Chapter 2

Transition Path Sampling with Quantum/Classical Mechanics for Reaction Rates

Frauke Gräter and Wenjin Li

Abstract

Predicting rates of biochemical reactions through molecular simulations poses a particular challenge for two reasons. First, the process involves bond formation and/or cleavage and thus requires a quantum mechanical (QM) treatment of the reaction center, which can be combined with a more efficient molecular mechanical (MM) description for the remainder of the system, resulting in a QM/MM approach. Second, reaction time scales are typically many orders of magnitude larger than the (sub-)nanosecond scale accessible by QM/MM simulations. Transition path sampling (TPS) allows to efficiently sample the space of dynamic trajectories from the reactant to the product state without an additional biasing potential. We outline here the application of TPS and QM/MM to calculate rates for biochemical reactions, by means of a simple toy system. In a step-by-step protocol, we specifically refer to our implementation within the MD suite Gromacs, which we have made available to the research community, and include practical advice on the choice of parameters.

Key words Protein folding, Biochemical reactions, QM/MM, Reactive paths, Rate calculations

1 Introduction

Processes such as biochemical reactions or conformational changes of biomolecules typically occur on timescales beyond those accessible by Molecular Dynamics (MD) simulations at atomistic detail. In many cases, reducing the resolution of the simulation by coarse-graining the biomolecule is not an option, as critical players such as hydrogen bonds or hydrophobic effects involved in the reaction under investigation might be lost or are described at insufficient accuracy.

Purely classical MD simulations at atomistic resolution routinely can reach microsecond time scales. In a few recent cases, millisecond scales were achieved, which allowed the prediction of quantitative rates for the folding of proteins, either by highly parallel distributed computing of many short trajectories or by special purpose high-performance computing to obtain a small number of ultralong trajectories [1–3]. However, the conventionally reached microsecond time scale is mostly insufficient to sample the process of interest such as a conformational change or (un)folding frequently enough to compute transition rates.

The problem of too short simulation time scales is even larger for the case for chemical reactions, in which covalent bonds are broken or formed. Here, a classical molecular mechanical (MM) description is not sufficient, as it relies on a harmonic potential for a covalent bond, which does not allow dissociation of the bonded atoms. Instead, a quantum mechanical (QM) description is required to treat the change in bonds within the biomolecule accurately. Taking the electronic degrees of freedom into account, however, entails substantially higher computational costs and restricts time scales typically to picoseconds or nanoseconds, very much depending on the theory and basis set of choice. In turn, chemical reactions typically feature high barriers, i.e., rates at the microsecond to millisecond scale.

From a computational point of view, the most interesting quantity for such processes often is the reaction rate. The reason is that, in contrast to a free energy barrier, rates are experimentally directly accessible, and thus a straightforward comparison is possible. Also, the rate is the quantity which is physiologically most relevant, as kinetics determine most of the biological processes. Rates can be obtained from free energy barriers using the Arrhenius or Eyring equations, which requires, however, the assumption of an attempt frequency, the value of which is debated and varies with the nature of the process and the solvent [4, 5]. An elaborate method to directly compute reaction rates is Transition Path Sampling (TPS) [6, 7]. TPS is an algorithm which efficiently searches the space of transition paths between two states. From the ensemble of sampled paths obtained from MD simulations combined with a Monte Carlo sampling scheme, reaction rates can be obtained, without the detour of free energy barriers. TPS can be straightforwardly used for chemical reactions treated with QM or combined QM/MM. It has been proven to be a useful method to obtain quantitative insight into the mechanism of, among others, the reactions catalyzed by lactate dehydrogenase [8] and human purine nucleotide phosphorylase [9]. We have employed QM/MM and TPS to obtain force-dependent rates for a redox reaction, namely, the reduction of a disulfide bond by a small reducing agent, dithiothreitol [10], and for peptide hydrolysis [11].

In this chapter, we outline the basics of TPS, and in particular the calculation of reaction rates based on TPS. We illustrate the methodological details by way of a toy model, namely, three argon atoms in a box of water. While our toy model is, for simplicity, described solely by MM, the same strategy can be employed to a chemical or enzymatic reaction treated by combined QM/MM. For the reader interested in the details of a QM/MM setup for Molecular Dynamics simulations and eventually for TPS, we refer to recent reviews on this subject [12–14], and in combination with TPS to our own work [10]. TPS does not restrict the QM–MM interface in any way. However, as the TPS and rate calculation scheme presented here is based on our implementation into Gromacs [15], only QM/MM features available within Gromacs can be employed along with our implementation. The recent review by Groenhof [14] is the most comprehensive introduction into the current QM/MM implementation within Gromacs, and reviews the available schemes to treat the interactions between the regions described by QM and MM, and to cap the QM region in case of covalent bonds at the QM–MM interface.

2 Theory

Many processes such as chemical reactions or protein folding can be simplified to processes with two stable states that are separated by a single high energy barrier. In Fig. 1a, regions A and B are the two stable states, and the energy barrier is highlighted in between. For chemical reactions, regions A and B represent the reactant and product states, respectively. In this example, the multidimensional space of the system is projected onto two order parameters, R1 and R2, both of which change during the reaction. Examples for order parameters, often distances, angles, or collective coordinates, are given further below. A reactive trajectory (shown as a black solid line) leads to the rare but crucial transition between A and B. The system spends considerably longer times in the two free energy wells of the reactant and product than in the high free energy states between the two. Thus, while the transition of interest might only take a few 100 fs, the dwell time of the system in A or B might be in the microsecond to second time scale. Transition path sampling (TPS) has been developed to enhance the sampling of the rare reactive trajectories, which are otherwise hardly harvested by conventional simulations [6, 7, 16–18].

2.1 Sampling the Transition Path Ensemble

The idea of transition path sampling (TPS) is to sample a new transition path based on an existing (old) one (a transition path refers to a reactive trajectory) with a Monte Carlo procedure, and the new path is made sure to be equally weighted with the old one in the transition path ensemble. In principle, there are many strategies to do this. For illustrating the concept of TPS, we here use the shooting move in a deterministic simulation as an example.

(a) *Defining the probability of a reactive path.* In molecular simulations, the time evolution of a system is represented by an ordered sequence of states, $X(T) \equiv \{X_0, X_{\Delta t}, X_{2\Delta t}, \ldots, X_T\}$ (*see* Fig. 1a, black solid line). Here, Δt is the time increment. $X(T)$ consists of $L = T/\Delta t + 1$ states, and its starting point is X_0.

