# FSATOOL: A Useful Tool to do the Conformational Sampling and Trajectory Analysis Work for Biomolecules

Haomiao Zhang, Qiankun Gong, Haozhe Zhang, and Changjun Chen *

Reliable conformational sampling and trajectory analysis are always important to the study of the folding or binding mechanisms of biomolecules. Generally, one has to prepare many complicated parameters and follow a lot of steps to obtain the final data. The whole process is too complicated to new users. In this article, we provide a convenient and user-friendly tool that is compatible to AMBER, called fast sampling and analysis tool (FSATOOL). FSATOOL has some useful features. First and the most important, the whole work is extremely simplified into two steps, one is the fast sampling procedure and the other is the trajectory analysis procedure. Second, it contains several powerful sampling methods for the simulation on graphics process unit, including our previous mixing replica exchange molecular dynamics method. The method combines the advantages of the biased and unbiased simulations. Finally, it extracts the dominant transition pathways automatically from the folding network by Markov state model. Users do not need to do the tedious intermediate steps by hand. To illustrate the usage of FSATOOL in practice, we perform one simulation for a RNA hairpin in explicit solvent. All the results are presented. © 2019 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.26083

## Introduction

How a biomolecule transfers between its important states is interesting. In the past, molecular dynamics had been proven to be a practical approach for this study. However, due to the limitation of the hardware, molecular simulation on the computer is not easy to reach the real transition timescales of biomolecules in vivo (typically milliseconds to seconds).[1] Nowadays, a lot of software can perform the simulations on the powerful graphics process unit (GPU), like AMBER,[2] OpenMM, [3] and GROMACS.[4] With GPU, simulations of biomolecule are considerably accelerated.[5] Moreover, there could be a lot of free energy barriers on the free energy landscape. The barriers slow down the state-to-state transitions in the simulation. To improve the sampling speed further, many enhanced sampling methods have been developed, like replica exchange molecular dynamics (REMD),[6] metadynamics,[7,8] steered molecular dynamics (SMD),[9] transition path sampling,[10] fluctuation amplification of specific traits,[11] and so on. For example, metadynamics builds an adaptive biasing potential in the simulation. The potential is a history-dependent function summed by a series of Gaussians. With the potential, the original free energy landscape of a molecule can be flattened. Actually, metadynamics is more than a sampling method. It is also used for the calculation of the free energies and the determination of the rare events or the transition pathways.

Different simulation conditions correspond to different forms of the Hamiltonians. To obtain the unbiased sampling data in the canonical ensemble, it is helpful to use multiple replicas of a molecule in the simulation and allow them to exchange with each other in a fixed time period.[6,12] The total number of replicas should be large enough to ensure a high exchange rate. This demands amount of computation resources. In general, these replica exchange methods can be divided into T-REMD,[6,13] H-REMD,[14–17] and HT-REMD.[18–20] T-REMD[6,13] uses different replicas with different temperatures. High temperatures allow the replicas to cross the free energy barrier and sample the conformational space fast. H-REMD[14–17] applies different biasing potentials to the replicas. Well-tuned artificial potentials are able to push the replicas out of the free energy minima. In practice, the artificial potentials can be built by the well-known metadynamics method.[7,8] The last one, HT-REMD,[18–20] is a combination of T-REMD and H-REMD. In this simulation, both of the temperatures and the potentials of the replicas are modified. In any case, the sampling data of different replicas can be combined by weighted histogram analysis method[21] to construct the free energy surface.

Furthermore, for the sampling methods with artificial potentials,[22] a set of collective variables must be determined before the formal simulation. However, finding out a proper collective variable is not easy. It should be able to recognize the different metastable states of a molecule and capture the rare events between them. One might have to perform some independent simulations and analyze the distribution of the sampling data in the collective variable space by diffusion-map method,[23] time-lagged independent component analysis (TICA) method,[24] or machine-learning based method.[25] These methods determine the *slow* collective variables[26] by solving eigenfunctions of the dynamical operator (or propagator) according to refs. [27,28].

[a] *H. Zhang, Q. Gong, H. Zhang, C. Chen*
*Biomolecular Physics and Modeling Group, School of Physics, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China*
*E-mail: cjchen@hust.edu.cn*
Contract Grant sponsor: National Natural Science Foundation of China;
Contract Grant number: 31770773

Recently, we propose a mixing REMD method.[29] It combines the biased simulation with the unbiased simulation. Biased simulation is for the fast sampling in the collective variable space and the unbiased simulation at multiple temperatures is for the production of the unbiased data for the free energy calculation. To perform the simulation, three different kinds of replicas are used, called biased replicas, equilibrium replicas and unbiased REMD replicas. In the simulation, the states of the biased replicas are copied to the equilibrium replicas at the fixed steps. After another fixed time steps, the states of the equilibrium replicas are further transferred to the unbiased REMD replicas according to the Metropolis criterion.[30] By the method, one can freely choose the number of the replicas and the collective variables. Free energy calculation is also convenient. No reweighting steps are required for different collective variables.

An accurate free energy surface of a biomolecule provides the distribution of its important states in the collective variable space.[31] But the surface changes with the definition of the collective variables. From a single free energy surface, one cannot tell how a molecule transfers between the different metastable states and how long it takes. In order to get these information, Markov state model (MSM) has been proposed and discussed in many papers.[32–34] MSM is able to construct a reliable transition network for a biomolecule from many short simulations. From the network, one can obtain the stationary distributions of the important states and the dominant transition pathways between them. There are two well-known MSM packages in this field: pyEMMA[35] and MSMBuilder.[36]

In this article, we present a useful tool that can do both the conformational sampling work and the MSM analysis work. It is called fast sampling and analysis tool (FSATOOL for short). It must be noted that some existing software have been developed for this purpose in the past. For example, the well-known PLUMED[37] is a powerful library that is compatible with many molecular dynamics (MD) software. It integrates a lot of sampling and analyzing methods. However, up to now, it lacks the GPU-accelerated ability in AMBER, which is critical for the simulation of large systems. The MSM analysis module is not included in the library too. Another well-known platform is ACEMD.[38] It is an MD engine that provides a scripting Python interface of high-throughput molecular dynamics.[39] And importantly, it has a MSM module to construct the transition network on the fly. The network can be further used to accelerate the sampling in the simulation.[40]

As an assistant tool, our FSATOOL has the following features. First, FSATOOL implements the mixing REMD sampling method. As discussed in our previous paper,[29] the method accelerates the simulation by the artificial potential and the high temperature simultaneously. The former is for the fast sampling in the collective variable space and the latter is for the global searching in the conformational space. No complicated reweighting step is required in the free energy calculation on different collective variables.

Second, FSATOOL is integrated into AMBER.[2] It provides several enhanced sampling methods. Besides the mixing REMD mentioned above, it also supports SMD,[9] essential dynamics sampling (EDS),[41] adaptively biased molecular dynamics (ABMD)[42] and standard REMD.[6] All of them can be applied to the simulations on GPU.

Finally, FSATOOL can analyze the sampling data by the MSM method.[32–34] In AMBER, CPPTRAJ[43] is good at many tasks like clustering, hydrogen bond analysis, root-mean-square deviation calculation, energy calculation, and so on. But, it does not have the ability to do a MSM-based analysis. Our FSATOOL is a supplementary to CPPTRAJ. It provides an automated MSM analysis module with minimal intervention from users. The application process is highly simplified.

