listed in Table I.

B. Flux factor from MD

Ten blocks of 10 ns MD simulations are carried out to compute the effective positive flux for each of the six stable states at its first interface λ_{1i}, a 20-degree-radius circle centering on the state's central point($\lambda_{1A} = \lambda_{1B} = \lambda_{1C} = \lambda_{1D} = \lambda_{1E} = \lambda_{1F}$=20). The calculated fluxes out of the first interfaces of stable states are given in the second column of Table II. In this effective positive flux computation, we only count the first crossing of λ_{1i} after the trajectory has escaped from the stable state i. The trajectory then first has to return to the stable state i before a new crossing can be counted.[17] The effective positive flux through the first interface is thus $\phi_{1i} = N_{1i}/t_i$ where N_{1i} is the number of times that the system crosses λ_{1i} when directly coming from state i, and t_i is the total residence time of state i, i.e., the total time the system stays in the basin of attraction of state i during the simulation. Note that this definition is slightly different from that of Ref. 17, because for the alanine dipeptide system a MD simulation can relatively easily leave a state spontaneously, e.g., from state A to B. Fluxes for state A, B, C, and D are extracted from straightforward MD, while fluxes for state E and F are from paths biased to those two states as they are so rarely visited in regular MD.

From the straightforward MD, we also extracted the fluxes for the 4-state MSM at their respective first interfaces. For the combined states, for example β, the flux at the first interface is simply the number of transitions out of $\beta = A \cup B$ divided by the total time t_β the system stays in β. Because the average residence times are additive $\langle t_\beta \rangle = \langle t_A \rangle + \langle t_B \rangle$, the flux is

$$\langle \phi_{1\beta} \rangle = N_{1\beta}/t_\beta = (N_{1A} + N_{1B})/(t_A + t_B), \tag{10}$$

provided the combined interfaces do not overlap.

TABLE II. Flux at the first interface (second column), the crossing probability from the first to the outermost interface (third column) and the flux at the outermost interface (last column). The subscript value is the error in the last two reported digits.

State	ϕ_{1i} (ps^{-1})	$P_i(\lambda_{mi}\|\lambda_{1i})$	ϕ_{mi} (ps^{-1})
A	2.031_{33}	0.0781_{82}	0.159_{17}
B	2.422_{21}	0.075_{15}	0.182_{37}
C	1.483_{59}	0.229_{23}	0.339_{38}
D	2.290_{81}	0.090_{15}	0.205_{35}
E	1.379_{54}	0.0719_{56}	0.0991_{80}
F	2.25_{11}	0.0266_{41}	0.0597_{87}
β	2.214_{21}	0.0259_{36}	0.0573_{79}
α	1.968_{58}	0.0725_{63}	0.142_{11}

C. Rate constants from MD

From the straightforward MD simulations, rate constants can be extracted directly by keeping track of the transitions between any pair of states. The rate constant $i \to j$ is then approximated by dividing the number of transitions between i and j by the total time the system spent in state i. This latter quantity is the time between first entering a stable state i coming from any other state, and entering any other state coming from i. Rate constants for transitions involving E and F are not available in this way, as these states are hardly visited.

In the same way, we also extracted the rate constants for the transitions involving the combined states α or β from the MD results.

D. Crossing probabilities $P_i(\lambda_{mi}|\lambda_{1i})$ by TIS

We performed ten blocks of TIS simulations consisting of around 12000 shooting moves each, for all the stable state interfaces listed in Table I. The average acceptance ratio is 47%.

For each TIS simulation, we collected the crossing probability histograms $P_i(\lambda|\lambda_{si})$ for interface λ_s by averaging over the path ensemble:

$$P_i(\lambda|\lambda_{si}) = \langle \theta(\lambda_i^{max}(\mathbf{x}(L)) - \lambda_{si}) \rangle_{si}, \tag{11}$$

where $\theta(x)$ is a step function, the $\langle \ldots \rangle_{si}$ denotes the average over the TIS path ensemble starting in state i and crossing interface λ_{si}, and the λ_i^{max} function returns the maximum value of the order parameter belonging to state i along the path $\mathbf{x}(L)$. In Fig. 4, we plot these histograms for state A, for each TIS simulation block. The histograms of the other states are shown in Appendix A. Note that the crossing probability histograms for each interface start at unity, as all paths in the ensemble have to cross that interface. Joining these histograms using weighted histogram analysis method (WHAM)

