PCCP

Accepted Manuscript

ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/pccp

Physical Chemistry Chemical Physics
CREATED USING THE RSC LaTeX PCCP ARTICLE TEMPLATE – SEE www.rsc.org/electronicfiles FOR DETAILS
ARTICLE TYPE www. rsc.org/xxxxxx | XXXXXXXX

Fast Recovery of Free Energy Landscapes via Diffusion-Map-directed Molecular Dynamics†

Jordane Preto,[a] and Cecilia Clementi[b]*

Received Xth XXXXXXXXXX 20XX, Accepted Xth XXXXXXXXX 20XX
First published on the web Xth XXXXXXXXXX 200X
DOI: 10.1039/b000000x

Reaction pathways characterizing macromolecular systems of biological interest are associated with high free energy barriers. Resorting to standard all-atom molecular dynamics (MD) to explore such critical regions may be inappropriate as the time needed to observe relevant transitions can be remarkably long. In this paper, we present a new method called Extended Diffusion-Map-directed Molecular Dynamics (extended DM-d-MD) used to enhance the sampling of MD trajectories in such a way as to rapidly cover all important regions of the free energy landscape including deep metastable states and critical transition paths. Moreover, extended DM-d-MD was combined with a reweighting scheme enabling to save *on-the-fly* information about Boltzmann distribution. Our algorithm was successfully applied to two systems, alanine dipeptide and alanine-12. Due to the enhanced sampling, Boltzmann distribution is recovered much faster than in plain MD simulations. For alanine dipeptide, we report a speedup of one order of magnitude with respect to plain MD simulations. For alanine-12, our algorithm allows to highlight all important unfolded basins in several days of computation when one single misfolded event is barely observable within the same amount of computational time by plain MD simulations. Our method is reaction coordinate free, shows little dependence on an *a priori* knowledge of the system, and can be implemented in such a way that the biased steps are not computationally expensive with respect to MD simulations thus making our approach well adapted for larger complex systems from which little information is known.

1 Introduction

For several decades now, understanding the dynamics underlying proteins function and reaction mechanisms has been a broad and fascinating topic. From the theoretical standpoint, the study of these systems implies the use of advanced computational techniques to account for the complexity of biomolecular systems. Among these techniques, all-atom molecular dynamics (MD) simulations have been widely used and provide an important complement to the experiments[1,2]. Nevertheless, MD has important limitations among which the sampling of complex high-dimensional configuration spaces. As the free energy barriers between different metastable minima of a given protein can be relatively high, the computational time needed by MD simulations to observe transitions of biological relevance may be extraordinarily long, should these transitions be eventually observed. Therefore, transition barriers are often poorly sampled thus giving very few details about the reaction paths and very rough estimates of the equilibrium distributions around these regions. To address these issues, several biased methods have been suggested over the past decades to rapidly find the transition barriers and generate a large number of samples along them (for a recent review of these methods, see Ref. 3). For instance, transition path sampling (TPS) relies on Monte Carlo procedures to build up the new samples from slight changes of a given trajectory along the transition path[4,5]. However, this implies that the initial trajectory is generated beforehand, thus requiring that initial and final states have been identified. Other recent TPS-like techniques such as transition interface sampling (TIS)[6] and forward flux sampling (FFS)[7,8] consist of defining interfaces along the transition regions so as to avoid the need to simulate the full reactive paths as well as to prevent multiple recrossings of the trajectories. For systems with multiple paths, TIS and FFS may have difficulty obtaining a good sampling of the various pathways. Further similar techniques were suggested recently to address this issue[9,10]. Another category of methods known as state-based methods such as string methods[11], Markov state models[12] or directional milestoning,[13] can also be used to generate a large number of samples along reaction paths (for a review of state-based methods, see Ref. 14). Most of these techniques require that we dispose of an initial set of configurations covering a wide area of the energy landscape. The initial configurations can be obtained, *e.g.*, from high-temperature simulations or biased methods based on the

† Electronic Supplementary Information (ESI) available: [details of any supplementary information available should be included here]. See DOI: 10.1039/b000000x/

[a] *Department of Chemistry, Rice University, Houston TX 77005, USA.*
[b] *Department of Chemistry, Rice University, Houston TX 77005, USA. Fax: +1-713-348-5155; Tel: +1-713-348-3485; E-mail: cecilia@rice.edu*

determination of appropriate reaction coordinates. It is worth mentioning that the definition of reaction coordinates remains a tricky issue [15].

A radically different approach to identify the reaction paths in macromolecular systems has been proposed based on the dynamics of the current density associated with molecular dynamics [16–20]. The method was recently referred to as transition current sampling [19]. While metastable basins are characterized by quasi-Boltzmann distributions and thus small currents, the regions constituting rare events - which contribute to reaction rates - are characterized by large steady currents when the reactants and products are out of equilibrium. In the same way as molecular dynamics can approximate the *probability density*, it is possible, by integrating appropriate Langevin equations, to generate stochastic trajectories which account for the *current density*, thus resulting in a large number of samples along the barriers. The above mentioned technique does not require any knowledge of appropriate reaction coordinates (indeed, current dynamics are integrated on the entire configuration space) which makes it well-adapted for a wide range of macromolecular systems. In particular, the method was used to explore the energy landscape of a short alanine peptide exhibiting a helix-coil transition [18]; in this case, the simulation time needed to populate the energy barrier of the system was found two orders of magnitude shorter than the characteristic time associated with normal MD. The same approach based on current density dynamics was also applied to the popular test-system of 38 atoms interacting through Lennard-Jones potential (LJ38). Again, the method allowed to rapidly explore the reactions that take place between the different phases that characterize the system [19].

Very recently, another reaction-coordinate-free algorithm designed to increase sampling along transition paths has been proposed, known as diffusion-map directed MD (DM-d-MD) [21]. The method is based on a previously developed dimensionality reduction technique, locally scaled diffusion map (LSDMap) [15,22–24], which extracts, directly from MD, a set of collective coordinates, referred to as diffusion coordinates (DCs), that decorrelate the motion of a macromolecular system over different time scales. By periodically restarting the dynamics from the furthest point along the slowest time scale (1^{st} DC), it becomes easier to visit unexplored regions of the configuration space instead of being trapped in local minima. Notably, DM-d-MD algorithm was applied to two systems, alanine dipeptide and alanine-12 and was reported in both cases to find the transition barriers about three orders of magnitude faster than normal MD.