In this article, we introduce the framework and the functionality of FSATOOL. As an illustration, we use the tool to analyze the folding pathway of a short RNA hairpin. The results are comparable to recent papers.[31,44,45]

## Materials and Methods

The framework of FSATOOL is presented in Figure 1. It consists of two main components: sampling module and MSM module. Both of them need input files. Input file 1 in the sampling module provides the parameters for different sampling methods. Currently there are five sampling methods implemented in FSATOOL: mixing REMD,[29] SMD,[9] essential dynamics sampling (EDS), [41] ABMD,[42] and standard REMD.[6] For example, mixing REMD[29] requires a set of collective variables to be included in the biasing potential. When the simulation is done, it provides the unbiased trajectories of Cartesian coordinates or collective variables. All the collective variables are calculated by the non-equilibrium free energy (NFE) module in AMBER.[2] If necessary, one can also use some other tools to get the special collective variables based on the output trajectories, like eRMSD in Barnaba.[46] After sampling, the output trajectories or the extracted collective variables are handled by the MSM module in FSATOOL. Input file 2 contains its required parameters, including the lag time, the number of the clusters and the coarse-grained macrostates. Based on MSM[32,33,47,48] and transition path theory,[49,50] the MSM module constructs a state-to-state transition network and generates some transition paths for the molecule. At last, some representative snapshots of the macrostates on each path are extracted from the trajectories of the Cartesian coordinates. From these snapshots, one can have all the details on a particular transition process.

At present, FSATOOL is only a sampling and analysis tool, not an independent simulation software. To use FSATOOL, it has to be compiled with AMBER software first.[2] In the simulation performed by pmemd.cuda of AMBER, the sampling module of FSATOOL is called at every MD step. According to the name list

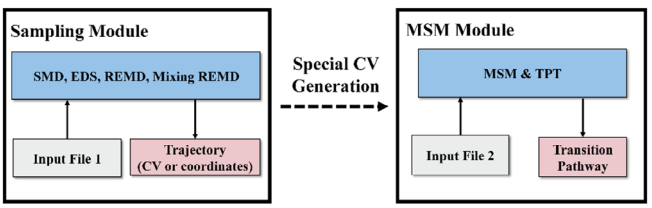


**Figure 1.** The framework of fast sampling and analysis tool. It has two main modules: sampling module and Markov state model module. The former does the conformational sampling work in the simulation and the latter does the trajectory analysis work after the simulation. Each of them needs a particular input file for the required parameters. [Color figure can be viewed at wileyonlinelibrary.com]

"task" in the input file, different sampling methods can be applied to the simulation, including the mixing REMD method that has all the advantages with the biased and unbiased sampling strategies.[29] The method is introduced and tested well in our previous paper.[29] In this work, we write an interface of the method to the sampling module of FSATOOL. The main idea is given in Figure 2.

Here, we give a brief description of the mixing REMD method.[29] There are three different kinds of replicas in the simulation: biased replicas, equilibrium replicas, and REMD replicas. Biased replicas are simulated with an adaptive biasing potential. They sample fast in a predefined collective variable space. At a fixed time step $\tau_1$, the states of the biased replicas are transferred to the equilibrium replicas directly for equilibrium simulation. Then at another time step $\tau_2$, they are further transferred to REMD replicas according to the Metropolis criterion eq. (1).[30]

$$P = \min\{1, \exp(-(V(\mathbf{r}') - V(\mathbf{r}))/k_B T)\} \quad (1)$$

Here $P$ is the probability for transferring the configuration from an equilibrium replica to a REMD replica, and $V(\mathbf{r}')$, $V(\mathbf{r})$ are their potential energies, respectively. The timescale $\tau_1$ is longer than the timescale $\tau_2$ to allow the structure of the equilibrium replica to be sufficiently relaxed before the transferring operation. REMD replicas are placed at different temperatures. They try to exchange with each other according to the eq. (2).[6]

$$P_{ij} = \min\left\{1, \frac{\exp\left(-V\left(\mathrm{r}_j^{(i)}\right)/k_B T^{(i)}\right)\cdot\exp\left(-V\left(\mathrm{r}_i^{(j)}\right)/k_B T^{(j)}\right)}{\exp\left(-V\left(\mathrm{r}_i^{(i)}\right)/k_B T^{(i)}\right)\cdot\exp\left(-V\left(\mathrm{r}_j^{(j)}\right)/k_B T^{(j)}\right)}\right\} \quad (2)$$

Here $P_{ij}$ is the exchange probability from state $i$ to state $j$, $V(\mathbf{r}i^{(j)})$ is the potential energy of the $i$th replica at $j$th temperature $T^{(j)}$.

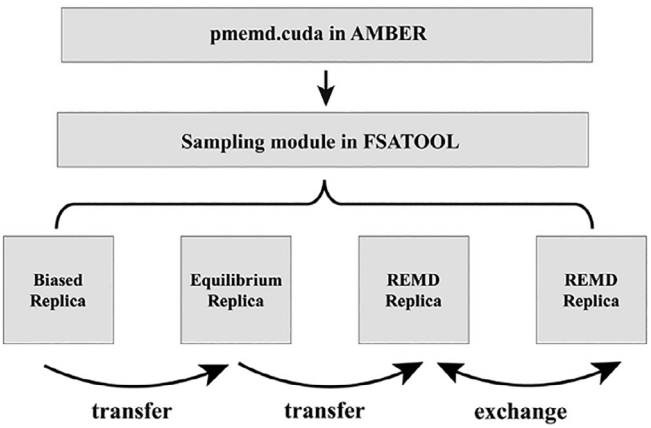


**Figure 2.** Diagram of the mixing replica exchange molecular dynamics (REMD) method in the sampling module of fast sampling and analysis tool. It is constituted of different replicas, including the biased replica for biased simulation, equilibrium replica for equilibrium MD simulation, and REMD replica for unbiased simulation. Each replica is run by an independent executable pmemd.cuda in AMBER. The simulation data transfer between the replicas periodically in the simulation. See the details of the method in our previous paper.[29]

In the simulation, REMD replicas are simulated in the canonical ensembles. They produce the unbiased sampling data, which are easy to use in the calculation of the thermodynamics quantities on any collective variables. As to the biased replica, the adaptive biasing potential is constructed by ABMD,[42] which is a simple, fast, and efficient method widely used for the sampling and the free energy calculation of biomolecules.[17,42,51] In the future, more biasing methods will be used in mixing REMD, like metadynamics[7,8] with an adaptive biasing potential and H-REMD[14–17] with a well-defined and static biasing potential. These methods can greatly improve the sampling efficiency in the simulation. Moreover, other potential-free biasing methods, like EDS,[41] can also be combined to the biased replica. It makes the molecule sample along the largest amplitude of the collective fluctuations. It should be noted that these biased simulations generate the biased sampling data on the predefined collective variable. Generally, reweighting the data from one collective variable to another is difficult. But in mixing REMD, this reweighting step is not necessary at all.