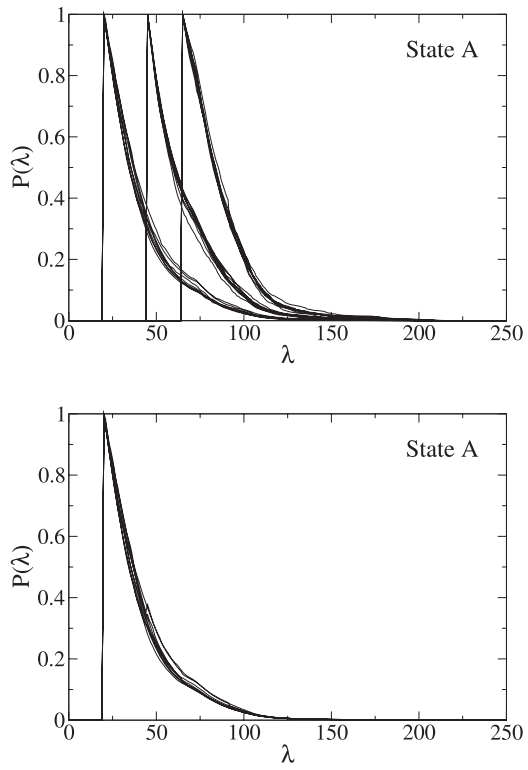

FIG. 4. Top: Crossing probability histograms $P(\lambda) = P_A(\lambda|\lambda_{sA})$ for state A. Note that for each TIS simulation the probability starts at unity. Bottom: The joint histograms $P(\lambda) = P_A(\lambda|\lambda_{1A})$ for each TIS block obtained using WHAM.

(Ref. 41) results in master histograms for each state. The joint histogram for state A is shown in Fig. 4, whereas for the other states, the histograms are shown in the right column Fig. 5 in Appendix A. These crossing probabilities all start at unity at the first interface, and quickly monotonically decay for higher values of λ. From these histograms follows the second factor in Eq. (8), the outer interface crossing probability $P_i(\lambda_{mi}|\lambda_{1i})$. The crossing probabilities from the first interface to the outermost interface are listed for all states in the third column of Table II.

E. The crossing probability $P_i(\lambda_{0j}|\lambda_{mi})$ by MSTIS

The last term in Eq. (8) $P_i(\lambda_{0j}|\lambda_{mi})$ is the crossing probability from the outermost interface λ_{mi} of state i to any other stable state j in the system. To establish these terms, we carry out multiple state TIS. To initialize the MSTIS simulation, we took a valid path crossing the λ_{mi} interface. The MSTIS simulation consisted of 10 blocks of around 18000 shooting moves each. Paths that connect any pair of *different* stable state pair $i \neq j$ are accepted. Paths that leave i and return to i are accepted as long as they cross the outermost interface of the state. As usual, the flexible path length algorithm requires the path not become longer than a maximum length, to obey detailed balance. The average acceptance ratio is 27%.

From the MSTIS path ensemble, we can extract the crossing probability matrix $P_i(\lambda_{0j}|\lambda_{mi})$ using Eq. (7). The path counts are summarized in Table III, where each line in the matrix lists the number of transition paths from one stable state to others. As the transition to E and F are rare due to the high barriers involved, we performed two MSTIS runs that biased the sampling towards the E and F states. The first run excludes the A→A, A→B, B→A, B→B and the C→C, C→D, D→C, D→D paths, while the second one only includes paths involving E or F. The path counts are summarized in Table III. The three matrices can be combined using WHAM. The

TABLE III. Left column: averaged MSTIS path counts over 10 blocks for the 6-state simulation rounded to the nearest integer. Right column: MSTIS path counts for the 4-state simulations. Top row: unbiased MSTIS. Middle and bottom row: biased MSTIS by excluding paths. The rows denote the leaving state, the columns the arriving state.