It should be stressed that resorting to biased methods such as DM-d-MD or transition current sampling implies that most of the information about the probability distribution associated with molecular dynamics is lost as a result of the bias introduced in the sampling. Additional techniques like umbrella sampling [25] or state-based models need to be applied to recover Boltzmann distribution from the biased trajectories [21]. However, as associated algorithms remain computationally expensive with respect to MD, the post-processing of the sampling may significantly hinder the efficiency of the overall approach. In the present paper, we present a new reaction-coordinate-free method that allows, from multiple MD trajectories, not only to rapidly enhance and stabilize the sampling around high-energy barriers but also to recover *on-the-fly* information on Boltzmann distribution from the biased data. As a consequence of the large number of sampled points, especially within the transition regions, equilibrium distribution is achieved much faster than normal MD simulations. In Section 2, we give the details of our algorithm: we use an extended version of the DM-d-MD technique mentioned above (extended DM-d-MD) which consists in periodically restarting MD trajectories from a set of chosen data points instead of restarting them from the same point like in the original DM-d-MD [21]; the new starting points are picked uniformly along the two slowest collective time scales of the dynamics, which are directly given by the two first DCs of the data points. In this way, the new MD trajectories are more likely to explore a wide area of the free energy landscape within a given time interval. As far as Boltzmann distribution is concerned, it is recovered by assigning a statistical weight to each trajectory. Similarly to the case of a single MD trajectory, the correct equilibrium distribution is obtained from ergodic property by merging several distributions of weights obtained at regular time intervals. When dealing with multiple MD trajectories, the focus is not on the simulation of the trajectories *per se* but rather on the evolution of the probability density. Therefore, it is possible to ”spawn” or ”kill” particular trajectories as long as we ensure that the global population evolution is conserved. Each time data points are selected as the starting points of new MD trajectories, an appropriate reweighting scheme based on nearest neighbor search is used to locally keep of the information about Boltzmann distribution. Similarly to the original DM-d-MD, our algorithm is applied to a well-studied test system, alanine dipeptide, and to alanine-12, characterized by a much more complex energy landscape. Main results are reported in section 3. In both cases, extended DM-d-MD algorithm allows to rapidly generate a large number of points covering the entire free energy landscape and to recover equilibrium distribution. For alanine dipeptide, we report a 1-order-of-magnitude speedup in both the sampling of critical regions and the recovery of Boltzmann distribution, with compared to plain MD simulations. For alanine-12, standard all-atom MD simulations do not allow to completely explore the free energy landscape at $T = 300$K in a reasonable amount of time, *i.e.*, about 30000 CPU-hours. Using extended DM-d-MD, all important regions of the free energy landscape are explored after only 32 hours of simulations on 64 CPU, namely, 2048

CPU-hours. The sampled regions include important unfolded basins that were left unexplored even using MD simulations at $T = 400$K. Equilibrium distribution is found after 7000 CPU-hours. Convergence criteria are used to estimate the time needed to achieve equilibrium for both systems.

2 Methods used

2.1 Locally scaled diffusion map (LSDMap)

Extended version of Diffusion-Map directed Molecular Dynamics (extended DM-d-MD) presented in this paper is based on the computation of the locally scaled diffusion map (LSDMap) associated with molecular dynamics. LSDMap is a recently developed method[15,22–24] used to decouple molecular dynamics on different time scales in terms of a few collective coordinates. More explicitly, given a set $\{\mathbf{x}_i\}_{i=1}^n$ of simulated data points from molecular dynamics, it consists in defining a graph built on the $\mathbf{x}_i$'s in such a way that each node of the graph corresponds to a point $\mathbf{x}_i$ and each edge is associated with a weight function $\mathbf{K}$ whose matrix elements $K_{ij} \equiv K(\mathbf{x}_i, \mathbf{x}_j)$ can be given as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{2\varepsilon_i \varepsilon_j}\right). \quad (1)$$

Here, $||\mathbf{x}_i - \mathbf{x}_j||$ refers to the RMSD (of all atomic positions) between configurations $\mathbf{x}_i$ and $\mathbf{x}_j$ whereas ε_i stands for a characteristic local scale associated with $\mathbf{x}_i$. The matrix element $K(\mathbf{x}_i, \mathbf{x}_j)$ accounts for the "ease" with which the system can diffuse from the configuration $\mathbf{x}_i$ to the configuration $\mathbf{x}_j$. By using Eq. (1) (up to appropriate renormalization factors[22]) as a starting point, it can be shown that the eigenvectors $\{\Phi_k\}_{k=1}^n$ of matrix $\mathbf{K}$ correspond to collective coordinates, referred to as *diffusion coordinates* (DCs), associated with different time scales of the dynamics [*]. The "first" eigenvector Φ_1, *i.e.*, the one associated with the largest eigenvalue of the spectrum, accounts for the slowest time scale of molecular dynamics, the second eigenvector Φ_2 refers to the second slowest time scale, and so on. The values of the scaling parameters ε_i appearing in Eq. (1) must be chosen carefully: first, ε_i should be small enough so that $\mathbf{K}$ accounts locally for the geometry of the manifold covering the set of data points and second, as we are dealing with a finite number of points, it should be large enough so as to reduce the effects of the local noise. We refer the interested readers to papers by Coifman and Lafon[26,27] for the original mathematical details on diffusion maps and the recent paper[22] about the motivation and construction of a locally scaled version of a diffusion map. In the latter, the local scale is estimated from the intrinsic dimensionality around each data point via multidimensional scaling (MDS)[28] (see also Refs. 15). This implies finding the eigendecomposition of the distance matrix associated with the data set. Even though some recent software libraries allows to quickly estimate the eigenfunctions of large sparse matrices[29], this kind of procedure turned out to be time-consuming with respect to short MD simulations (*i.e.* from 1 to 10 ps). As detailed in the next section, extended DM-d-MD algorithm computes the DCs of each data point periodically after short MD simulations. At this stage, computing a correct local scale for each point using MDS is no longer feasible. The way to set the local scale in the extended DM-d-MD algorithm is discussed below.

According to the original papers on diffusion maps[26,27], computing LSDMap kernel from Eq. (1) requires that the distribution of the $\mathbf{x}_i$'s corresponds to a quasi thermal equilibrium. Nevertheless, it is possible to compute a LSDMap kernel when a non-equilibrium distribution of points is involved. This implies that the statistical weights w_i which account for the equilibrium situation is known for each $\mathbf{x}_i$[24]. The elements of the "weighted" kernel are computed as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_i w_j}\exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{2\varepsilon_i \varepsilon_j}\right). \quad (2)$$

As shown in the next section, even though extended DM-d-MD is a sampling technique that deals with non-equilibrium distributions of points, we have combined it with a reweighting scheme. Whenever new samples are generated, we are able, based on nearest neighbors search, to attribute the correct weights to each point in order to recover the correct quasi-equilibrium distribution. These weights are in turn used as the w_i's needed to compute the weighted kernel elements (2) during the next selection step, and so on. Further details on the implementation of the reweighting scheme are given in the next section. When applying extended DM-d-MD to alanine dipeptide and alanine-12, a uniform local scale $\varepsilon = \varepsilon_i$ is used for all i. However, the value of ε is recomputed at each iteration. To account both for the average effect of the local noise and for the non-equilibrium distribution of points, we set ε as the average distance between each point $\mathbf{x}_i$ and it's p neighbor, where p is the first neighbor such that the sum of the weights of all previous neighbors is approximately equal to $\sqrt{n}$, n being the total number of points. More explicitly, by introducing the function of two variables η such that $\eta(x_i, k)$ gives the index of the k^{th} nearest neighbor of point x_i, we have

$$\varepsilon = \langle ||x_i - x_{\eta(x_i,p)}|| \rangle_i \text{ where } p \text{ satisfies } \sum_{k=1}^{p} w_{\eta(x_i,k)} \simeq \sqrt{n}, \quad (3)$$

where $||...||$ stands for the RMSD and $\langle ... \rangle_i$ is an average over all configurations.