As discussed in our previous paper,[29] mixing REMD is an extension of the existing methods. But it does have some advantages over them. First, unlike H-REMD,[14–17] HT-REMD,[18–20] mixing REMD does not need a reference biasing potential before the formal simulation. It is easy to start. Second, the adaptive potential of the biased replicas evolves all the time. All the minima and barriers on free energy landscape are flattened out gradually in the simulation. The sampling is efficient. Finally, mixing REMD completely separates the biased replicas from the unbiased replicas. It does not need a lot of replicas to ensure a high exchange rate between them. This is helpful for the simulations with limited computer resources.

When the simulation by the sampling module is finished, its output data (trajectories or the extracted collective variables) are processed by the MSM module in FSATOOL. As we mentioned before, MSM module tries to build a state-to-state transition network and the optimal transition pathways from the sampling data. According to the MSM theory and the well-known pyEMMA[35] and MSMBuilder[36] software, we implement the MSM module as the following steps (see the framework in Fig. 3).

1. **Dimension reduction**: Original trajectories from the sampling simulation are a set of high dimensional data. They are not intuitive. For the sake of convenience, sometimes it is better to reduce the dimensionality of the data by principal component analysis (PCA)[52] or the TICA.[27,53,54] Both of the two methods are proposed for the determination of the representative degrees of freedom according to the distribution of the sampling data in the Cartesian coordinate space or the collective variable space. PCA[52] is good at finding the most independent components and TICA[27,53,54] is good at finding the slowest modes.
2. **Clustering**: This step is to divide the sampling data (snapshots) into several clusters. All the snapshots that belong to the same cluster have the similar conformations. In the past, a lot of famous clustering methods have been developed to do this, such as k-means,[55] k-medoids, [56] and k-centers.[57]

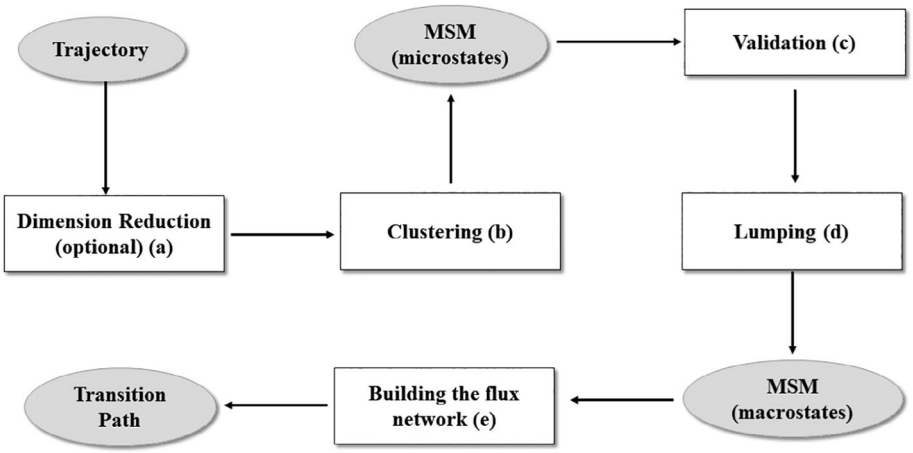


**Figure 3.** The framework of Markov state model module in fast sampling and analysis tool. Rectangular boxes represent the different calculation steps and the elliptic boxes represent the generated data.

The former two methods are included in our MSM module. After the clustering step, each cluster is viewed as a microstate in MSM.

3. **Building and validating the microstates**: At this step, a transition count matrix (TCM) is built first by counting the number of transitions between the clusters at a given lag time. To satisfy the detailed balance condition, the matrix is further symmetrized by the maximum likelihood estimator[47,48] to produce a transition probability matrix (TPM). Now, we have to check if the TPM is Markovian at the particular lag time. This can be validated in two ways. One way is calculating the implied timescale. The current lag time is only proper when the implied timescale begins to level off along the axis of lag time. The other way is to use Chapman–Kolmogorov equation.[48] MSM is Markovian only when the predicted distribution probabilities of the microstates at a particular lag time $\tau$ are close to those at the lag time $m\tau(m > 1)$. Moreover, TPM can also be built by the transition-based reweighting analysis method.[58] It is able to combine the simulation data from multiple ensemble in a multiensemble Markov model.[58]
4. **Lumping**: In general, the number of the microstates is too large for the description of the transition process of the biomolecules in the simulation. They need to be coarse-grained into macrostates. There are many methods to do this. Most of them are based on the fact that the interstate transitions between the metastable states are much slower than the intrastate transitions. For example, Perron cluster cluster analysis (PCCA)[59] and Robust Perron cluster analysis (PCCA+)[60,61] perform the spectral analysis on the TPM. And Bayesian agglomerative clustering engine[62] compares the likelihood of the states come from the same macrosates to the likelihood of the states come from different macrostates.
5. **Building the flux network**: After the lumping step, what we want to know is the dominant transition pathways between any two macrostates. For convenience, we define state A as the initial state and B as the final state. The analysis uses an important variable: committor probability (or $P_{fold}$).[33,49,50] The committor probability $q_i^+$, represents the probability of a molecule being at state $i$ goes to state B first instead of state $A$. And the backward committor probability $q_i^-$ represents the probability of a molecule being at state $i$ comes from state A previously instead of B.

   Following ref. [33], we build the effective flux network according to eq. (3). In the equation, $T_{ij}$ is the transition probability in TPM from state $i$ to $j$, $f_{ij}$ is the effective flux along the edge $i$–$j$ contributing to the transition process from A to B.[33]

$$f_{ij} = \pi_i q_i^- T_{ij} q_j^+ . \quad (3)$$

Since $f_{ij}$ may contains the detoured fluxes, it is preferred to use the net flux[33]

$$f_{ij}^+ = \max\left(0, f_{ij} - f_{ji}\right). \quad (4)$$

From the net flux network, a couple of individual pathways can be determined. The strongest pathway is extracted first and then removed from the network. The next path is the second strongest one. The whole process goes on repeatedly until there is no path available in the flux network from A and B.[33]

In summary, we provide a simulation tool FSATOOL in this article. It contains a conformational sampling module and a MSM analysis module. The two modules work closely with each other. The program is written by Fortran language and it is compatible with AMBER. More introduction of the tool is given on the website: https://fsatool.github.io. In the next section, we illustrate the usage of FSATOOL by an RNA hairpin.

## Results and Discussions

RNA hairpin tetraloop contains four nucleotides (L1–L4) in the loop that play an import role in RNA structure and function.[63] In recent years, it has been used as a test model by many enhanced sampling methods.[64–66] Now, we give an example of how to simulate the folding of an RNA hairpin by FSATOOL.