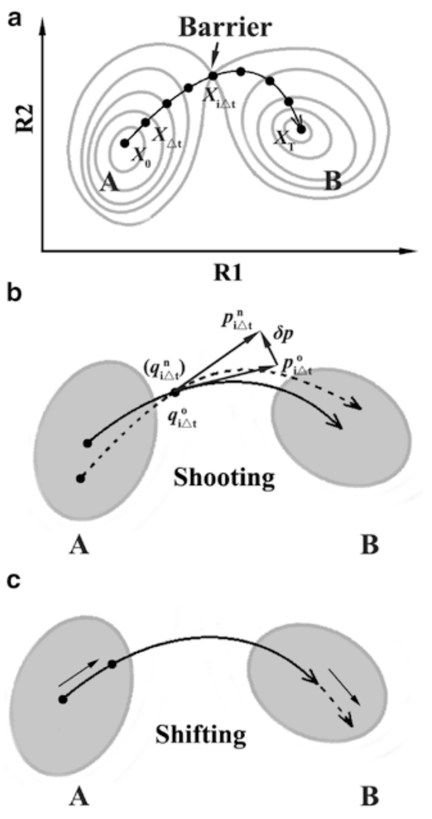

Fig. 1 Schematic description of the free energy landscape of a system and the shooting and shifting moves in TPS. (**a**) A typical free energy landscape of a process is shown with two stable states (labelled with A and B) and a barrier in the *middle*. R1 and R2 are two arbitrary coordinates. A transition pathway (*black solid line*) connecting states A and B is given as well. The transition path is represented by an ordered sequence of states $X(T) \equiv \{X_0, X_{\Delta t}, X_{2\Delta t}, \ldots, X_T\}$. (**b**) An example of shooting moves. The two *filled grey areas* represent the states A and B mentioned above. A state $\{q^{o}_{i\Delta t}, p^{o}_{i\Delta t}\}$ is randomly chosen from an old transition path (*solid line*). The momentum $p^{o}_{i\Delta t}$ is perturbed to be $p_{i\Delta t}{}^{n}$, where $p_{i\Delta t}{}^{n} = p_{i\Delta t}{}^{o} + \delta p$, while the coordinate is unchanged with $q^{o}_{i\Delta t} = q_{i\Delta t}{}^{n}$. From the newly generated state $\{q_{i\Delta t}{}^{n}, p_{i\Delta t}{}^{n}\}$, a new transition path (*dashed line*) is obtained by evolving the system backward in time to zero and forward in time to *T*. (**c**) An example of forward shifting moves. A new path is generated by removing a small segment from the beginning of the old path (the starting frame, shown as a *black dot*, moves forward to a new start) and evolving the system forward from the last frame to create a new part with the same length as the removed one (the *dashed line*). Figure adopted from Hierarchical Methods for Dynamics in Complex Molecular Systems, Lecture Notes, Eds. Grotendorst et al, Juelich, 2012" with permission

For deterministic dynamics, the probability of a trajectory equals to the probability of the initial state in a given ensemble, $\rho(X_0)$. Therefore, the probability of trajectory $X(T)$ to be a reactive trajectory is given below:

$$P_{AB}\left(X(T)\right) = h_A\left(X_0\right)\rho\left(X_0\right)h_B\left(X_T\right) / Z_{AB}(T) \tag{1}$$

Here, $h_A(X)(h_B(X))$ is the characteristic function of region A(B). $h_A(X)$ equals 1 if state X lies in A, and it equals zero otherwise. $Z_{AB}(T)$ is the normalizing factor, the sum of all the possible reactive trajectories with length T in a given ensemble.

$$Z_{AB}(T) \equiv \int \mathrm{d}X_0 h_A(X_0)\rho(X_0)h_B(X_T) \tag{2}$$

(b) *Sampling the transition path ensemble by shooting*. In a transition path ensemble, the distribution of transition paths is given in Eq. 1. To make sure that the correctly weighted transition paths are sampled, the following two probabilities should equal: the probability to generate a new transition path from a old one $P_{\text{gen}}(X^{\text{o}}(T) \to X^{\text{n}}(T))$, and the probability to generate the old transition path from the new one $P_{\text{gen}}(X^{\text{n}}(T) \to X^{\text{o}}(T))$. In a shooting move, a state $X_{i\Delta t}{}^{\text{o}}$, $i \in [0, L]$, is randomly chosen. Then, a new state $X_{i\Delta t}{}^{\text{n}}$ is generated by adding a small perturbation to $X_{i\Delta t}{}^{\text{o}}$. Here, the superscript o and n refer to the old path and the new path, respectively. Note that a state X consists of the coordinate q and the momentum p, $X = \{q, p\}$, the perturbation can be added to q or/and p. In practice, it is convenient to keep q untouched and change p by δp. As illustrated in Fig. 1b, the selected state $X_{i\Delta t}{}^{\text{o}} = \{q_{i\Delta t}{}^{\text{o}}, p_{i\Delta t}{}^{\text{o}}\}$ in an old transition path (the solid line in Fig. 1b) is changed to $X_{i\Delta t}{}^{\text{n}} = \{q_{i\Delta t}{}^{\text{n}}, p_{i\Delta t}{}^{\text{n}}\}$, where $p_{i\Delta t}{}^{\text{n}} = p_{i\Delta t}{}^{\text{o}} + \delta p$. Starting with $X_{i\Delta t}{}^{\text{n}}$, one can evolve the system backward in time to 0 and forward in time to T, then a new transition path is generated if it initials from region A and ends in region B (the dashed line in Fig. 1b). The probability to generate a new transition path from an old one is the product of four parts, the probability of the old path in the given ensemble, the probability to generate $X_{i\Delta t}{}^{\text{n}}$ from $X_{i\Delta t}{}^{\text{o}}$ ($P_{\text{gen}}(X_{i\Delta t}{}^{\text{o}} \to X_{i\Delta t}{}^{\text{n}})$), the probability of that the new path is reactive, and the probability to accept the new transition path $P_{\text{acc}}(X^{\text{n}}(T) \to X^{\text{o}}(T))$.

$$\begin{aligned} P_{\text{gen}}\left(X^{\text{o}}(T) \to X^{\text{n}}(T)\right) &= P_{\text{AB}}\left(X^{\text{o}}(T)\right) P_{\text{gen}}\left(X^{\text{o}}_{i\Delta t} \to X^{\text{n}}_{i\Delta t}\right) h_{\text{A}}\left(X^{\text{n}}_0\right) h_{\text{B}}\left(X^{\text{n}}_T\right) \\ &\quad \times P_{\text{acc}}\left(X^{\text{o}}(T) \to X^{\text{n}}(T)\right) \end{aligned} \tag{3}$$