[*] More explicitly, if we call $\{\lambda_k\}_{k=1}^n$ the eigenvalues associated to the eigenvectors $\{\Phi_k\}_{k=1}^n$, it is found that the component i of Φ_k corresponds to the k^{th} diffusion coordinate evaluated at $\mathbf{x}_i$. In other words, one can note $\Phi_{k,i} \equiv \Phi_k(x_i)$, so that the diagonalization of the matrix $\mathbf{K}$ is equivalent to $\sum_{j=1}^n K(x_i, x_j)\Phi_k(x_j) = \lambda_k \Phi_k(x_i)$.

2.2 Extended diffusion-map-directed MD and nearest-neighbors-based reweighting

As mentioned in the introduction, our main algorithm has been designed (i) to enhance the sampling of MD trajectories, especially along high-energy barriers of macromolecular systems and (ii) to recover information about Boltzmann distributions directly from the biased trajectories. The former is made possible via an extended version of diffusion-map-directed MD (extended DM-d-MD)[21] which lies in periodically restarting MD trajectories by selecting the new starting points so as to obtain a uniform distribution of their first and second DCs. Since the first two DCs are associated with the slowest time scales of the dynamics, by restarting new MD trajectories from such a distribution of points, it becomes possible to visit a wider area of the configuration space without remaining trapped in local minima. Regarding Boltzmann distribution, *i.e.*, point (ii), it is based on the assignment of statistical weights to each trajectory. In order to save information associated with molecular dynamics, an appropriate reweighting scheme, based on nearest neighbor search, is used each time MD trajectories are restarted. The weights of each point are used self-consistently in order to compute LSDMap at the next iteration on the basis of equation (2).

The algorithm can be summarized in four steps:

1. When $t = 0$, run short MD trajectories starting from an initial distribution of n points (one trajectory per point) and assign to each of them a statistical weight w_i equal to 1. When $t > 0$, run short MD trajectories starting from the endpoints selected in step 3 (the starting velocities are the velocities associated with each endpoint coming from previous MD simulations).

2. Compute LSDMap from the endpoints of each trajectory and store their first and second DCs. LSDMap kernel elements should be computed from Eq (2) by taking into account the weights w_i computed in step 4 (at the first iteration, all the weights are equal to 1 in accordance with step 1). The local scale is computed according to Eq (3).

3. Select n new MD starting points among the endpoints so that the distribution of new points is uniform along the first and second DCs. The same endpoint can be selected as a new point more than once or can be not selected at all. The exact procedure is detailed in the paragraph "selection step" below.

4. Before running new MD simulations, update the statistical weights w_i of each MD trajectory so as to conserve locally the probability density associated with molecular dynamics after step 3. Each time more than one trajectory will restart from a given endpoint, the weight of the new trajectories will be the original weight divided by the number of trajectories. Whenever an endpoint is not used as a new starting point, distribute the original weight over the first nearest new MD trajectories (see details in paragraph "reweighting step" below). Go back to step 1.

In the following, additional information are given regarding steps 3 and 4.

Selection step (step 3) Let us suppose that we have run n short MD trajectories and that we have computed the 1st and 2nd DCs of each endpoint using LSDMap (steps 1 and 2 above). We want to select n new starting points among the endpoints so that the distribution of starting points is uniform along the first two DCs.† The procedure to select one of those starting points is the following: first, a histogram is built up from the 1st DC values of the endpoints. Then, a random number is generated uniformly between the minimum and maximum 1st DC values. The endpoints located in the same bin as the random number are identified. A second random number is generated uniformly between the minimum and maximum 2nd DC values of the endpoints located in the bin. The new starting point will be the one, among these endpoints, with the closest 2nd DC value to the random number. The above procedure is repeated in order to select n new configurations. To sum up, at each selection step, new starting points are picked from the endpoints in such a way that the distribution of 1st DCs is uniform. For each "bin" of 1st DCs values, the 2nd DCs of the new points are also distributed uniformly. Our choice of a uniform sampling along the first two DCs is motivated by the observation that, in general, points located around transition regions will have very different values of their DCs whereas the DCs of points located inside a given minimum will be globally the same. Within these minima, regions associated with faster transitions can be identified from points having very different values of their 2nd DC whereas points with similar 2nd DC will account for local minima, and so on. Using a uniform sampling along the first two diffusion coordinates to select the new starting points thus appears to be a natural choice to cover the largest possible area of the configuration space, *i.e.*, to visit yet unexplored regions without being trapped within local minima. Further details on the motivation behind our procedure to select the new starting points are given in the supporting information.

As mentioned in the introduction, our sampling method is comparable to the recently proposed diffusion-map-directed molecular dynamics (DM-d-DM)[21]. However, several changes have been made with respect to the original version: (i) multiple parallel MD trajectories are run rather than a

† The number of endpoints can be slightly different from the number of new starting points since extra endpoints can be selected as new starting points during the reweighting step (step 4) in case they have no close nearest neighbors.

single one; (*ii*) we do not restart all the *n* trajectories from the same endpoint, rather, the new starting points are picked among the endpoints of each short MD trajectory so that the distribution of 1st and 2nd DCs is uniform; (*iii*) since the new starting points will cover a large area of the energy landscape, appropriate reweighting scheme can be used to locally save the information about Boltzmann distribution despite the biased introduced with the sampling of the new starting points (see paragraph "reweighting step" below).

Our procedure is also similar in spirit to the recently proposed transition current sampling method[16–20] mentioned briefly in the introduction. Apart from our reweighting scheme, the main difference stems from the way of selecting the new points; for transition current sampling, the new points are selected so as to push the dynamics towards regions of high current density – transition regions, typically. In this case, the probability of starting a new trajectory from an endpoint is computed via the Hessian matrix of the potential energy at this particular endpoint, *i.e.*, which accounts locally for the curvature of the free energy landscape. For complex high-dimensional systems, periodically estimating the Hessian matrix can be computationally expensive with respect to MD simulations.