The hairpin structure comes from the X-ray structure[67] (protein data bank code (PDB id) 1F7Y, residues 6–15). It consists of 10 nucleotides with the sequence GCC(UUCG)GGC. From now on, its nucleotides UUCG in the tetraloop region are represented by $U_{L1}$, $U_{L2}$, $C_{L3}$, $G_{L4}$, respectively. As shown in Figure 4, the hairpin structure can be divided into two distinct regions: the stem region and the tetraloop region. The stem region has three typical G–C base pair interactions. The tetraloop has a $G_{L4}$–$U_{L1}$ trans-Watson–Crick/Sugar-Edge base pair interaction and a $U_{L1}$–$C_{L3}$ stacking interaction.[45,71] In addition, the $C_{L3}$ amino group and $U_{L2}$ phosphate form a 7BPh interaction according to ref. [69]. These complicated interactions between the nucleotides make the sampling difficult.[31,44,45]

The way to do the simulation is the same as pmemd.cuda in AMBER. Three files must be prepared at the beginning, including the topology file, the restart file, and the parameter file. Some important name lists are provided in the parameter file. The most important one is "cntrl," which gives the detailed conditions in a standard AMBER simulation like the time step, the temperature and the number of the simulation steps. Another name list is "colvar." It defines a set of collective variables that are calculated by the NFE module in AMBER.[2] Besides of them, we add a new name list "task." It belongs to the sampling module of FSATOOL, which determines the particular sampling method in the simulation. For example, "ifsmd" represents the SMD simulation, "iffood" represents the standard ABMD simulation or the mixing REMD simulation. Their required parameters are given in the name list "smd" and "flood," respectively.

In our work, we perform a simulation for the RNA hairpin from a fully extended structure by the mixing REMD method.[29] The system is emerged in a periodic box of 64 Å × 58 Å × 45 Å with TIP3P waters.[72] Nine $Na^+$ counterions are added to the box to neutralize the system. The force field is ff99bsc0 + $\chi$OL3 RNA force field[73,74] and the time step is 2.0 fs. SHAKE algorithm[75] is used to constrain the bonds involving the hydrogens. There are four replicas in the simulation: one biased replica at 305 K, one equilibrium replica 305 K, and two REMD replicas at 300 K and 305 K, respectively. The biased replica is simulated independently with an adaptive biasing potential as in a standard ABMD simulation.[42] It copies its state to the equilibrium replica every 500,000 steps, and the transferring steps from the equilibrium replica to the hottest REMD replica is 50,000. The simulation time for each replica is 4 μs. The whole simulation time of all the replicas is 16 μs.

We choose two collective variables for the biased replica: the number of the hydrogen bonds in the native structure and the RMSD of C3′ atoms to the native structure. The choice of the collective variables is considered from two aspects. First, RMSD and the native H-bonds are the proper collective variables to describe the native state of the RNA tetraloop. On the free energy landscape spanned by these collective variables, the native state with a small RMSD and a large native H-bond does not overlap with the other metastable states. Second, simulation of the folding process of RNA is time-consuming. We use the two collective variables to improve the sampling on the paths leading to the native state. This is important in the study of the folding mechanism of the molecule. Actually, RMSD and other native structural information have already been used in the folding of RNA in previous papers.[31,45,76–78]

After the simulation, the free energy surface of the unbiased REMD replica at 300 K is plotted in Figure 5. It presents that the molecule can finally fold into its native state (bottom right corner). But there are too many mixed metastable states on the free energy surface. The data are not suitable for MSM analysis. In this case, as a general rule one has to change the collective variables and reperform the whole simulation, or, reweight the output data to some new collective variables by a complicated step. Both of the two tasks are unnecessary to mixing REMD. It produces the unbiased sampling data for the free energy calculation on any collective variables.

Recently, eRMSD[79] and C1′–C1′ end-to-end distance have been shown to be suitable in the representation of the free energy surface of UUCG tetraloop.[31] So, we change the

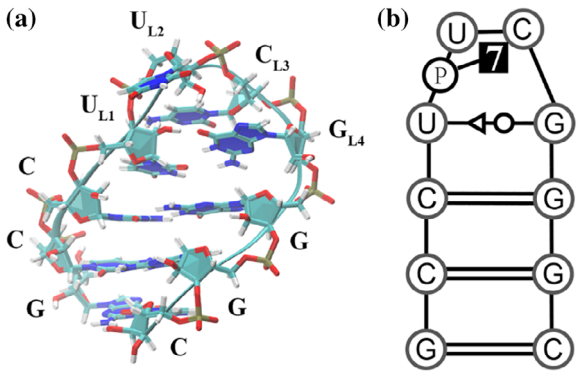


**Figure 4.** a) The native structure of the RNA hairpin in our work (drawn by visual molecular dynamics (VMD)[68]). b) Illustration of the critical interactions between the nucleotides. The symbols of the base-pairing and the 7BPh interactions in the tetraloop region are defined by Leontis–Westhof–Zirbel nomenclature.[69,70] [Color figure can be viewed at wileyonlinelibrary.com]

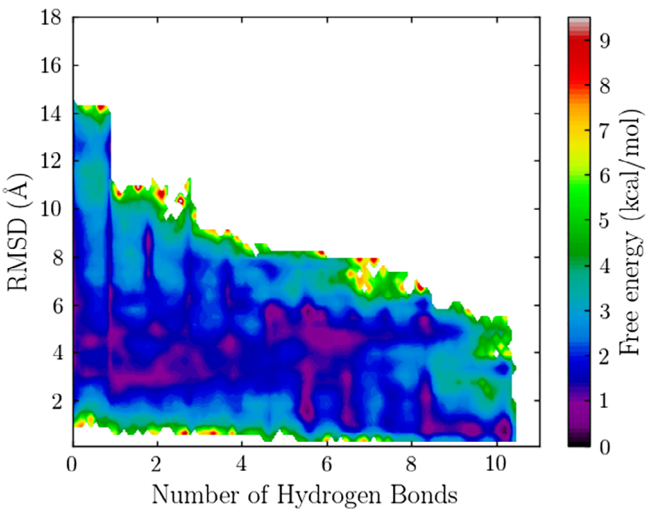


**Figure 5.** Free energy surface of the RNA hairpin in the mixing replica exchange molecular dynamics simulation by fast sampling and analysis tool. The two collective variables are the number of the hydrogen bonds in the native structure and the RMSD of C3′ atoms to the native structure. [Color figure can be viewed at wileyonlinelibrary.com]