	A	B	C	D	E	F
A	2335	6036	161	69	1	1
B	5917	1816	38	36	0	16
C	172	46	665	599	0	0
D	58	38	616	626	4	0
E	1	0	1	3	141	19
F	1	15	0	0	17	16

	β	α	E	F
β	12448	729	0	28
α	764	3207	3	0
E	1	6	26	2
F	25	0	2	1

	A	B	C	D	E	F
A	0	0	4516	1621	8	19
B	0	0	1094	951	2	286
C	4586	1043	0	0	5	0
D	1702	948	0	0	277	0
E	14	2	6	223	705	91
F	10	318	1	2	80	93

	β	α	E	F
β	0	8182	10	306
α	8280	0	282	0
E	16	230	705	91
F	327	2	80	93

	A	B	C	D	E	F
A	0	0	0	0	69	90
B	0	0	0	0	27	2870
C	0	0	0	0	37	2
D	0	0	0	0	1410	5
E	73	23	30	1441	6079	651
F	109	2809	3	5	642	880

	β	α	E	F
β	0	0	96	2960
α	0	0	1447	7
E	97	1470	6079	651
F	2918	9	642	880

TABLE IV. MSTIS Crossing Probability matrices obtained by combining the path counts using WHAM and normalizing. Top: 6-state. Bottom: 4 state. The subscript value indicates the error in the last two digits. Rows denote leaving, columns arriving states.

	A	B	C	D	E	F
A	0.269_{25}	0.698_{27}	0.0236_{48}	0.0086_{19}	0.000050_{12}	0.000072_{33}
B	0.754_{14}	0.232_{14}	0.0063_{15}	0.0056_{16}	0.0000213_{78}	0.00221_{66}
C	0.144_{43}	0.033_{11}	0.434_{45}	0.388_{50}	0.000167_{78}	0.0000097_{70}
D	0.062_{24}	0.034_{14}	0.443_{65}	0.454_{48}	0.0072_{37}	0.000023_{14}
E	0.0091_{29}	0.0028_{11}	0.0039_{11}	0.171_{44}	0.733_{43}	0.080_{11}
F	0.023_{13}	0.610_{85}	0.0013_{15}	0.0018_{24}	0.158_{46}	0.207_{47}
	β	α	E	F		
β	0.9444_{90}	0.0532_{92}	0.000076_{13}	0.00233_{70}		
α	0.181_{25}	0.815_{25}	0.0042_{18}	0.000021_{12}		
E	0.0120_{37}	0.178_{50}	0.731_{48}	0.079_{12}		
F	0.635_{86}	0.0032_{36}	0.156_{45}	0.205_{48}		

resulting crossing probability matrices are given in Table IV, where each row in the matrix lists the probabilities from the outermost interface of a certain stable state to all other stable states in the system.

The 4-state crossing probability matrix required an additional MSTIS simulation as the state definitions are different. However, no additional simulation was needed to obtain the two biased matrices. We simply can extract a 4×4 matrix for states β, α, E and F from the 6-situation calculations by adding the path counts contributed by the different substates, as there are no transitions between A and B or C and D in those runs anyway. The 4×4 transition matrices are given in Table III, where each row in the matrices lists number of transitions starting from a certain stable state. Using WHAM on the three matrices, a transition probability matrix for the 4-state calculation is obtained, given in Table IV.

F. Rate constants

The rate constant matrix $\mathbf{k}_{ij}$ was calculated from Eq. (8), where ϕ_{1i} was taken from the straightforward MD runs, while $P_i(\lambda_{mi}|\lambda_{1i})$ from TIS calculations and $P_i(\lambda_{0j}|\lambda_{mi})$ from MSTIS calculations. As we have ten blocks of data for each of the three factors in Eq. (8), the rate constant can be evaluated in two different ways. The first is to compute the rate constants for each data set followed by averaging those 10 rate matrices. The resulting rate constants are summarized in Table V. The rate matrices of the 4-state division can be obtained via two routes: either directly using Eq. (8), or by a reduction of the 6-state rate matrix using the procedure outlined in Appendix B. These two matrices are also given in Table V.

Alternatively, we can first average each of the three factors separately, and then compute the rate constant $\mathbf{k}_{ij}$ by taking the product. The resulting matrices are given in Appendix C. The differences between the taking the average of the product and the product of the averages is within the reported error bar, indicating that the data sets are uncorrelated.