Reweighting step (step 4) The idea of the reweighting is to recover information about the probability density associated with molecular dynamics right after the selection step and before MD simulations are restarted from the new selected points. In the case of normal MD simulations, free energy landscapes are constructed by locally estimating the density of configurations that were obtained as a result of the simulations. In the density estimate, each MD configuration has a weight equal to 1. The weights are summed locally to compute the correct density and thus the correct free energy landscape. In the case of extended DM-d-MD, a bias in the density distribution is introduced as soon as some configurations are selected as the starting points of new MD trajectories (selection step, step 3). In order to correct for this bias a reweighting scheme is introduced: new weights are reassigned to each configuration in such a way that locally the sum of the weights is approximately the same before and after the selection step. By computing the local density from the sum of the weights, the free energy landscape associated with the unbiased dynamics is approximately recovered. Supposing that the starting points of the new MD trajectories were determined in step 3, our reweighting scheme can be summarized as follows: each time *one or more* trajectories will restart from a particular endpoint, we attribute to these trajectories statistical weights such that their sum equals the statistical weight of the "old trajectory", *i.e.*, each new trajectory will have the statistical weight of the trajectory that stopped at this particular endpoint at the last iteration, divided by the number of new starting trajecto-

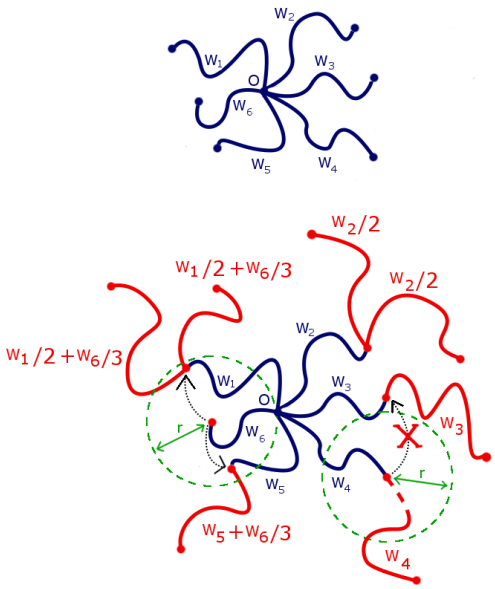

Fig. 1 Illustration of the reweighting scheme used in our algorithm with six trajectories starting from the same point O. We note $\{w_i\}_{i=1}^{6}$ the initial weights of each trajectory. (a) shows the trajectories before the first selection step; the endpoints of each trajectory are highlighted. (b) shows trajectories before the first selection step (in blue) and trajectories between the first and second selection steps (in red). The number of red trajectories starting from each endpoint of the blue trajectories is given for illustrative purpose. The weights of the red trajectories are computed according to our reweighting scheme. If an endpoint of a blue trajectory is not used as the starting point of red trajectories, we look for new starting points within a RMSD range of *r* around the endpoint (when applying our algorithm, we only considered the first ten nearest points within this range). The weights of the new starting trajectories within the range will be increased by the weight of the old trajectory divided by the number of trajectories. If no new starting points is located within the RMSD range (for example, at the bottom right of Figure (b)), the trajectory is kept. In this case, the number of new trajectories may slightly differ from the original number set at $t = 0$. If an endpoint of a blue trajectory is used as the starting point of red trajectories, we attribute to these trajectories weights such that their sum equals the weight of the blue trajectory (plus the weights of possible dead nearest neighbors).

ries. Alternatively, if *no* trajectories will restart from a given endpoint, we look for the new MD trajectories whose starting points are among the 10 nearest neighbors in RMSD ‡ of this specific endpoint. The weight of the old trajectory is distributed over these new trajectories, *i.e.*, the weight of each new nearest MD trajectory is increased by the weight of the old trajectory divided by the number of trajectories. The reason of considering the first ten nearest starting points – *e.g.*, instead of the first nearest point – is that we want to smooth locally the distribution of weights in such a way that we do not finally end up with single points having a very large weight. Such singularities might be a problem especially when computing LSDMap that is sensitive to local dynamical properties. Moreover, to avoid spreading the weights over nearest neighbors located too far away from a given endpoint, we introduce a cutoff distance in RMSD r. If a given endpoint should be removed after the selection step but no nearest neighbors is found within this range, the point is finally kept, *i.e.*, it will be used as the starting point of new MD trajectories. A natural choice for r is provided by the local scale used in LSDMap computation (step 2) which accounts for the scale of the noise. In other words, we set $r = \varepsilon$, where ε is given by Eq. (3). In this way, the sum of the weights of the trajectories is expected to be locally the same before and after the selection step (step 3). Since the statistical weights are initially set to 1 for each trajectory, by summing them locally, we obtain the probability density associated with MD; thus this quantity will be approximately unaffected by the bias introduced in step 3.

3 Results and discussion

In this section, we present the main results obtained by applying the algorithm introduced in section 2.2 to two systems: alanine dipeptide and alanine-12. Alanine dipeptide is a standard test case for sampling methods, as it is a small (22 atoms) and well-studied system, and its dynamics contains processes at very different time scales. Alanine-12 is a much more challenging system (135 atoms) characterized by a complex free energy landscape whose exploration using all-atoms MD requires significant computational resources. As specified below, the number of short MD trajectories was set to $n = 10000$ for alanine dipeptide and to $n = 5000$ for alanine-12. At first sight, running multiple MD trajectories may appear to be computationally demanding compared with single MD trajectories. However, since the dynamics of MD trajectories are all independent, our code has been easily parallelized. Information about the parallelization procedure can be found in the supporting information.

‡ For complex systems, other metrics than RMSD may lead to more accurate partitioning of the conformational space [30,31]. Nevertheless, using RMSD appears reasonable here as only close configurations are involved.

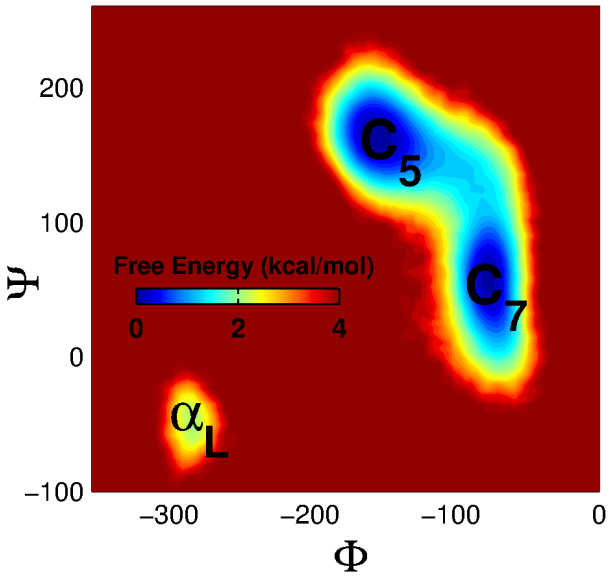

Fig. 2 Free energy landscape of alanine dipeptide in equilibrium in terms of Φ/Ψ angles. The system is characterized by three minima: C_5, C_7 and α_L. The transition between α_L and C_5-C_7 minima is the slowest one; using regular MD, it is around three orders of magnitude slower than the $C_5 \leftrightarrow C_7$ transition.