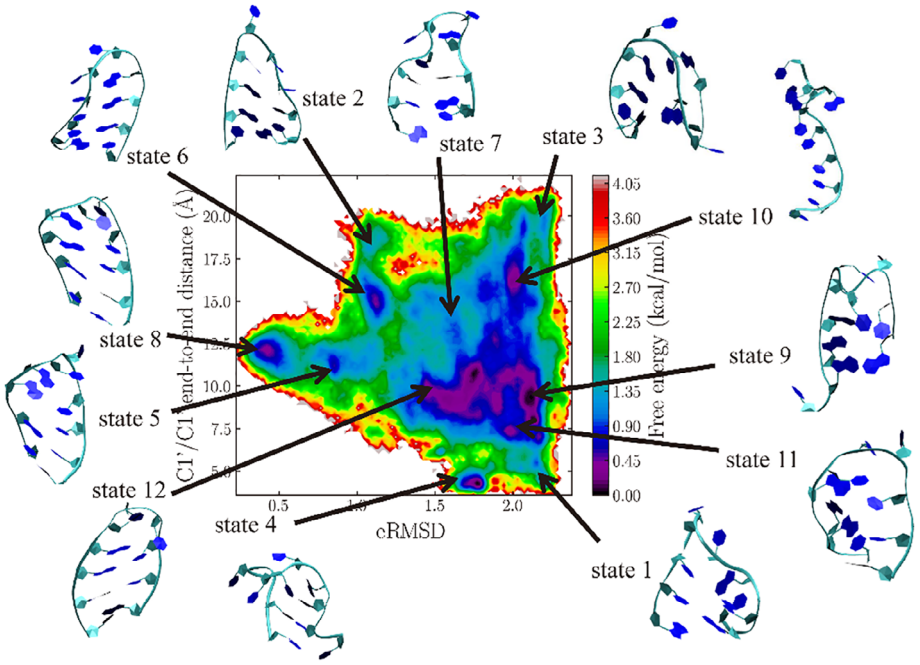


**Figure 6.** Free energy surface of the RNA hairpin in the mixing replica exchange molecular dynamics simulation by fast sampling and analysis tool. The two collective variables are the eRMSD to the native structure (calculated by Barnaba software[46]) and the C1′–C1′ end-to-end distance. The snapshots of the Markov state model macrostates are also presented in the figure (drawn by VMD[68]). [Color figure can be viewed at wileyonlinelibrary.com]

collective variables and recalculate a new free energy surface on them based on the same simulation data. Here, the eRMSD to the native state for each snapshot is calculated by Barnaba software.[46] The surface is given in Figure 6. Compared to Figure 5, this surface is more informative. There are several separated stable states on the surface, including the native state with eRMSD <0.6 Å.

Figure 7 shows the change of eRMSD in the mixing REMD simulation at 300 K (black solid lines). The data come from two unbiased REMD replicas. It is found that the eRMSD oscillates between a higher and a lower value all the time. The lower value (eRMSD < 0.6 Å) corresponds to the native state and the higher one (eRMSD > 1.5 Å) is the unfolded state. This oscillation reveals a slow and repeated folding-unfolding process during the simulation. Moreover, eRMSD increases and decreases with the backbone $\chi$ angle of $G_{L4}$ (red dashed lines in Fig. 7). It indicates a strong correlation between these two variables. As to the backbone $\chi$ angle of $G_{L4}$, it is often used to define two typical conformations: *syn* and *anti* conformation. Both of them are shown in Figure 8. In the *anti* conformation, $U_{L1}$ and $G_{L4}$ form a wobble pair. And in the *syn* conformation, $U_{L1}$ and $G_{L4}$ form a trans Watson–Crick/Sugar-Edge base-pair. The coordinated variation of eRMSD and the backbone $\chi$ angle of $G_{L4}$ indicates that this particular nucleotide goes from the *anti* conformation to the *syn* conformation during the folding process of the RNA hairpin.

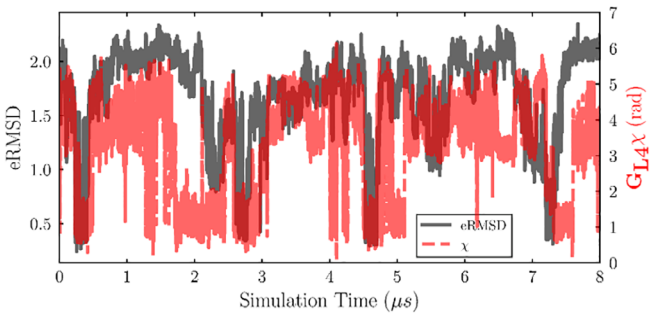


**Figure 7.** The changes of eRMSD (black solid line) and backbone $\chi$ angle of $G_{L4}$ (red dashed line) in the mixing replica exchange molecular dynamics (REMD) simulation at 300 K. The data come from two unbiased REMD replicas. [Color figure can be viewed at wileyonlinelibrary.com]

Besides eRMSD and the backbone angle, the sugar pucker phase is also an important variable for RNA.[80] In Figure 9, we plot the changes of sugar pucker of the four nucleotides in the loop, $U_{L1}$, $U_{L2}$, $C_{L3}$, and $G_{L4}$. It presents that different nucleotides have different sampling behaviors. $U_{L2}$ and $C_{L3}$ prefer to stay in the stable C2′-endo state. And $G_{L4}$ likes the C3′-endo state instead. These results indicate that the OL3 force field has a well performance on the sampling of the sugar pucker phases.[74,81] Compared to $U_{L2}$ and $C_{L3}$, $U_{L1}$ and $G_{L4}$ are more flexible due to the multiple interactions between them. $U_{L1}$ transforms quickly between the C2′-endo state and the C3′-endo state in the whole simulation (Fig. 9a). And $G_{L4}$ transforms among three states, C3′-endo, C2′-endo, and C2′-exo. This reveals a significant conformational dynamics of the nucleotide (Fig. 9d).[82]

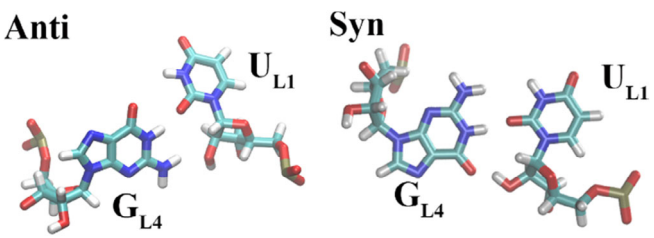


**Figure 8.** Schematic illustration of the *anti* (left) and the *syn* (right) conformation of $G_{L4}$. The figure is produced by VMD.[68] [Color figure can be viewed at wileyonlinelibrary.com]

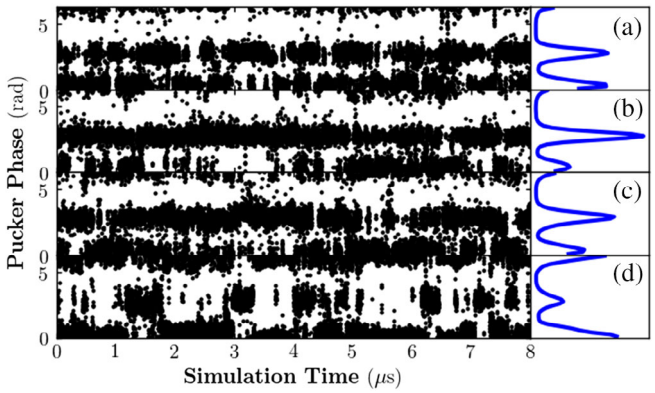


**Figure 9.** The changes of the sugar pucker of $U_{L1}$ (a), $U_{L2}$ (b), $C_{L3}$ (c), and $G_{L4}$ (d) in the mixing replica exchange molecular dynamics simulation at 300 K (two replicas). The blue curves at the right side are the corresponding probability distributions of the sugar pucker of the four nucleotides. [Color figure can be viewed at wileyonlinelibrary.com]

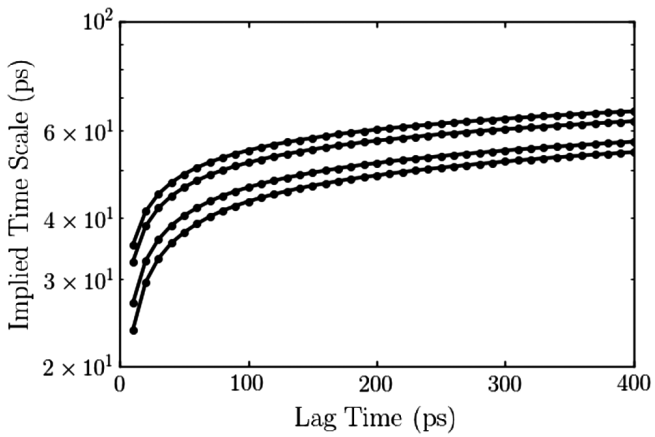


**Figure 10.** Top four implied timescales in the simulation of the RNA hairpin as a function of the lag time. The data come from two unbiased replica exchange molecular dynamics replicas at 300 K.