While the rates for the different computations are within each others error bar, there seems to be a significant difference between the rates in the middle and bottom matrices of Table V for transitions leaving state α or β. This difference might be due to the fact that that the barrier between A and B is probably not sufficiently high to have exponential kinetics and avoid correlations.

Note that while the MSTIS simulations successfully improved the sampling of (E,F)→(E,F) transitions, (E,F)→(β,α) transitions are still relatively rare, which sometimes makes the standard deviation of the rate constant significant (or even of the same order) compared to the rate constant itself.

Still, MSTIS is more efficient in calculating the rate constant at a fixed accuracy compared to straightforward MD.

TABLE V. Rate constant matrix from the average of products in Eq. (8). Top: 6-state computation. Middle: reduction of 6-states to 4-states using Eqs. (B2) and (B3). Bottom: direct 4-state calculation from Eq. (8). Rows denote leaving, columns arriving states. All rates in ps^{-1}.

	A	B	C	D	E	F
A		0.111_{14}	0.00377_{97}	0.00136_{29}	0.0000078_{18}	0.0000115_{53}
B	0.137_{27}		0.00114_{33}	0.0010240	0.0000039_{16}	0.00040_{14}
C	0.049_{14}	0.0111_{36}		0.132_{25}	0.000058_{30}	0.0000033_{25}
D	0.0124_{47}	0.0069_{25}	0.092_{25}		0.00145_{73}	0.0000048_{30}
E	0.00088_{24}	0.00027_{10}	0.00038_{11}	0.0170_{50}		0.0080_{16}
F	0.00136_{86}	0.0366_{93}	0.000074_{78}	0.00011_{15}	0.0093_{27}	
	β	α	E	F		
β		0.00376_{75}	0.00000600_{94}	0.000192_{64}		
α	0.035_{10}		0.00092_{45}	0.0000042_{19}		
E	0.00115_{29}	0.0174_{50}		0.0080_{16}		
F	0.0380_{94}	0.00019_{21}	0.0093_{27}			
	β	α	E	F		
β		0.00305_{73}	0.00000437_{99}	0.000133_{47}		
α	0.0256_{37}		0.00060_{27}	0.0000030_{19}		
E	0.00117_{31}	0.0177_{55}		0.0079_{16}		
F	0.0381_{93}	0.00019_{21}	0.0092_{27}			

TABLE VI. Comparison of our results with Ref. 18.

RESOURCE	$k_{\beta\to\alpha}$	$k_{\alpha\to\beta}$	$k_{\alpha\to E}$	$k_{E\to\alpha}$
Ref. 18	0.0046	0.0335	0.0001	0.0185
MD runs	0.00318_{50}	0.0265_{67}		
MSTIS 6-state	0.00376_{75}	0.035_{10}	0.00092_{45}	0.0174_{50}
MSTIS 4-state	0.00305_{73}	0.0256_{37}	0.00060_{27}	0.0177_{55}

For each set of our 10-set calculation, MSTIS required a total of 402 ns (this includes 10 ns MD, 260 ns TIS and 131 ns MSTIS). A conventional MD simulation aiming for the same accuracy of the rare C-F transition, e.g., would need to sample at least a few tens of microseconds. Moreover, we did not attempt to optimize the computation, for instance, by changing the location of the interfaces.[42,43]

For alanine dipeptide, MSTIS is also more efficient than two state TIS because the switching of transitions induces faster decorrelation between paths.[16,25] Moreover, MSTIS avoids the trapping of paths. Also, for two state TIS, the second and third terms of Eq. (8) need to be computed independently for each pair of states.