3.1 Alanine dipepdide

Simulations on alanine dipeptide have been performed using GROMACS with Amber03 force field in vacuum. In order to visualize the results, we projected the trajectory data on the dihedral angles Φ and Ψ, two commonly used reaction coordinates for alanine dipeptide. The free energy landscape of alanine dipeptide is characterized by three important energy minima: C_5, C_7 and α_L as shown in Figure 2. Whereas the transition between C_5 and C_7 is relatively fast – of the order of 50 ps using normal MD with our setup – the transition involving α_L with the two other minima is definitely slower, around two orders of magnitude slower than the $C_5 - C_7$ transition. Since our algorithm has been designed to handle high-energy barriers, we will focus the application on the transition to α_L minimum. In our simulations, $n = 10000$ trajectories have been considered initially, each starting at $t = 0$ from the same configuration inside the C_7 minimum with coordinates $(\Phi, \Psi) = (-68.0, 54.9)$. Parameters related to MD such as the temperature, the viscosity (per mass unit) and the time step h have been set to $T = 300$K, $\gamma = 1.0$ ps^{-1} and $h = 10^{-3}$ ps, respectively. To run extended DM-d-MD algorithm, we had to set the length Δt of the MD trajectories. The algorithm was run for different values of Δt from $\Delta t = 800$ fs to $\Delta t = 2$ ps. As expected, fast exploration of the free energy landscape was reported for small values of Δt. As reported in the support in-

formation, significant speedup with respect to plain MD simulations was observed for $\Delta t < 1.5$ ps, *i.e.*, of the order of $1/\gamma$. Using smaller values of Δt does not contribute to significantly increase the overall performance of the algorithm.

The free energy was computed from the equation: $F(\Phi,\Psi,t) = -kT\ln(P(\Phi,\Psi,t))$ where the probability density $P(\Phi,\Psi,t)$ has been estimated by summing locally the weights w_i associated with each trajectory. In other words, we have

$$P(\Phi,\Psi,t) \simeq \sum_{i=1}^{n} w_i \delta(\Phi - \Phi_i(t))\delta(\Psi - \Psi_i(t)).$$

On the top of Figure 3, we have plotted the free energy landscape of alanine dipeptide after 2 iterations of our algorithm. MD trajectories with length $\Delta t = 1$ps were used, thus the figure was obtained at $t = 2$ps. Even though equilibrium has not been reached, many samples have been generated along the transition path to α_L, including the minimum itself. As a fair comparison, using plain MD simulations with the very same parameters, the α_L minimum is significantly populated after at least 50 ps, *i.e.*, around one order of magnitude slower than our algorithm. Although the free energy landscape of alanine dipeptide is explored very rapidly using extended DM-d-MD, one must wait longer before Boltzmann distribution is completely recovered. On the bottom of Figure 3, we have plotted the free energy landscape obtained at $t = 600$ps (that is, 600 iterations using MD trajectories with length $\Delta t = 1$ps) where we can see that Boltzmann distribution has been achieved. To compare the convergence to equilibrium between plain MD simulations and extended DM-d-MD, we have introduced the coefficient σ defined as the difference between the minimum free energy in α_L minimum and the global minimum free energy on the entire free energy landscape. On the top of Figure 4, $\sigma(t)$ has been plotted using 10000 plain MD simulations (again with $T = 300$K, $\gamma = 1.0\ \text{ps}^{-1}$ and $h = 10^{-3}$ ps) showing that Boltzmann distribution is reached after $t = 2000 - 3000$ ps. On the bottom of Figure 4, we show the comparison with $\sigma(t)$ computed from our algorithm using trajectories of length $\Delta t = 1$ps. The convergence is reached after $t = 300 - 400$ ps, that is, about one order of magnitude faster than plain MD simulations. The limit value for σ matches perfectly with MD results. Additional details on the convergence and its sensitivity to the parameters can be found in the supporting information.

Even though the distribution of weights is changing over hundreds of picoseconds, the associated distribution of points becomes stationary very soon (after a dozen iterations) covering all the important minima of the free energy landscape. In Figure 5, we have reported the distribution of points as a function of Φ and Ψ when Boltzmann distribution is reached. The density is given as an effective free energy. The reaction paths leading to α_L minimum can be clearly identified; it is seen that points starting from the C_7 minimum are unlikely to reach α_L

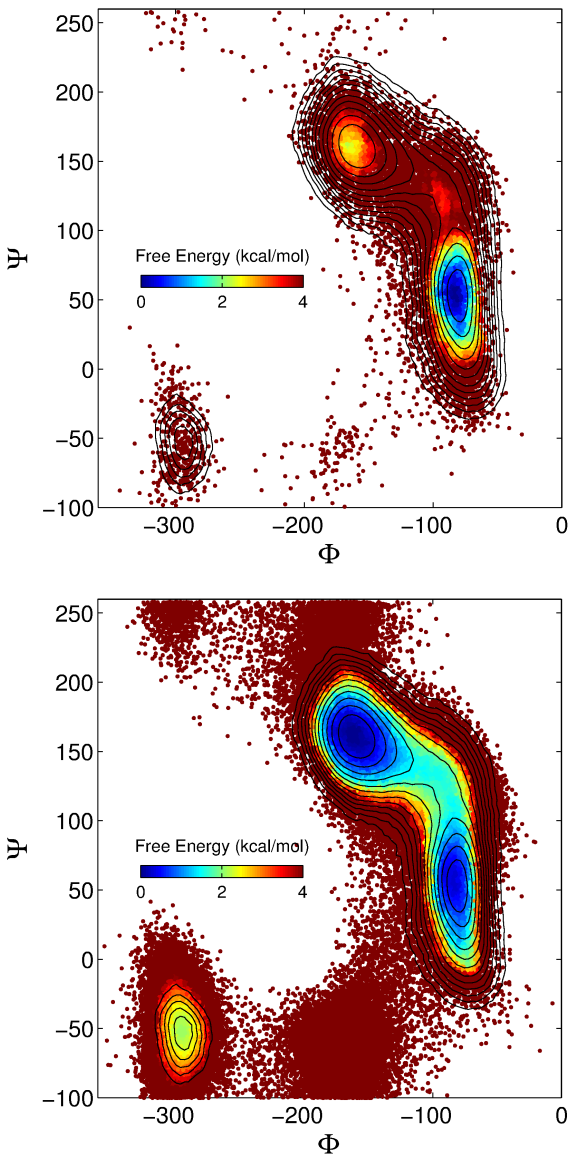

Fig. 3 Top: free energy landscape of alanine dipeptide obtained at $t = 2$ ps using extended DM-d-MD algorithm with reweighting (two iterations using trajectories with length $\Delta t = 1$ps). Bottom: free energy landscape of alanine dipeptide obtained at $t = 600$ ps using extended DM-d-MD algorithm with reweighting. The contour plots correspond to Boltzmann distribution obtained with regular MD; each contour marks an increase of the free energy of 0.4 kcal/mol. Note that on the bottom figure, 34 snapshots around $t = 600$ ps with 10000 points each, have been merged together for the sake of clarity.