Now we begin to analyze the folding of the RNA hairpin by the MSM module in FSATOOL. First, all the snapshots at 300 K are clustered into 800 microstates in the two dimensional space of eRMSD and the C1′–C1′ end-to-end distance. The clustering method is kmeans++.[55,83] The changes of the top four implied timescales as a function of lag time are shown in Figure 10. The timescales begin to level off at 200 ps. So, we build the transition count matrix and the TPM with a lag time of 200 ps. From the matrix, the microstates are further lumped into 12 macrostates by PCCA+ method.[60,61]

The representative conformations of all the MSM macrostates are provided in Figure 6. Their positions on the free energy surface are also marked in the figure. It shows that state 10 has an extended structure and state 8 has the lowest eRMSD. So, they are set as the initial and final state in the analysis of the transition path, respectively. Then a flux network is constructed by the MSM module of FSATOOL between the two states. The individual pathways extracted from the network is given in Figure 11. The most dominant pathway follows the sequence: state 10 → state 12 → state 5 → state 8 and the second pathway follows: state10 → state 7 → state 6 → state 5 → state 8. These two pathways occupy 53% and 15% of the total flux, respectively:

Next, in order to get a clear knowledge of the folding process of the RNA hairpin, we pick five critical macrostates on the first and second dominant folding pathways. They are state 5, 6, 7, 8, and 12. All the snapshots in the central clusters of the five macrostates are extracted for the calculation of the secondary structures. The calculation is performed by the Barnaba software[46] according to the Leontis–Westhof nomenclature.[70] The final calculated structures are shown in Figure 12. At the right side of each secondary structure, it is the probability distribution of the $\chi$ angle of $G_{L4}$ nucleotide in the central cluster of the corresponding macrostate. In the figure, the upper pathway is the most dominant one in the folding of the RNA hairpin. On the path, $U_{L1}$ and $G_{L4}$ form a wobble pair firstly and $G_{L4}$ stays in the *anti* conformation (state 12). Then $U_{L1}$ and $G_{L4}$ form a trans-Watson–Crick/Sugar Edge base pair and $G_{L4}$ transforms to the *syn* conformation (state 5). Finally, all the G–C base pairs form correctly in the stem region (state 8). In comparison to the first path, the second dominant path at the bottom only shows the difference at the early stage of the folding. $U_{L1}$ and $G_{L4}$ form a

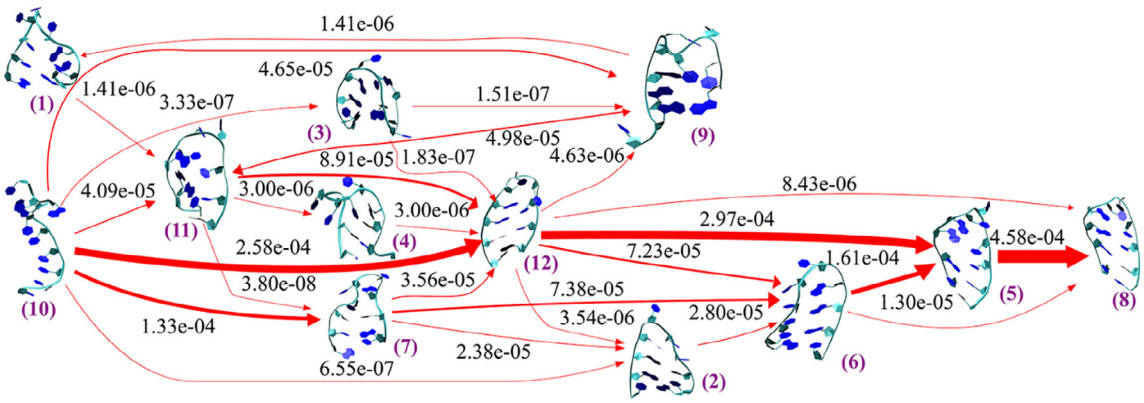


**Figure 11.** The dominant pathways in the folding of the RNA hairpin from an extended state to the native state. The data are calculated by the Markov state model module in fast sampling and analysis tool. The magnitudes of the fluxes between the states are indicated by the thickness of the red arrows. Their real values are also provided in the figure. The indices of all the states are given in the parentheses. All the representative conformations of the macrostates are produced by VMD.[68] [Color figure can be viewed at wileyonlinelibrary.com]

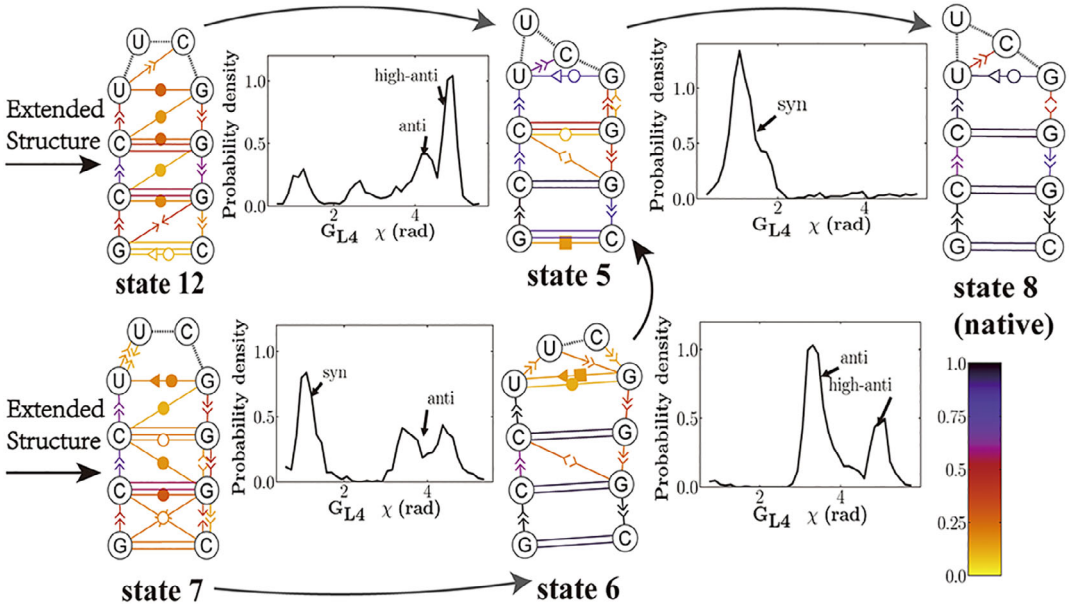


**Figure 12.** Two dominant folding pathways of the RNA hairpin. Each marcostate is represented by a secondary structure in its central cluster. The occupancies of the formed interactions in the structures are drawn by different colors in the colorbar at the bottom-right corner. The subplot near the macrostate's secondary structure is the probability distribution of the $\chi$ angle of $G_{L4}$ nucleotide in the corresponding central cluster. The data and the figure are generated by Barnaba software.[46] [Color figure can be viewed at wileyonlinelibrary.com]

cis-Watson–Crick/Sugar-edge base pair first and $G_{L4}$ more prefers the *syn* conformation (state 7). Subsequently, $G_{L4}$ transforms to *anti* conformation (state 6) and then back to *syn* conformation (state 5). Upon the two pathways, the folding process clearly depends on the *anti* to *syn* conformation transformation of $G_{L4}$. And it also corresponds to the transition from the wobble pair to the trans-Watson–Crick/Sugar-Edge base pair between $U_{L1}$ and $G_{L4}$ in the secondary structure. This result is in agreement with the recent findings that the *anti*-to-*syn* transition corresponds to the folding process of the UUCG tetraloop.[44,45]