G. Comparison with previous work

As most studies within the existing literature on the rate constants for alanine dipeptide conformational change, e.g., Refs. 18–23, employ different force fields, set-up, state definitions or thermodynamic conditions, a direct comparison with our results is not very useful. The exception is the work of Chodera *et al.*,[18] which we took as a basis for this work. Because we use the same force field, conditions, and moreover (more or less) the same stable state definitions, our rate constant estimates and free energy differences should be identical to those of Ref. 18, as illustrated for several transitions in Table VI. However, the authors actually report not the rate but the *transition matrix* after a certain lag time. To compare, we can translate our rate matrix K into a transition matrix T by using the expression $T = \exp(K\tau)$, where τ is a lag time. For $\tau = 10$ ps, the resulting transition matrix together with results from Ref. 18, are given in Table VII. The two matrices agree quite well, except for the transitions to state F. This difference also shows up in the equilibrium population which is the first eigenvector of the transition matrix. Our rate matrix yields a population vector $p_{eq} = \{0.4953, 0.4009, 0.0388, 0.0558, 0.0048, 0.0043\}$, for state A-F, whereas the population vector from Ref. 18 is $p_{eq} = \{0.4852, 0.4090, 0.0403, 0.0625, 0.0013, 0.0017\}$. Again, these results are quite reasonable for the populated states, but do not agree for the states E and F.

TABLE VII. Comparison of the transition matrix based on our rate matrix(top) with that from Ref. 18 (bottom) for $\tau = 10$ ps.

	A	B	C	D	E	F
A	0.5677	0.4903	0.2194	0.1514	0.0221	0.1128
B	0.3972	0.4813	0.1336	0.0925	0.0190	0.1863
C	0.0170	0.0130	0.2813	0.2518	0.0318	0.0038
D	0.0169	0.0131	0.3613	0.4951	0.1011	0.0075
E	0.0002	0.0002	0.0039	0.0086	0.7704	0.0646
F	0.0010	0.0020	0.0004	0.0005	0.0555	0.6251
	A	B	C	D	E	F
A	0.5730	0.4590	0.2320	0.1570	0.0220	0.1380
B	0.3850	0.5200	0.1100	0.0720	0.0330	0.3330
C	0.0180	0.0130	0.2860	0.2350	0.0300	0.0050
D	0.0220	0.0080	0.3710	0.5350	0.1110	0.0090
E	0.0000	0.0000	0.0010	0.0010	0.7450	0.1270
F	0.0020	0.0000	0.0000	0.0000	0.0590	0.3880

TABLE VIII. Rate constant matrices as product of averages in Eq. (8). Top: 6-state computation. Middle: reduction of 6-states to 4-states using Eqs. (B2) and (B3). Bottom: direct 4-state calculation from Eq. (8). All rates in ps^{-1}.

	A	B	C	D	E	F
A		0.111_{13}	0.00374_{86}	0.00137_{34}	0.0000079_{21}	0.0000115_{53}
B	0.137_{28}		0.00115_{36}	0.00101_{36}	0.0000039_{16}	0.00040_{14}
C	0.049_{16}	0.0112_{40}		0.132_{22}	0.000057_{27}	0.0000033_{24}
D	0.0126_{54}	0.0070_{31}	0.091_{21}		0.00148_{79}	0.0000047_{29}
E	0.00090_{30}	0.00027_{11}	0.00038_{12}	0.0169_{46}		0.0080_{13}
F	0.00136_{81}	0.0365_{78}	0.000078_{90}	0.00011_{14}	0.0094_{31}	
	β	α	E	F		
β		0.00375_{98}	0.0000060_{19}	0.000192_{69}		
α	0.035_{13}		0.00094_{50}	0.0000042_{27}		
E	0.00117_{41}	0.0173_{47}		0.0080_{13}		
F	0.0378_{86}	0.00019_{23}	0.0094_{31}			
	β	α	E	F		
β		0.00305_{68}	0.00000437_{97}	0.000133_{44}		
α	0.0258_{43}		0.00060_{27}	0.0000029_{17}		
E	0.00119_{38}	0.0176_{52}		0.0079_{13}		
F	0.0380_{80}	0.00019_{22}	0.0093_{31}			

IV. CONCLUSION

In this paper, we computed the rate matrix for the solvated alanine dipeptide system by MSTIS simulations. To our knowledge, this is the first report of a complete kinetic rate constant matrix of the solvated alanine dipeptide.