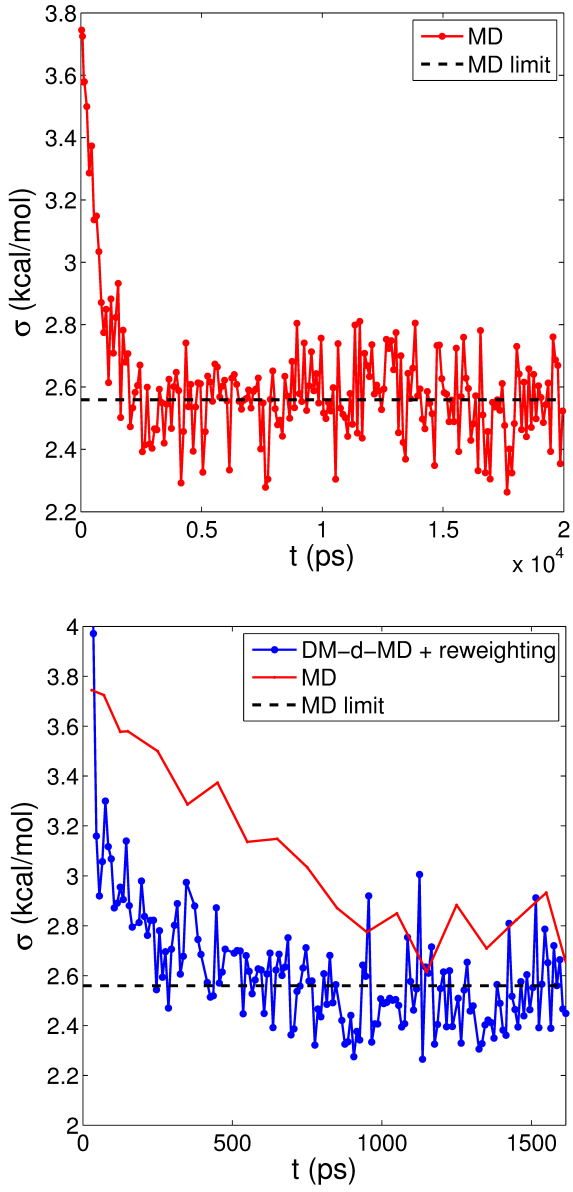

Fig. 4 Top: using standard MD simulations, convergence parameter σ of alanine dipeptide in vacuum as a function of time. σ is computed as the difference between the minimum free energy in α_L minimum and the global minimum free energy on the entire free energy landscape. Each value of $\sigma(t)$ was obtained by merging 5-6 snapshots of the free energy landscape at times t' close to t. The black dashed line corresponds to the (averaged) limit of $\sigma(t)$ at large t using plain MD simulations. Bottom: comparison with $\sigma(t)$ obtained from extended DM-d-MD algorithm (the length of each short MD trajectory is $\Delta t = 1$ps).

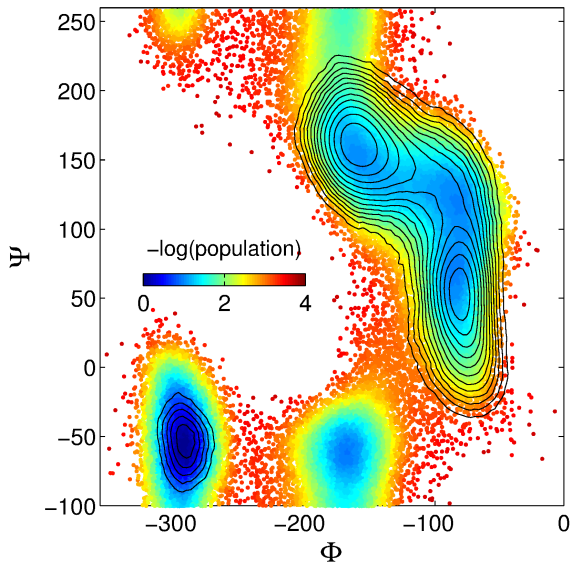

Fig. 5 Distribution of points obtained when applying extended DM-d-MD on alanine dipeptide by merging 34 snapshots around $t = 600$ ps. It is given as an effective free energy $-kT\ln(n)$ where n is the density of points. The contour plots correspond to Boltzmann distribution obtained with regular MD; each contour marks an increase of the free energy of 0.4 kcal/mol.

without going through the C_5 minimum first. In order to visualize the efficiency of the selection step in our main algorithm (step 3), we have reported, in Figure 6, the two most probable distributions of 1st DCs that can be observed over time. Both types of distributions are generally observed within 1-2 iterations of our algorithm. Again, we should recall that, at each iteration, the new starting points are picked so as to get a uniform distribution of points along the first two DCs. On the top of Figure 6, we can see the distribution of 1st DCs when both α_L and $C5-C7$ minima are well populated (the points have been stored between step 2 and step 3 (see section 2.2), *i.e.*, right before the selection step). In this case, it turns out that points located inside α_L or $C5-C7$ minima share globally the same 1st DC whereas points located on both sides of α_L barrier cover a wide range of the 1st DC values. Thus new starting points will be mainly located on the barrier whereas points within both minima will be more likely to be definitely remove. The weight of points from which multiple MD trajectories are restarted will be spread over all the new starting trajectories whereas the weights of killed trajectories will be distributed to their nearest neighbors according to our reweighting scheme (see section 2.2). The distribution of 1st DCs on the bottom of Figure 6 is representative of a distribution when $C5-C7$ minimum is less populated. As noticed, the new points are still favorably selected around the barrier