## Conclusions

As now it is still difficult to reach the timescales of many important conformation changes of a biomolecule in a normal MD simulation. Generally, this timescale ranges from milliseconds to seconds. To circumvent the problem, one has to rely on the enhanced sampling method or the MSM analysis method. The former makes the sampling more efficient and the latter is able to extract the long-time kinetics from the short simulations.

In this article, we develop a FSATOOL package to do both things. It uses the sampling module to perform an accelerated simulation with an adaptive biasing potential and the MSM module to analyze the transition network and pathway. It combines REMD[6] with the ABMD.[42] Biased simulation by ABMD accelerates the sampling in the collective variable space and unbiased REMD simulation is superior for the fast sampling on all the degrees of freedom. In the MSM module of FSATOOL, all the unbiased sampling data are clustered to build a transition network. From the network, some dominant transition pathways can be extracted quickly.

The usage of FSATOOL is quite easy. Besides the standard input files for AMBER,[2] only two additional input files are required. One is for the sampling process and the other is for the MSM analysis. We hope the tool be helpful for the researchers who are interested in the long-time statistical dynamics of biomolecules.

## Acknowledgment

*This work is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 31770773.*

**Keywords:** RNA hairpin · free energy calculation · Markov state model · transition path

How to cite this article: H. Zhang, Q. Gong, H. Zhang, C. Chen. *J. Comput. Chem*. **2019**, 9999, 1–9. DOI: 10.1002/jcc.26083

[1] T. J. Lane, D. Shukla, K. A. Beauchamp, V. S. Pande, *Curr. Opin. Struct. Biol.* **2013**, *23*, 58.

[2] D. A. Case, I. Y. B, S. R. Brozell, D. S. Cerutti, T. E. Cheatham, III., V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, D. Ghoreishi, M. K. Gilson, H. Gohlke, A. W. Goetz, D. Greene, R. Harris, N. Homeyer, S. Izadi, A. Kovalenko, T. Kurtzman, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D. J. Mermelstein, K. M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C. L. Simmerling, J. Smith, R. Salomon-Ferrer, J. Swails, R. C. Walker, J. Wang, H. Wei, R. M. Wolf, X. Wu, L. Xiao, D. M. York, P. A. Kollman, AMBER 2018, University of California, San Francisco, CA, **2018**, p. 2018.

[3] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L. P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2013**, *9*, 461.