Similarly, for generating the old path from the new one, we have

$$\begin{aligned} P_{\text{gen}}\left(X^{\text{n}}(T) \to X^{\text{o}}(T)\right) &= P_{\text{AB}}\left(X^{\text{n}}(T)\right) P_{\text{gen}}\left(X^{\text{n}}_{i\Delta t} \to X^{\text{o}}_{i\Delta t}\right) h_{\text{A}}\left(X^{\text{o}}_0\right) h_{\text{B}}\left(X^{\text{o}}_T\right) \\ &\quad \times P_{\text{acc}}\left(X^{\text{n}}(T) \to X^{\text{o}}(T)\right) \end{aligned} \tag{4}$$

The *detailed balance* of moves in trajectory space requires P_{gen} $(X^{\text{o}}(T) \to X^{\text{n}}(T)) = P_{\text{gen}}(X^{\text{n}}(T) \to X^{\text{o}}(T))$, which gives

$$\frac{P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)}{P_{\text{acc}}\left(X^{\text{n}}(T)\rightarrow X^{\text{o}}(T)\right)}=\frac{P_{\text{AB}}\left(X^{\text{n}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)h_{\text{A}}\left(X_{0}^{\text{o}}\right)h_{\text{B}}\left(X_{T}^{\text{o}}\right)}{P_{\text{AB}}\left(X^{\text{o}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)} \quad (5)$$

This condition can be satisfied using a Metropolis criterion [19]

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=\min\left[1,\frac{P_{\text{AB}}\left(X^{\text{n}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)h_{\text{A}}\left(X_{0}^{\text{o}}\right)h_{\text{B}}\left(X_{T}^{\text{o}}\right)}{P_{\text{AB}}\left(X^{\text{o}}(T)\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)}\right] \quad (6)$$

Note that the old path is reactive, i.e., $h_{\text{A}}(X_0^{\text{o}})=1$ and $h_{\text{B}}(X_T^{\text{o}})=1$. Equation 6 can be simplified as

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right)\times\min\left[1,\frac{\rho\left(X_{i\Delta t}^{\text{n}}\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}}\right)}{\rho\left(X_{i\Delta t}^{\text{o}}\right)P_{\text{gen}}\left(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}}\right)}\right] \quad (7)$$

Here, we apply Eq. 1 and the fact that the probabilities of the states on the same path in deterministic dynamics are the same. Although Eq. 7 is obtained based on deterministic dynamics, it can be also inferred based on a general dynamics [18]. In the implementation of shooting moves, a symmetric generation probability is normally ensured, and thus $P_{\text{gen}}(X_{i\Delta t}^{\text{o}}\rightarrow X_{i\Delta t}^{\text{n}})=P_{\text{gen}}(X_{i\Delta t}^{\text{n}}\rightarrow X_{i\Delta t}^{\text{o}})$. Specific strategies are always applied to ensure that states $X_{i\Delta t}^{\text{o}}$ and $X_{i\Delta t}^{\text{n}}$ are within the same microcanonical ensemble, i.e., $\rho(X_{i\Delta t}^{\text{o}})=\rho(X_{i\Delta t}^{\text{n}})$. Thus, the acceptance probability becomes

$$P_{\text{acc}}\left(X^{\text{o}}(T)\rightarrow X^{\text{n}}(T)\right)=h_{\text{A}}\left(X_{0}^{\text{n}}\right)h_{\text{B}}\left(X_{T}^{\text{n}}\right) \quad (8)$$

This equation states that any new trajectory will be accepted if it initiates from region A and ends in region B.

A new path can be generated by evolving forward from the last frame (forward shifting move, *see* Fig. 1c) or backward from the starting frame (backward shifting move) of the old path to grow a new path segment with a certain length and then deleting a path segment with the same length from the other end to maintain a fixed total length. Such shifting moves can be combined with shooting moves to improve the sampling efficiency.

2.2 Computing Rate Constants

In this section, we explain how to obtain rate constants from the transition path ensemble [17]. Given a system with two stable states A and B, which are separated by a single high energy barrier, molecules transit from one state to the other at equilibrium, while the populations of states remain unchanged. Since such transitions are rare, the time correlation function, $C(t)$, relates to the reaction time of the system ($\tau_{\text{rxn}}\equiv(k_{\text{AB}}+k_{\text{BA}})^{-1}$) via the following formula [20]

$$C(t)\approx\langle h_{\text{B}}\rangle\left(1-\exp\{-t/\tau_{\text{rxn}}\}\right) \quad (9)$$

If the time required for a system to cross the energy barrier and commit to the other stable state (τ_{mol}) is far smaller than the reaction time of the system (i.e., $\tau_{\text{mol}} << \tau_{\text{rxn}}$), $C(t)$ scales linearly in the intermediate time region, and we have

$$C(t) \approx k_{\text{AB}}t, \quad \tau_{\text{mol}} < t << \tau_{\text{rxn}} \tag{10}$$

For a system at equilibrium, $C(t)$ characterizes the conditional probability to find the system in state B at time t, if it was in state A at time zero, and is defined as follows

$$C(t) \equiv \frac{\langle h_{\text{A}}(X_0) h_{\text{B}}(X_t) \rangle}{\langle h_{\text{A}}(X_0) \rangle} \tag{11}$$

Here, $\langle \ldots \rangle$ is the ensemble average of all initial states. In deterministic dynamics, $C(t)$ can be written in terms of the probability of all initial states $\rho(X_0)$:

$$C(t) = \frac{\int dX_0 \rho(X_0) h_{\text{A}}(X_0) h_{\text{B}}(X_t)}{\int dX_0 \rho(X_0) h_{\text{A}}(X_0)} \tag{12}$$

Equations 10 and 12 together provide a way to calculate the forward reaction rate constant k_{AB} by molecular simulations. One can simply run a large set of simulations that start in region A and are of the same time length t, and then count the probability of the end state to be in region B, which gives the value of $C(t)$. The derivative of $C(t)$ over time gives the rate constant. However, this apparently involves numerous computational efforts.