A current limitation of MSTIS is that all states should be defined prior to starting the simulation. In systems more complex than alanine dipeptide with possibly many (meta)stable states this might be a challenge. A way to improve the algorithm such that previously undefined states can be defined "on the fly". Another possible improvement of the method might be the combination with replica exchange TIS.[44,45]

While the aim of this paper was to provide a reasonably realistic application of a biomolecular system, we realize that the solvated alanine dipeptide is extremely simple. Nevertheless, we hope that MSTIS can enhance the sampling of kinetics in larger biomolecular systems, and open up sampling of processes such as the folding of small polypeptides.

ACKNOWLEDGMENTS

This work was financially supported by the "Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)."

APPENDIX A: CROSSING PROBABILITY HISTOGRAMS FOR ALL STATE DEFINITIONS

Figure 5 shows the crossing probability histograms obtained from the TIS simulations as explained in Sec. III D, as well as the joint histograms resulting from the WHAM analysis.

APPENDIX B: REDUCTION OF THE RATE MATRIX

We can extract the 4-state (β, α, E, F) rate constant matrix from a 6-state (A, B, C, D, E, F) rate constant matrix as follows. First, we focus on the rates between the combined states β and α. Because $\alpha = C \cup D$, $\beta = A \cup B$ according to our definition, the calculation of $k_{\beta\to\alpha}$ can be divided into four parts: transitions $A \to C$ and $A \to D$ for paths that start in A, transitions $B \to C$ and $B \to D$ for paths that start in B. The individual rates for these transition follow from application of Eq. (8).

Combining two rates from the same initial state towards different final states, e.g., k_{AC} and k_{AD} can be done by simply adding them

$$k_{A\alpha} = \langle \phi_{1A} \rangle P_A(\lambda_{mA}|\lambda_{1A})(P_A(\lambda_{0C}|\lambda_{mA}) + P_A(\lambda_{0D}|\lambda_{mA})) = k_{AC} + k_{AD}. \quad \text{(B1)}$$

The combination of two rates from two different initial states toward the same final states, e.g., k_{AC} and k_{BC}, is slightly more involved. The combined rate is equal to the number of paths that leave the combined state and go on to reach the final state C in a total time $t_\beta = t_A + t_B$. The combined rate is