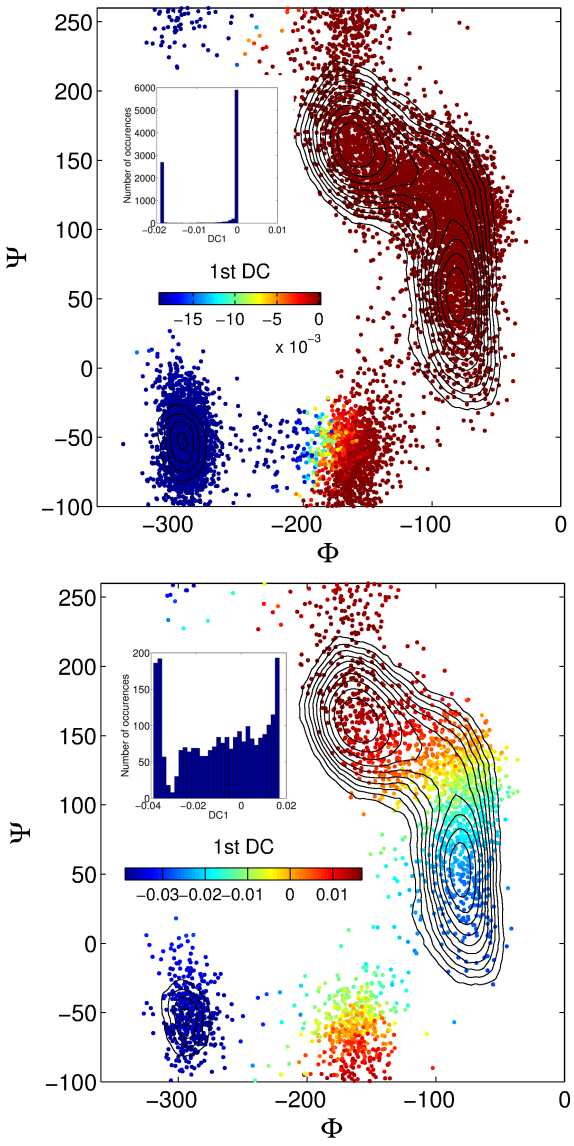

Fig. 6 Top and bottom: the two most representative distributions of 1st DCs that can be obtained over time from our algorithm as a function of Φ/Ψ angles. Insets stand for corresponding histograms along the 1st DC.

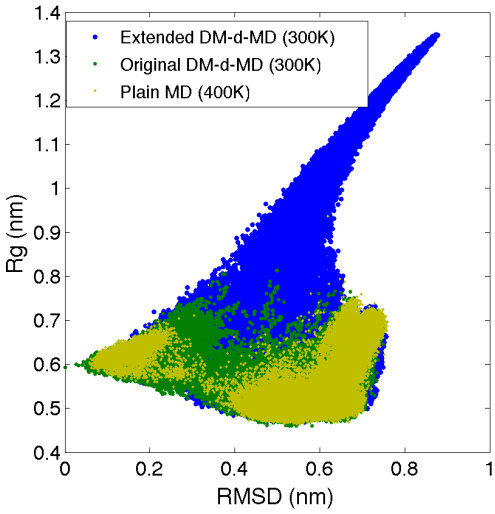

Fig. 7 In blue, configurations obtained when applying extended DM-d-MD algorithm on alanine-12 as a function of the RMSD from the helical native state and the radius of gyration Rg (blue). The data are obtained after 32 hours of simulation on 64 CPUs using 5000 trajectories, corresponding to 2048 CPU-hours. We have also superimposed the original DM-d-MD simulations data (in green) and plain MD simulations data (in yellow) obtained by Zheng et al.[21] at 300K after 6000 CPU-hours and at 400K after 24000 CPU-hours, respectively.

to α_L minimum but also along $C5-C7$ minima. More details about the distributions of diffusion coordinates (including the role of 2nd DCs) can be found in the supporting information both for alanine dipeptide and alanine-12.

3.2 Alanine-12

In order to test our approach on a more challenging system, we have applied our extended DM-d-MD algorithm to sample the configuration space of alanine-12, which has a much more complex free energy landscape than alanine dipeptide. Simulations were performed using GROMACS with Amber96 force field in vacuum. The value of the viscosity and the MD time step were set equal to $\gamma = 0.5$ ps^{-1} and $h = 2.10^{-3}$ ps, respectively, whereas the length of the short MD trajectories was set to $\Delta t = 1$ ps.

Starting from a helical configuration, the unfolding events for this system are too rare to be adequately sampled using standard MD simulations at 300 K with our computational resources. We have observed one single misfolding event in 2 full CPU days of 100 MD trajectories in parallel. To give an idea, Zheng et al.[21] performed plain MD simulations of alanine-12 with the same setup at 400K. Starting from the same folded helical configuration, they have been able to find

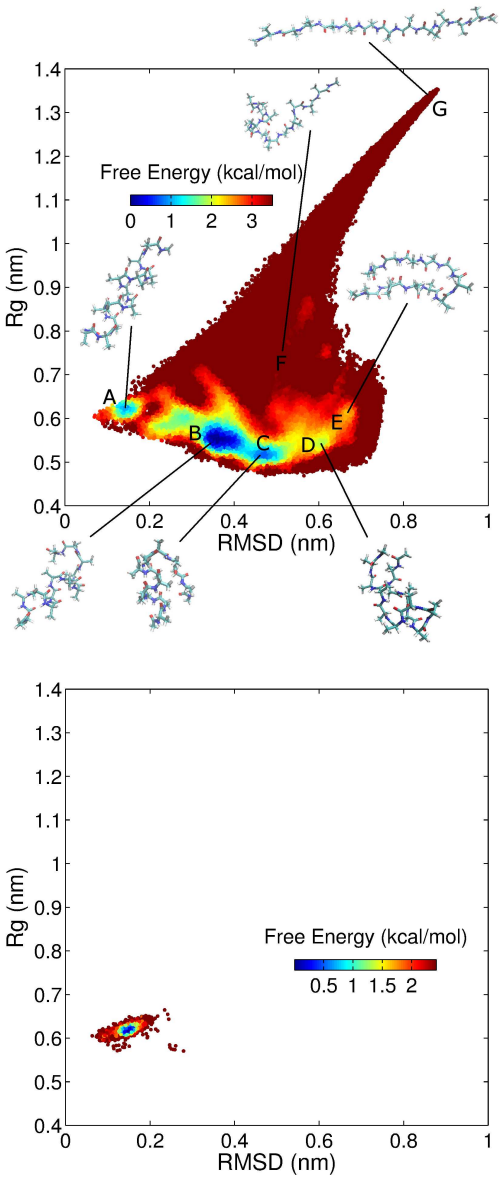

Fig. 8 Top: free energy landscape of alanine-12 obtained with extended DM-d-MD algorithm, typically when main unfolded basins become visible (convergence is not reached yet). Representative configurations of important local minima or critical regions are shown. Bottom: free energy landscape of alanine-12 obtained by plain MD simulations after the same amount of computational time as the top figure. Both plots were obtained after 65 hours of simulations on 64 CPUs using 5000 trajectories, which corresponds to 4160 CPU-hours. In both cases, the same parameters for MD simulations are used: $T = 300$K, $\gamma = 0.5$ ps^{-1} and $h = 2.10^{-3}$ ps. For the sake of accuracy on the top figure, 60 snapshots (*i.e.*, about 300000 points) stored around the same CPU time, are merged together.