If region B can be defined by an order parameter $\lambda(X)$, and the distribution of the end states, i.e., $X(t)$, along the order parameter $P(\lambda, t)$ is known, $C(t)$ is simply the integral of $P(\lambda, t)$ along λ over region B.

$$C(t) = \int_{\lambda\_\min}^{\lambda\_\max} d\lambda P(\lambda, t). \tag{13}$$

Here, λ_min and λ_max are the lower and upper bound of region B along λ. $P(\lambda, t)$ is given by

$$P(\lambda, t) = \frac{\int dX_0 \rho(X_0) h_{\text{A}}(X_0) \delta\left[\lambda - \lambda(X(t))\right]}{\int dX_0 \rho(X_0) h_{\text{A}}(X_0)}, \tag{14}$$

where $\delta(X)$ is Dirac's delta function. $P(\lambda, t)$ can be divided into several overlapped windows, and its distribution in each window can be estimated separately. The distribution of $P(\lambda, t)$ over the whole range of λ is then obtained by connecting all windows. In each window, transition path sampling can be applied to enhance the sampling of paths that connect region A and the window region. Therefore, computational efforts to compute $C(t)$ are dramatically reduced.

The above-mentioned method can only compute $C(t)$ in time t at a time, and the evaluation of k_{AB} requires $C(t)$ at different times to be evaluated. Therefore, it is laborious. Fortunately, $C(t)$ can be factorized to be written as [18]

$$C(t) = \frac{\langle h_B(t) \rangle_{AB}}{\langle h_B(t') \rangle_{AB}} C(t'), \quad 0 < t < T \tag{15}$$

where $\langle \ldots \rangle_{AB}$ denotes an average on the ensemble of the reactive paths, which start in region A and visit region B within the time length of T. T is the time length of the transition path. $\langle h_B(t) \rangle_{AB}$ is then the proportion of reactive paths whose configuration at time t belongs to region B, and can be estimated by a single transition path sampling run. Only $C(t')$, the $C(t)$ at time $t'(t' < T)$, is needed to be evaluated.

Combining Eqs. 10 and 15, the rate constant is given by

$$k_{AB} = \frac{\mathrm{d}\langle h_B(t) \rangle_{AB} / \mathrm{d}t}{\langle h_B(t') \rangle_{AB}} \times C(t'), \quad \tau_{mol} < t << \tau_{rxn} \tag{16}$$

$\mathrm{d}\langle h_B(t) \rangle_{AB}/\mathrm{d}t$ should show a plateau in the intermediate time range.

3 Materials

A GROMACS-4.0.7 package [15] with a TPS implementation can be downloaded from http://wenjin.people.uic.edu/download/Gromacs4_tps_patch.tar.gz, which is implemented by Dr. Wenjin Li and currently maintained by him as well (*see* **Note 1**). The package can be installed by following the installation instructions of the original GROMACS-4.0.7 version at http://www.gromacs.org. A Linux or Unix system is required for compilation, as well as FFTW libraries.

4 Methods

In this section, we will describe how to (1) establish a toy system, (2) define the stable basins, (3) obtain the $\langle h_B(t) \rangle_{AB}$ curve, (4) obtain the $P(\lambda, t)$ distribution, (5) calculate rate constants, and (6) monitor TPS. All the files necessary to complete this tutorial are available at http://wenjin.people.uic.edu/download/example_3_Ar.tar.gz.

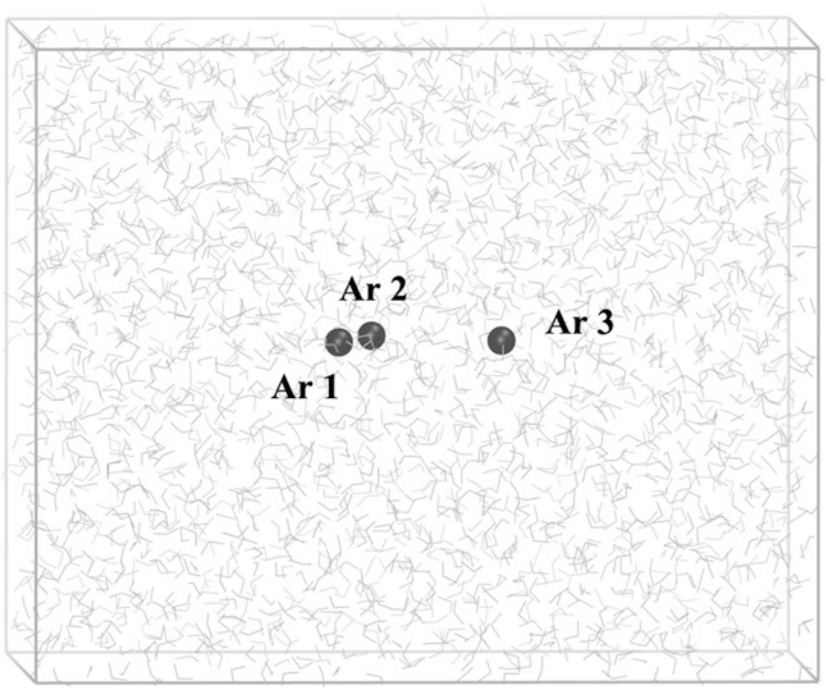

Fig. 2 Simulation setup of the toy system. *Black spheres*: Ar atoms. *Grey lines*: water molecules

To complete this tutorial, we assume the reader to have a basic knowledge of GROMACS and experience in the use of a Linux or Unix operation system.

4.1 A Toy System

We here will illustrate how to use TPS to calculate the rate constant of a rare event with a toy system, which consists of three Ar atoms in a water box (*see* Fig. 2). All three Ar atoms are lying in a line along the *Z*-axis. Atoms 1 and 3 are held by position restraints along the *X*-, *Y*-, and *Z*-axis, while atom 2 is restrained along the *X*- and *Y*-axis, but free to move along the *Z*-axis. Position restrains were switched on by setting *define= -DPOSRES* in the .mdp file, with parameters for position restraints given in *posre.itp*. Atoms 1 and 3 are separated by approximately 1.0 nm. Due to the van der Waals interaction with the other two Ar atoms, atom 2 has two preferred positions (or stable basins). One position is about 0.2 nm, the other 0.8 nm away from atom 1. There is a relatively high barrier between the two minima. Atom 2 can overcome the attraction of one Ar atom and transit from one stable basin to the other. Here, we will estimate the rate of these transitions with TPS. The parameters for van der Waals interaction between two Ar atoms have been modified to unrealistic values (see file *ffoplsaanb.itp*) to increase the barrier between the two minima to make sure that the transition is a rare event (*see* **Note 2**). Therefore, we here are looking at an unphysical toy model to solely focus on the procedure to run TPS with the modified GROMACS package.