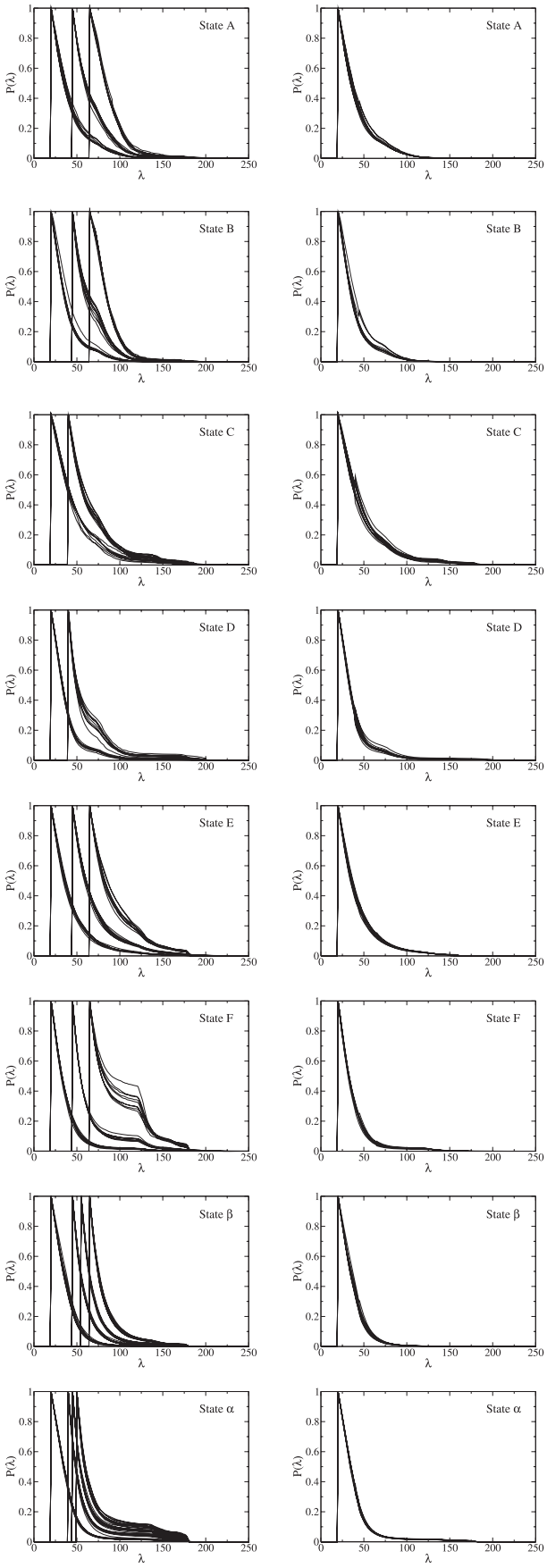

FIG. 5. Left column: Crossing probability histograms for all states, for all TIS blocks. Each line denotes a single TIS simulation. Right column: Combined histograms for each block obtained using WHAM.

thus obtained similar to the combined flux, Eq (10):

$$k_{\beta C} = \frac{N_A}{t_A + t_B} P_A(\lambda_{mA}|\lambda_{1A}) P_A(\lambda_{0C}|\lambda_{mA}) + \frac{N_B}{t_A + t_B} P_B(\lambda_{mB}|\lambda_{1B}) P_B(\lambda_{0C}|\lambda_{mB}) = \frac{t_A}{t_A + t_B} k_{AC} + \frac{t_B}{t_A + t_B} k_{BC}. \quad \text{(B2)}$$

The rate constant can be considered as a weighted sum of the two rates, with the weight being the relative residence times. States A and B are close enough to each other to be treated as one combined state, thus the ratio between the total time spent in those two states, t_A and t_B, could be easily obtained from short MD simulations. $k_{\beta\to\alpha}$ can now be obtained by combining (B1) and (B2)

$$k_{\beta\alpha} = \frac{t_A}{t_A + t_B}(k_{AC} + k_{AD}) + \frac{t_B}{t_A + t_B}(k_{BC} + k_{BD}). \quad \text{(B3)}$$

$k_{\alpha\beta}$ was generated in the same way. Rate constants for transitions between α, β and E or F can be calculated as in Eq (B2).

Using the algorithm described above, we can, based on the 6×6 matrices obtained from the 6-state simulation, compute two 4×4 matrices corresponding to the average of products and product of averages for states β, α, E, and F (see Tables V and VIII).

APPENDIX C: RATE CONSTANT MATRICES USING THE PRODUCT OF AVERAGES

In addition to applying Eq. (8) for the individual data sets, we can take the average of each of the three factors separately, and then compute the rate constant $\mathbf{k}_{ij}$ by taking the product. An average ϕ_{1i} can be obtained from ten blocks of MD data. Similarly, we get the average $P(\lambda_{mi}|\lambda_{1i})$ and $P(\lambda_{0j}|\lambda_{mi})$ for all $i \to j$ transitions. Subsequent application of Eq. (8), results in the rate matrices shown in Table VIII. The difference between these results and the ones in Table V are minimal, indicating that the different data sets are uncorrelated.