[4] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, E. Lindahl, *SoftwareX* **2015**, *1–2*, 19.
[5] T.-S. Lee, D. S. Cerutti, D. Mermelstein, C. Lin, S. LeGrand, T. J. Giese, A. Roitberg, D. A. Case, R. C. Walker, D. M. York, *J. Chem. Inf. Model.* **2018**, *58*, 2043.
[6] Y. Sugita, Y. Okamoto, *Chem. Phys. Lett.* **1999**, *314*, 141.
[7] A. Barducci, G. Bussi, M. Parrinello, *Phys. Rev. Lett.* **2008**, *100*, 020603.
[8] A. Barducci, M. Bonomi, M. Parrinello, *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2011**, *1*, 826.
[9] B. Isralewitz, M. Gao, K. Schulten, *Curr. Opin. Struct. Biol.* **2001**, *11*, 224.
[10] G. Diaz Leines, B. Ensing, *Phys. Rev. Lett.* **2012**, *109*, 020601.
[11] M. I. Zimmerman, G. R. Bowman, *J. Chem. Theory Comput.* **2015**, *11*, 5747.
[12] R. Affentranger, I. Tavernelli, E. E. Di Iorio, *J. Chem. Theory Comput.* **2006**, *2*, 217.
[13] Y. Sugita, A. Kitao, Y. Okamoto, *J. Chem. Phys.* **2000**, *113*, 6042.
[14] H. Fukunishi, O. Watanabe, S. Takada, *J. Chem. Phys.* **2002**, *116*, 9058.
[15] J. Hritz, C. Oostenbrink, *J. Chem. Phys.* **2008**, *128*, 144121.
[16] S. Kannan, M. Zacharias, *Proteins* **2007**, *66*, 697.
[17] V. Babin, C. J. Sagui, *Chem. Phys.* **2010**, *132*, 03B602.
[18] R. Laghaei, N. Mousseau, G. Wei, *J. Phys. Chem. B* **2010**, *114*, 7071.
[19] R. Laghaei, N. Mousseau, G. Wei, *J. Phys. Chem. B* **2011**, *115*, 3146.
[20] M. Deighan, J. Pfaendtner, *Langmuir* **2013**, *29*, 7999.
[21] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, P. A. Kollman, *J. Comput. Chem.* **1992**, *13*, 1011.
[22] A. Laio, M. Parrinello, *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 12562.
[23] J. Preto, C. Clementi, *PCCP* **2014**, *16*, 19181.
[24] M. M. Sultan, V. S. Pande, *J. Chem. Theory Comput.* **2017**, *13*, 2440.
[25] A. Mardt, L. Pasquali, H. Wu, F. Noé, *Nat. Commun.* **2018**, *9*, 5.
[26] F. Noe, C. Clementi, *Curr. Opin. Struct. Biol.* **2017**, *43*, 141.
[27] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, F. Noé, *J. Chem. Phys.* **2013**, *139*, 015102.
[28] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, F. Noé, *J. Chem. Theory Comput.* **2014**, *10*, 1739.
[29] H. Zhang, Q. Gong, H. Zhang, C. Chen, *J. Comput. Chem.* **2019**, *40*, 1806.
[30] W. K. Hastings, *Biometrika* **1970**, *57*, 97.
[31] S. Bottaro, P. Banas, J. Sponer, G. Bussi, *J. Phys. Chem. Lett.* **2016**, *7*, 4032.
[32] D. Shukla, C. X. Hernandez, J. K. Weber, V. S. Pande, *Acc. Chem. Res.* **2015**, *48*, 414.
[33] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, T. R. Weikl, *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 19011.
[34] J. D. Chodera, F. Noe, *Curr. Opin. Struct. Biol.* **2014**, *25*, 135.
[35] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, F. Noé, *J. Chem. Theory Comput.* **2015**, *11*, 5525.
[36] M. P. Harrigan, M. M. Sultan, C. X. Hernández, B. E. Husic, P. Eastman, C. R. Schwantes, K. A. Beauchamp, R. T. McGibbon, V. S. Pande, *Biophys. J.* **2017**, *112*, 10.
[37] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, G. Bussi, *Comput. Phys. Commun.* **2014**, *185*, 604.
[38] M. J. Harvey, G. Giupponi, G. D. Fabritiis, *J. Chem. Theory Comput.* **2009**, *5*, 1632.
[39] S. Doerr, M. Harvey, F. Noé, G. De Fabritiis, *J. Chem. Theory Comput.* **2016**, *12*, 1845.
[40] S. Doerr, G. De Fabritiis, *J. Chem. Theory Comput.* **2014**, *10*, 2064.
[41] A. Amadei, A. B. Linssen, H. J. Berendsen, *Proteins* **1993**, *17*, 412.
[42] V. Babin, C. Roland, C. Sagui, *J. Chem. Phys.* **2008**, *128*, 134101.
[43] D. R. Roe, T. E. Cheatham, III., *J. Chem. Theory Comput.* **2013**, *9*, 3084.
[44] P. Kührová, P. Banáš, R. B. Best, J. Í. Šponer, M. Otyepka, *J. Chem. Theory Comput.* **2013**, *9*, 2115.
[45] S. Haldar, P. Kührová, P. Banáš, V. c. Spiwok, J. í. Šponer, P. Hobza, M. Otyepka, *J. Chem. Theory Comput.* **2015**, *11*, 3866.
[46] S. Bottaro, G. Bussi, G. Pinamonti, S. Reisser, W. Boomsma, K. Lindorff-Larsen, *RNA* **2018**, *25*, 219.
[47] G. R. Bowman, K. A. Beauchamp, G. Boxer, V. S. Pande, *J. Chem. Phys.* **2009**, *131*, 124101.
[48] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, F. Noé, *J. Chem. Phys.* **2011**, *134*, 174105.
[49] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, E. S. Shakhnovich, *J. Chem. Phys.* **1998**, *108*, 334.
[50] E. Weinan, E. Vanden-Eijnden, *J. Stat. Phys.* **2006**, *123*, 503.
[51] M. Moradi, V. Babin, C. Roland, T. A. Darden, C. Sagui, *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 20746.
[52] S. Wold, K. Esbensen, P. Geladi, *Chemometrics Intellig. Lab. Syst.* **1987**, *2*, 37.
[53] L. Molgedey, H. G. Schuster, *Phys. Rev. Lett.* **1994**, *72*, 3634.
[54] C. R. Schwantes, V. S. Pande, *J. Chem. Theory. Comput.* **2015**, *11*, 600.
[55] J. A. Hartigan, M. A. Wong, *J. R. Stat. Soc. Ser. C Appl. Stat.* **1979**, *28*, 100.
[56] H. S. Park, C. H. Jun, *Expert Syst. Appl.* **2009**, *36*, 3336.
[57] T. F. Gonzalez, *Theor. Comput. Sci.* **1985**, *38*, 293.
[58] H. Wu, F. Paul, C. Wehmeyer, F. Noé, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, E3221.
[59] P. Deuflhard, W. Huisinga, A. Fischer, C. Schutte, *Linear Algebra Appl.* **2000**, *315*, 39.
[60] P. Deuflhard, M. Weber, *Linear Algebra Appl.* **2005**, *398*, 161.
[61] S. Röblitz, M. Weber, *Adv. Data Anal. Classif.* **2013**, *7*, 147.
[62] G. R. J. Bowman, *Chem. Phys.* **2012**, *137*, 134111.
[63] C. Woese, S. Winker, R. Gutell, *Proc. Natl. Acad. Sci. USA* **1990**, *87*, 8467.
[64] A. Gil-Ley, G. Bussi, *J. Chem. Theory Comput.* **2015**, *11*, 1077.
[65] N. M. Henriksen, D. R. Roe, T. E. Cheatham, III, *J. Phys. Chem. B* **2013**, *117*, 4014.
[66] D. R. Roe, C. Bergonzo, T. E. Cheatham, III, *J. Phys. Chem. B* **2014**, *118*, 3543.
[67] E. Ennifar, A. Nikulin, S. Tishchenko, A. Serganov, N. Nevskaya, M. Garber, B. Ehresmann, C. Ehresmann, S. Nikonov, P. Dumas, *J. Mol. Biol.* **2000**, *304*, 35.
[68] W. Humphrey, A. Dalke, K. Schulten, *J. Mol. Graph.* **1996**, *14*, 33.
[69] C. L. Zirbel, J. E. Šponer, J. Šponer, J. Stombaugh, N. B. Leontis, *Nucleic Acids Res.* **2009**, *37*, 4898.
[70] N. B. Leontis, E. Westhof, *RNA* **2001**, *7*, 499.
[71] J. Sponer, G. Bussi, M. Krepl, P. Banas, S. Bottaro, R. A. Cunha, A. Gil-Ley, G. Pinamonti, S. Poblete, P. Jurecka, N. G. Walter, M. Otyepka, *Chem. Rev.* **2018**, *118*, 4177.
[72] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, *J. Chem. Phys.* **1983**, *79*, 926.
[73] A. Perez, I. Marchan, D. Svozil, J. Sponer, T. E. Cheatham, III, C. A. Laughton, M. Orozco, *Biophys. J.* **2007**, *92*, 3817.
[74] M. Zgarbova, M. Otyepka, J. Sponer, A. Mladek, P. Banas, T. E. Cheatham, III, P. Jurecka, *J. Chem. Theory Comput.* **2011**, *7*, 2886.
[75] J.-P. Ryckaert, G. Ciccotti, H. J. Berendsen, *J. Comput. Phys.* **1977**, *23*, 327.
[76] P. Kührová, R. B. Best, S. Bottaro, G. Bussi, J. Šponer, M. Otyepka, P. Banáš, *J. Chem. Theory Comput.* **2016**, *12*, 4534.
[77] C. Yang, M. Kulkarni, M. Lim, Y. Pak, *Nucleic Acids Res.* **2017**, *45*, 12648.
[78] A. N. Borkar, P. Vallurupalli, C. Camilloni, L. E. Kay, M. Vendruscolo, *PCCP* **2017**, *19*, 2797.
[79] S. Bottaro, F. Palma, G. Bussi, *Nucleic Acids Res.* **2014**, *42*, 13306.
[80] J. S. Richardson, B. Schneider, L. W. Murray, G. J. Kapral, R. M. Immormino, J. J. Headd, D. C. Richardson, D. Ham, E. Hershkovits, L. D. Williams, *RNA* **2008**, *14*, 465.
[81] T. E. Cheatham, III, P. Cieplak, P. A. Kollman, *J. Biomol. Struct. Dyn.* **1999**, *16*, 845.
[82] J. Koplin, Y. Mu, C. Richter, H. Schwalbe, G. Stock, *Structure* **2005**, *13*, 1255.
[83] D. Arthur, S. Vassilvitskii, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, New Orleans, LA, **2007**, p. 1027.

Received: 1 August 2019
Revised: 10 September 2019
Accepted: 12 September 2019