4.1.1 Definition of Stable Basins

TPS requires reasonable definitions of the stable basins in terms of one or multiple order parameters. The chosen order parameter should be able to distinguish the two stable basins, but must not necessarily be a good reaction coordinate. Here, the order parameter we choose to distinguish the two minima is $\Delta d = d_{12} - d_{23}$, where d_{12} is the distance between atoms 1 and 2, and d_{23} is the distance between atoms 2 and 3. Then, region A is defined as $-1 < \Delta d < -0.5$ and region B as $0.5 < \Delta d < 1$. The modified Gromacs version includes a section to define these TPS parameters. To define the stable states mentioned above, the parameters are (*see* **Note 3**),

```
========================
tps_npost            = 4
tps_grps1            = a_1 a_2
tps_grps2            = a_2 a_3
tps_dimension        = one
tps_weight_dim       = 1   -1
tps_initial_max      = -0.5
tps_initial_min      = -1
tps_final_max        = 1
tps_final_min        = 0.5
========================
```

Here, *tps_grps1* and *tps_grps2* define the groups to build the order parameters. Currently, the order parameters consist of only distances between two atoms (or two groups of atoms). *tps_grps1* specifies the first group, while *tps_grps2* specifies the second group. The coordinate is the distance between the *i*th group in *tps_grps1* and *tps_grps2*. For example, d_{12} is calculated by the distance between the first group in *tps_grps1* and *tps_grps2*. *a_1*, *a_2*, and *a_3* are the name of atom 1, atom 2, and atom 3 in the index file. *tps_dimension = one* means the two coordinates specified by *tps_grps1* and *tps_grps2* are combined into one parameter using the weights in *tps_weight_dim*. *tps_weight_dim = 1 -1* means the order parameter $\Delta d = 1 \times d_{12} + (-1) \times d_{23}$ or $d_{12} - d_{23}$ (*see* **Note 4**). The values of the upper bound and lower bound of region A and region B are given by *tps_initial_max*, *tps_initial_min*, *tps_final_max*, and *tps_final_min*. The number of groups in *tps_grps1* and *tps_grps2* should be the same. *tps_npost* is the total number of groups in *tps_grps1* and *tps_grps2*.

4.2 Obtaining an Initial Transition Path

In TPS, the shooting and shifting moves are based on an initial path, which is not necessary to be physically meaningful, as the subsequent TPS will allow to relax towards more representative paths (*see* **Note 5**). There are many ways to get an initial path. Here, we generate the first path by shooting forward and backward from a structure near the transition state to ensure that we can get an initial reactive path with high probability. Velocities are adapted from a Boltzmann distribution at the given temperature. The setting of the parameters in the TPS section are (the complete parameter file is available in *tps_ini.mdp* provided in the tutorial package),

```
=====   Part of tps_ini.mdp   =====
tps_npost                  = 4
tps_grps1                  = a_1 a_2
tps_grps2                  = a_2 a_3
tps_dimension              = one
tps_weight_dim             = 1    -1
tps_initial_max            = -0.5
tps_initial_min            = -1
tps_final_max              = 1
tps_final_min              = 0.5
tps                        = rand_ini
tps_maxcycle               = 5
tps_maxshoot               = 10
tps_endpoint               = yes
tps_kin_ref                = 100
tps_Temperature            = 300
tps_forward_steps          = 400
tps_backward_steps         = 400
tps_maxframe               = 1
tps_ntrrout                = 1
=================================
```

tps=rand_ini defines the attempt to get an initial transition path. *tps_maxcycle* and *tps_maxshoot* specifies the number of sampling circles and the number of samples in each circles. Here, we try $5 \times 10 = 50$ times to get an initial path. The search will stop immediately if we find one. *tps_endpoint=yes* means the endpoint of the path should be in region B. *tps_forward_steps* and *tps_backward_steps* specify the number of frames saved for forward and backward shooting, respectively. The total frames of the trajectory generated will be the sum of them. Here, the number of frames in the transition path will be 800. The frequency of saving frames in the transition path is defined by *nstxout=10* and the integration step is 2 fs (*see* **Note 6**). Therefore, we will obtain a reactive path with 800 frames, with an interval between each frame of 20 fs and a total length of $t = 16$ ps. *tps_Temperature=300* produces initial atomic velocities from a Boltzmann distribution at a temperature of 300K. *tps_kin_ref* specifies the amount of perturbation to the momenta of the atoms in the frame to which a shooting move is applied. The value will affect the acceptance ratio of shooting moves, with larger perturbations leading to smaller acceptance ratios. *tps_maxframe* is the number of frames in the input trajectory. In this case, *tps_maxframe=1* because the input is a .gro file which contains only one frame. *tps_ntrrout=1* makes the MD code saving every reactive trajectories.

The following input commands initiate the search for an initial path:

```
tar -zxvf example_3_Ar.tar.gz

cd example_3_Ar && mkdir -p tps/initial

grompp -f tps_ini.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o tps/initial/tps.tpr

cd tps/initial &&  mdrun -s tps.tpr -rerun ../../TS.gro -deffnm tps_output
```

The input structure is read via option *-rerun*. *TS.gro* is the structure near the transition region. *3_Ar.gro* is a structure at region A. *index.ndx* and *topol.top* define the index of groups used in the .mdp file and the topology of the system, respectively. Tutorials to prepare these files can be found elsewhere [21] and is out of the scope of this chapter. We therefore provide them in the tutorial package. This run will take about 10 min on a single processor to generate an initial transition path saved as *traj_0.trr*. We rename it as *tps.trr* by executing

```
mv traj_0.trr ../tps.trr
```

4.3 Obtaining the $\langle h_B(t)\rangle_{AB}$ Curve

A requisite to compute the rate constant using TPS is the flux versus time, or the $\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}$ curve, and the probability distribution along an order parameter $P(\lambda, t)$, or specifically $P(\Delta d)$ in this case, which is then used to calculate the value of $C(t)$ at a specific time t (see Theory). With an initial path at hand, we can start TPS to obtain these ingredients for the rate constant calculations. The settings for this purpose are:

```
=====Part of tps.mdp =====
tps_npost           = 4
tps_grps1           = a_1 a_2
tps_grps2           = a_2 a_3
tps_dimension       = one
tps_weight_dim      = 1  -1
tps_initial_max     = -0.5
tps_initial_min     = -1
tps_final_max       = 1
tps_final_min       = 0.5
tps                 = normal
tps_maxcycle        = 150
tps_maxshoot        = 10
tps_maxshift        = 10
tps_endpoint        = no
tps_kin_ref         = 100
tps_reput_length    = 300
tps_maxframe        = 800
tps_ntrrout         = 0
=========================
```