[1] G. M. Torrie and J. P. Valleau, Chem. Phys. Lett. **28**, 578 (197).
[2] H. Grubmüller, Phys. Rev. E **52**, 2893 (1995).
[3] T. Huber, A. Torda, and W. van Gunsteren, J. Comput.- Aided Mol. Des. **8**, 695 (1994).
[4] E. Darve and A. Pohorille, J. Chem. Phys. **115**, 9169 (2001).
[5] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. U.S.A. **99**, 12562 (2002).
[6] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).
[7] A. Faradjian and R. Elber, J. Chem. Phys **120**, 10880 (2004).
[8] W. E. W. Ren, and E. Vanden-Eijnden, Phys. Rev. B **66**, 052301 (2002).
[9] C. Dellago, P. G. Bolhuis, F. S. Csajka, and D. Chandler, J. Chem. Phys. **108**, 1964 (1998).
[10] P. G. Bolhuis, D. Chandler, C. Dellago, and P. L. Geissler, Ann. Rev. Phys. Chem. **53**, 291 (2002).
[11] C. Dellago and P. G. Bolhuis, in *Atomistic Approaches in Modern Biology: From Quantum Chemistry to Molecular Simulations*, Top. Curr. Chem., Vol. 268, edited by M. Reiher (Springer, Berlin, 2007), pp. 291–317.
[12] C. Dellago and P. G. Bolhuis, in *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, Adv. Polym. Sci., Vol. 221, edited by C. Holm and K. Kremer (Springer, Berlin, 2009), pp. 167–233.
[13] D. Moroni, P. R. ten Wolde, and P. G. Bolhuis, Phys. Rev. Lett. **94**, 235703 (2005).
[14] J. Juraszek and P. G. Bolhuis, J. Chem. Phys. **129**, 224107 (2008).
[15] P. G. Bolhuis and C. Dellago, in *Rev. Comp. Chem.*, edited by K. B. Lipkowitz (Wiley, New York, 2010), Vol. 27, pp. 111–202.
[16] J. Rogal and P. G. Bolhuis, J. Chem. Phys. **129**, 224107 (2008).
[17] T. S. van Erp, D. Moroni, and P. G. Bolhuis, J. Chem. Phys. **118**, 7762 (2003).
[18] J. D. Chodera, W. C. Swope, J. W. Pitera, and K. A. Dill, Multiscale Model. Simul. **5**, 1214 (2006).
[19] C. Velez-Vega, E. E. Borrero, and F. A. Escobedo, J. Chem. Phys. **130**, 1225101 (2009).
[20] D. S. Chekmarev, T. Ishida, and R. M. Levy, J. Phys. Chem. B **108**(50), 19487 (2004).
[21] P. G. Bolhuis, C. Dellago, and D. Chandler, Proc. Natl. Acad. Sci. U.S.A. **97**, 5877 (2000).
[22] A. M. A. West, R. Elber, and D. Shalloway, J. Chem. Phys. **126**, 145104 (2007).
[23] C. A. F. de Oliveira, D. Hamelberg, and J. A. McCammon, J. Chem. Phys. **127**, 175105 (2007).
[24] J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, and W. C. Swope, J. Chem. Phys. **126**, 155101 (2007).
[25] J. Rogal and P. G. Bolhuis, J. Chem. Phys. **133**, 034101 (2010).
[26] E. E. Borrero and C. Dellago, J. Chem. Phys. **133**, 134112 (2010).
[27] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, J. Chem. Theory Comput. **4**, 435 (2008).
[28] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, J. Comput. Chem. **26**, 1701 (2005).
[29] E. Lindahl, B. Hess, and D. van der Spoel, J. Mol. Model **7**, 306 (2001).
[30] H. J. C. Berendsen, D. van der Spoel, and R. van Drunen, Comput. Phys. Commun. **91**, 43 (1995).
[31] P. A. Kollman, R. Dixon, W. Cornell, T. Vox, C. Chipot, and A. Pohorille, in *Computer Simulation of Biomolecular Systems*, edited by W. van Gunsteren, P. Weiner, and A. Wilkinson (Springer, Berlin, 2008), Vol. 3, pp. 83–96.
[32] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**, 926 (1983).
[33] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije, J. Comp. Chem. **18**, 1463 (1997).
[34] T. Darden, D. York, and L. G. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
[35] U. Essman, L. Perela, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
[36] S. Nose, Mol. Phys. **52**, 255 (1984).
[37] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).
[38] W. Humphrey, A. Dalke, and K. Schulten, J. Mol. Graph. **14**, 33 (1996).
[39] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).
[40] D. Frenkel and B. Smit, *Understanding molecular simulation*, 2nd ed. (Academic, San Diego, CA, 2002).
[41] S. Kumar, D. Bouzida, R. H. Swendsen, P. A. Kollman, and J. M. Rosenberg, J. Comput. Chem. **13**, 1011 (1992).
[42] E. E. Borrero and F. A. Escobedo, J. Chem. Phys. **129**, 024115 (2008).
[43] E. E. Borrero, M. Weinwurm, and C. Dellago, J. Chem. Phys. **134**, 244118 (2011).
[44] T. S. van Erp, Phys. Rev. Lett. **98**, 268301 (2007).
[45] P. G. Bolhuis, J. Chem. Phys. **129**, 144108 (2008).