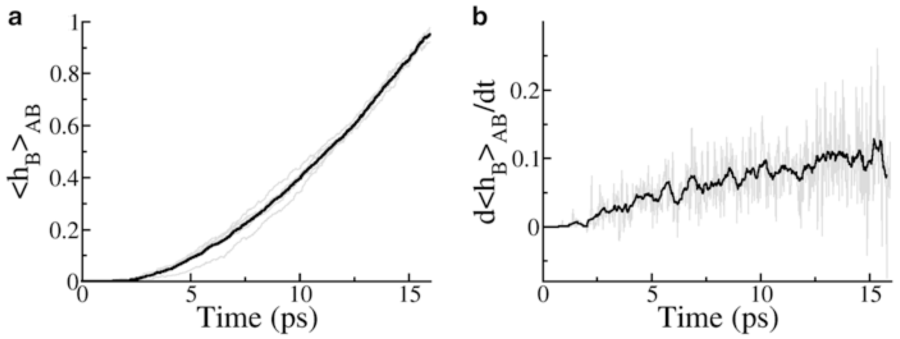

Fig. 3 Results for the $\langle h_B(t)\rangle_{AB}$ curve. (**a**) *Black curve*: the averaged $\langle h_B(t)\rangle_{AB}$ curve. *Grey curves*: the five $\langle h_B(t)\rangle_{AB}$ curves obtained from five independent samplings. (**b**) The derivative of the $\langle h_B(t)\rangle_{AB}$ curve shows a plateau, indicating a length of 16 ps to be sufficient. *Grey*: the derivative of the *black curve* in **a**. *Black*: the smoothed curve of the *grey* one by averaging over five nearby points

tps=normal means that we now perform TPS with an initial path already at hand. *tps_maxcycle* specifies the number of TPS cycles. In each cycle, we perform several shooting moves and shifting moves, and the number of these moves is specified by *tps_maxshoot* and *tps_maxshift*, respectively. *tps_endpoint=no* means the end structure of a reactive trajectory, given the trajectory starts in A, may not be in B, but the structure should reach B at some point within the trajectory (see Theory). *tps_reput_length* specifies the maximum shifting length in shifting moves. *tps_ntrrout =0* means no intermediate reactive trajectories are saved.

Performing the TPS requires executing the following commands:

```
cd ../../ && grompp -f tps.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o tps/tps.tpr

cd tps && mdrun -s tps.tpr -rerun tps.trr -deffnm tps_output
```

We read the initial reactive path again via the option *-rerun*. Here, we run 150 cycles of TPS, with 10 shooting moves and 10 shifting moves in each cycle. In total there are 3,000 TPS runs. This will take about 4 days to complete on a single standard processor. The results of the $\langle h_B(t)\rangle_{AB}$ curve is saved in *hahb.dat*. To obtain an accurate $\langle h_B(t)\rangle_{AB}$ curve, we recommend the reader to run five independent simulations (*see* **Note** 7), and to then combine the resulting five hahb.dat files into one by simple averaging (Fig. 3a). Here, the derivative of $\langle h_B(t)\rangle_{AB}$ reaches a plateau at 13 ps with $d\langle h_B(t)\rangle_{AB}/dt = 0.1\ \text{ps}^{-1}$ as shown in Fig. 3b (*see* **Note 8**).

4.4 Obtaining the P(λ, t) Distribution

In the next step, we run TPS in different windows to obtain the distribution of the end points of transition paths (the $P(\Delta d)$ distribution in Eq. 14). Here, we set the length of the trajectory t to be $t' = 6$ ps, with each path containing 300 frames, which is much shorter than 16 ps or 800 frames and saves computational cost (*see* **Note 9**). Therefore, we read $\langle h_B(t')\rangle_{AB} = 0.14$ from Fig. 3a, as $t' = 6$ ps. In order to get the $P(\Delta d)$ distribution, we divide the configuration space into five windows, which are defined as window 1: $-1 < \Delta d < -0.45$, window 2: $-0.55 < \Delta d < -0.15$, window 3: $-0.25 < \Delta d < 0.25$, window 4: $0.15 < \Delta d < 0.55$, and window 5: $0.45 < \Delta d < 1$. A small overlap between adjacent windows is necessary to merge the distributions of adjacent windows into one. By deleting the segments from the termini of the transition path obtained above (16 ps long), one can easily get an initial path of 6 ps long (*see* **Note 10**). The sampling procedures in each window are similar. For each window, we define the correct region B and adjust *tps_kin_ref* and *tps_reput_length* to maintain a reasonable acceptance ratio. The parameter files for all windows are provided in the tutorial package and are named as *tps_win1.mdp* to *tps_win5.mdp*. We here show the TPS parameters for window 5 as an example to illustrate the procedure.

```
=====Part of tps_win5.mdp =====
tps_npost             = 4
tps_grps1             = a_1 a_2
tps_grps2             = a_2 a_3
tps_dimension         = one
tps_weight_dim        = 1    -1
tps_initial_max       = -0.5
tps_initial_min       = -1
tps_final_max         = 1
tps_final_min         = 0.45
tps                   = normal
tps_maxcycle          = 300
tps_maxshoot          = 10
tps_maxshift          = 10
tps_endpoint          = yes
tps_kin_ref           = 200
tps_reput_length      = 100
tps_maxframe          = 300
tps_ntrrout           = 0
==========================
```

To obtain the endpoint distribution, we need to make sure the endpoints of the transition path to be within the defined region B by setting *tps_endpoint=yes*. The simulation is started as follows:

```
cd ../ && mkdir win5
grompp -f tps_win5.mdp -c 3_Ar.gro -n index.ndx -p topol.top -o win5/tps_win5.tpr
cd win5 && mdrun -s tps_win5.tpr -rerun tps_win5.trr -deffnm tps_output
```

Here, *tps_win5.trr* is the constructed initial transition path. The endpoints of each transition path are saved in *endpoint.dat*.

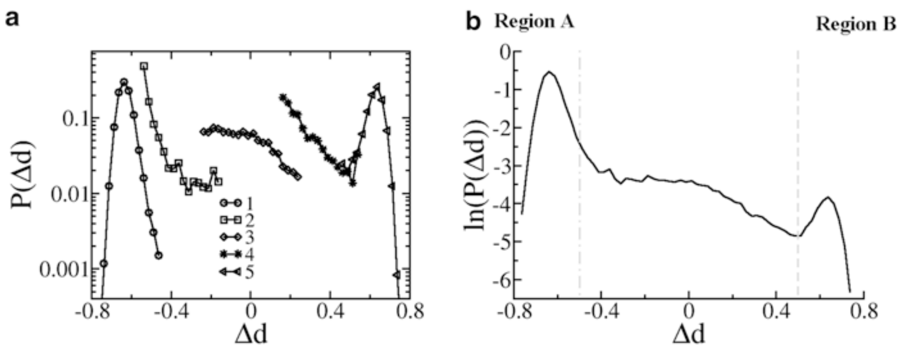

Fig. 4 Calculation of $P(\Delta d)$ through TPS in windows. (**a**) Distribution of $P(\Delta d)$ in different windows. (**b**) The connected distribution of $P(\Delta d)$ over the whole configuration space. *Dashed grey lines*: the boundaries between regions A and B and the transition region

We recommend the reader to run three independent simulations for each window and then collect all the endpoints into a single file of *endpoint.dat*. From the *endpoint.dat* file, the distribution along Δd can be easily obtained. The distributions of in the overlapped regions are the same but weighted differently. Therefore, we can connect all windows by re-weighting them properly. The connected window is then normalized. The distributions in different windows and the normalized distribution are shown in Fig. 4. By integrating the distribution in the range of $0.5 < \Delta d < 1$, we obtain a value of $C(t')$ of 0.00056 (see Theory).

4.5 Calculating the Rate Constant

Given $C(t')$ is 0.00056, $\langle h_B(t')\rangle_{AB}$ is 0.14, and $d\langle h_B(t)\rangle_{AB}/dt$ is 0.1 ps^{-1}, we get a rate constant k of 4.0×10^{-4} ps using Eq. 16 (*see* **Note 11**).

4.6 Monitoring TPS

The *mdrun* command will generate four output files that help to monitor the progress of the sampling: *acc.dat*, *endpoint.dat*, *hahb.dat*, and *summary.dat*. They are explained below:

acc.dat: It summarizes the number of shooting trials, the number of successful shooting trials, the number of shifting trials, and the number of successful shifting trials at each frame. It also includes the acceptance ratio for shooting and shifting.

endpoint.dat: It gives the endpoints of the transition paths in the value of the order parameter, which is used to calculate $P(\lambda, t)$ when *tps_endpoint=yes.*

hahb.dat: It gives the $\langle h_B(t)\rangle_{AB}$ curve when *tps_endpoint=no.*

summary.dat: It summarizes the overall number of shooting and shifting cycles and their acceptance ratio. An example is given below:

```
=============== summary.dat ==================
The totol TPS cycle is ---------------------------3000
The totol shooting cycle is -----------------------1524
The totol leftshift cycle is ----------------------747
The totol rightshift cycle is --------------------729
The totol acceptance is -------------------------0.3076667
The acceptance for shooting is ------------------0.0577428
The acceptance for leftshift is -----------------0.5689424
The acceptance for rightshift is -----------------0.5624143
=====================================
```

Here, the acceptance for shooting is quite low, around 0.06. Adjusting *tps_kin_ref* and *tps_reput_length* allows to tune the probability of generating a reactive trajectory. A higher acceptance ratio can be achieved by shortening the length of the transition path (*see* **Note 12**). To achieve a better sampling efficiency, the acceptance for shooting it recommended to be about 0.4 [22].

5 Notes

1. The modified GROMACS package supports simulations on only a single CPU and not in parallel, as neither domain decomposition nor particle decomposition are supported in the current implementation.
2. Equation 10 is based on the assumption that the barrier is so high that the time of the actual transition is much smaller than the inverse of the rate constant. Therefore, Eq. 10 is only applicable to systems with high energy barriers, i.e., of several $k_B T$.
3. For many systems, the choice of an order parameter is trivial. One can run a relatively long simulation at the two stable states, and then find an order parameter to distinguish the stable states by inspection of the coordinate spaces that the two simulations sampled at both basins. Usually, an inspection by eye is enough. If not, principle component analysis [23] can assist in identifying an order parameter. Once an order parameter is found, one defines the two stable states according to their distribution of the sampled configuration along the order parameter. Make sure that the two basins are separated and cover the major part of the sampled configurations in that state.
4. One can use multiple coordinates to define regions A and B if the interest is to investigate the mechanism of the transition process rather than the rate constant. If one want to get the rate constant, region A can be defined with multiple coordinates, while region B is preferably defined with a single coordinate, as this reduces the computational expense. If defining region B by multiple coordinates is nevertheless essential, the distribution of $P(\lambda, t)$ is required in the multidimensional

space, which might be feasible but is not recommended. It is generally beneficial to invest some efforts to find a single coordinate to define region B. The coordinates to define regions A and B are not required to be identical.

5. One can generate an initial transition path in many ways, depending on the system under investigation. In general, one can apply a bias to the system to enforce the transition to happen with high probability and short transition times. The bias can be from for example high temperature [24], replica exchange [24], position restraints [10], steering forces [26], conformational flooding [27], or metadynamics [28]. The resulting transition path is a biased transition path, but the bias can be removed gradually [26], or by generating an unbiased path by TPS with shooting moves starting from one frame (e.g., a frame within or close to the transition state ensemble) of the biased path.
6. The number of steps for a simulation defined by *nsteps* in the .mdp file should be larger than the total length of the TPS trajectory. Here, *nsteps* should be no less than 8,000 given the integral timestep of 2 fs.
7. The TPS simulation will generate files with predefined names in the working directory. If one runs multiple independent simulations, it is recommended to start them from separate directories to avoid overwriting output files.
8. In addition to the $\langle h_B(t)\rangle_{AB}$ curve, the simulation will harvest an ensemble of transition paths (one can save the transition paths by setting *tps_ntrrout* to a positive integer to specify the frequency of saving the transition paths). Based on the transition path ensemble, the mechanism of the studied rare events can be elucidated at the atomistic level. Applications include (just to name a few) a reaction catalyzed by lactate dehydrogenase [8], the β-hairpin folding [25], and the chorismate-mutase-catalyzed conversion of chorismate into prephenate [29]. If committor probabilities of the frames in the transition path ensemble are estimated, transition states [10, 29] and reaction coordinates [30] can be identified as well.
9. Choosing a small t' reduces the computational costs of the TPS, but increases the uncertainty of $\langle h_B(t')\rangle_{AB}$. As a compromise, we recommend to choose t' such that $\langle h_B(t')\rangle_{AB}$ is about 0.2. Usually, the cost to calculate $C(t')$ is several times higher than the cost to obtain $\langle h_B(t)\rangle_{AB}$. For this reason, it is worth to invest more efforts into an accurate $\langle h_B(t)\rangle_{AB}$ curve, and to then use a smaller t' to calculate $C(t')$.
10. The initial path for each window to obtain the $\langle h_B(t)\rangle_{AB}$ curve can be obtained following the same procedure as the one for the initial path for the TPS. Alternatively, the initial path for

window 5 should have 300 frames, which can be constructed by shortening the transition paths sampled in the section of obtaining the $\langle h_{\mathrm{B}}(t)\rangle_{\mathrm{AB}}$ curve, which comprises 800 frames. Then, the initial path for window 4 can be obtained from one of transition paths sampled in window 5. The initial path for other windows can be taken from one of transition paths from the subsequent window.

11. As a way of validating the computed rates, one can vary t and/or t' as well as the definitions of regions A and/or B to test if the results are quantitatively consistent.

12. It is not necessary to randomly select a frame from the entire transition path to do shooting moves. Specifying the range from which shooting frames are selected allows to increase the acceptance ratio. One possibility is to choose points near the previous shooting point from which a reactive trajectory has been generated. Alternatively, one can choose points only from the barrier region to improve the acceptance ratio. In this case the range to choose shooting points is variable, and the probability to accept a reactive trajectory needs to be modified accordingly [25].

Acknowledgment

We are grateful to the Klaus Tschira Foundation for financial support.

References

1. Lane TJ, Bowman GR, Beauchamp K et al (2011) Markov state model reveals folding and functional dynamics in ultra-long MD trajectories. J Am Chem Soc 133:18413–18419
2. Bowman GR, Pande VS (2010) Protein folded states are kinetic hubs. Proc Natl Acad Sci U S A 107:10890–10895
3. van der Spoel D, Seibert MM (2006) Protein folding kinetics and thermodynamics from atomistic simulations. Phys Rev Lett 96:238102
4. Best RB, Hummer G (2006) Diffusive model of protein folding dynamics with Kramers turnover in rate. Phys Rev Lett 96:228104
5. Popa I, Fernández JM, Garcia-Manyes S (2011) Direct quantification of the attempt frequency determining the mechanical unfolding of ubiquitin protein. J Biol Chem 286:31072–31079
6. Dellago C, Bolhuis PG, Csajka FS et al (1998) Transition path sampling and the calculation of rate constants. J Chem Phys 108:1964
7. Dellago C, Bolhuis PG, Chandler D (1998) Efficient transition path sampling: application to Lennard-Jones cluster rearrangements. J Chem Phys 108:9236
8. Quaytman SL, Schwartz SD (2007) Reaction coordinate of an enzymatic reaction revealed by transition path sampling. Proc Natl Acad Sci U S A 104:12253–12258
9. Saen-Oon S, Quaytman-Machleder S, Schramm VL et al (2008) Atomic detail of chemical transformation at the transition state of an enzymatic reaction. Proc Natl Acad Sci U S A 105:16543–16548
10. Li W, Gräter F (2010) Atomistic evidence of how force dynamically regulates thiol/disulfide exchange. J Am Chem Soc 132:16790–16795
11. Xia F, Bronowska AK, Cheng S et al (2011) Base-catalyzed peptide hydrolysis is insensitive to mechanical stress. J Phys Chem B 115:10126–10132
12. van der Kamp MW, Mulholland AJ (2013) Combined quantum mechanics/molecular mechanics (QM/MM) methods in computational enzymology. Biochemistry 52:2708–2728

13. Steinbrecher T, Elstner M (2013) QM and QM/MM simulations of proteins. In: Monticelli L, Salonen E (eds) Biomolecular simulations. Humana Press, New York, pp 91–124
14. Groenhof G (2013) Introduction to QM/MM simulations. In: Monticelli L, Salonen E (eds) Biomolecular simulations. Humana Press, New York, pp 43–66
15. Hess B, Kutzner C, Van Der Spoel D et al (2008) GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. J Chem Theory Comput 4:435–447
16. Bolhuis PG, Dellago C, Chandler D (1998) Sampling ensembles of deterministic transition pathways. Faraday Discuss 110:421–436
17. Dellago C, Bolhuis PG, Chandler D (1999) On the calculation of reaction rate constants in the transition path ensemble. J Chem Phys 110:6617
18. Dellago C, Bolhuis PG, Geissler PL (2002) Transition path sampling. Adv Chem Phys 123:1–78
19. Metropolis N, Rosenbluth AW, Rosenbluth MN et al (1953) Equation of state calculations by fast computing machines. J Chem Phys 21: 1087
20. Chandler D (1978) Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. J Chem Phys 68:2959
21. Lindahl EL (2008) Molecular dynamics simulations. In: Kukol A (ed) Molecular modeling of proteins. Humana Press, Totowa, NJ, pp 3–23
22. Bolhuis PG, Chandler D, Dellago C et al (2002) Transition path sampling: throwing ropes over rough mountain passes, in the dark. Annu Rev Phys Chem 53:291–318
23. Jolliffe I (2005) Principal component analysis. Wiley Online Library
24. Dellago C, Bolhuis PG, Geissler PL (2006) Transition path sampling methods. In: Computer simulations in condensed matter systems: from materials to chemical biology, vol 1. Springer, Berlin, pp 349–391
25. Bolhuis PG (2003) Transition-path sampling of β-hairpin folding. Proc Natl Acad Sci U S A 100:12129–12134
26. Hu J, Ma A, Dinner AR (2006) Bias annealing: a method for obtaining transition paths de novo. J Chem Phys 125:114101
27. Grubmüller H (1995) Predicting slow structural transitions in macromolecular systems: conformational flooding. Phys Rev E 52:2893
28. Laio A, Gervasio FL (2008) Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. Rep Prog Phys 71:126601
29. Crehuet R, Field MJ (2007) A transition path sampling study of the reaction catalyzed by the enzyme chorismate mutase. J Phys Chem B 111:5708–5718
30. Ma A, Dinner AR (2005) Automatic method for identifying reaction coordinates in complex systems. J Phys Chem B 109:6769–6779