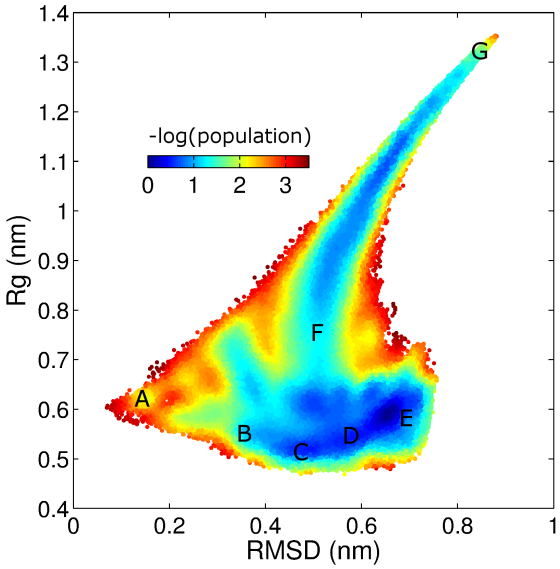

Fig. 9 Distribution of points obtained when applying extended DM-d-MD on alanine-12 after 4160 CPU-hours using 5000 trajectories (simulation time: 2.5 ns). It is given as an effective free energy $-kT\ln(n)$ where n is the density of points.

important unfolded basins using 40 trajectories of length 4μs, which is equivalent to 24000 CPU-hours. Zheng et al.[21] also tested the original DM-d-MD algorithm on alanine-12. In the original DM-d-MD, the short MD trajectories are all restarted from the same endpoint (associated with the configuration with the largest first DC). As a result, information about Boltzmann distribution cannot be saved during the procedure, which implies that some additional method, like umbrella sampling or state-based models, needs to be used to recover the correct equilibrium distribution. All in all, it was reported that, at 300K, original DM-d-MD algorithm explored the region colored in dark green in Figure 7, in 6000 CPU-hours. Using the extended DM-d-MD algorithm at 300K, the free energy landscape of alanine-12 was completely explored after 2048 CPU-hours using 5000 trajectories, more specifically, after 32 hours of simulations on 64 CPUs. The configurational space spanned by extended DM-d-MD incluses the unfolded regions that were left completely unexplored by original DM-d-MD algorithm at 300K or standard MD simulations at 400K. The data are reported in Figure 7 in terms of two commonly used reaction coordinates for alanine-12, the RMSD from the helical native state and the radius of gyration Rg. After 2048 CPU-hours of extended DM-d-MD, which corresponds to a simulation time of 1.2 ns for 5000 trajectories, the distribution of points remains quasi-stationary. Region around the helical basin $(\text{RMSD},\text{Rg}) = (0.14, 0.62)$ can

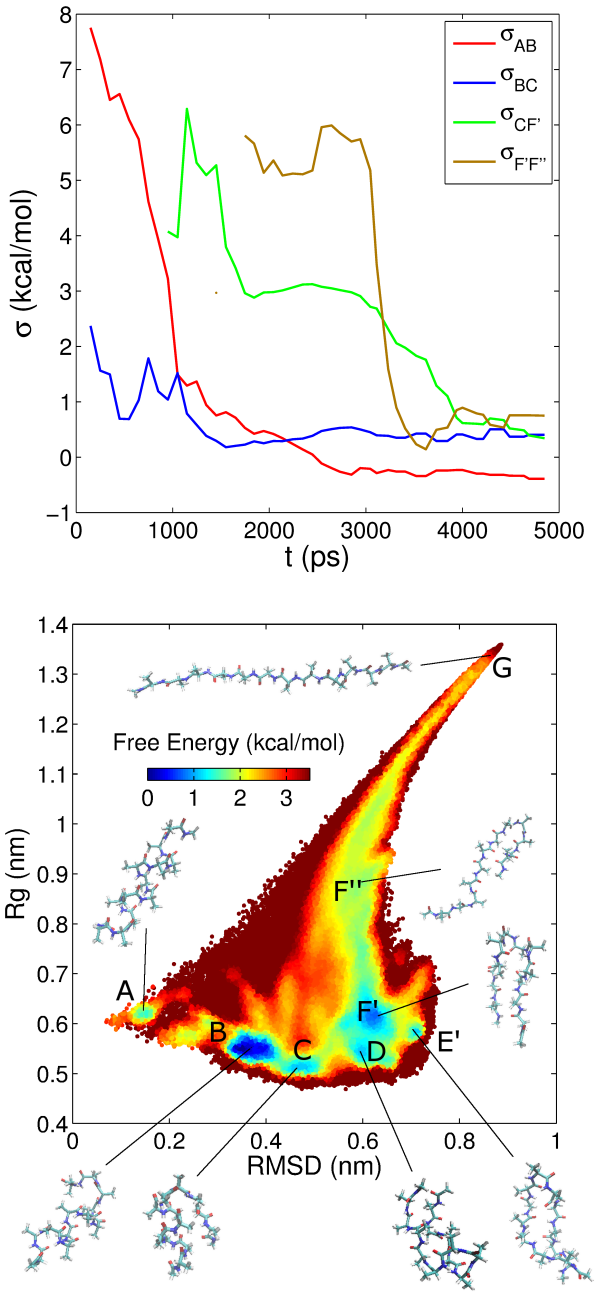

Fig. 10 Top: Evolution of the σ values as a function of the simulation time t. We have noted $\sigma_{AB} = F_B - F_A$, *i.e.*, the difference of free energies between minima B and A and so on. Labels associated with each minimum can be found on the bottom figure. Bottom: free energy landscape of alanine-12 obtained with extended DM-d-MD algorithm after 7000 CPU-hours using 5000 trajectories (simulation time: 4.1 ns), typically when the σ values become all stable.

remain unsampled for short periods of time but, similarly to MD simulations, data from multiple snapshots obtained at previous times can be merged together to obtain more accurate free energy estimate. Like in the case of alanine dipeptide, even when the free energy landscape has been completely explored, extra time is required to recover the correct distribution of weights associated with thermal equilibrium. On the top of Figure 8, we have plotted the free energy landscape obtained after 4160 CPU-hours using 5000 trajectories (simulation time: 2.5ns), typically when important unfolded minima become apparent. On the bottom of Figure 8, we show the free energy landscape after the very same computational amount of time using 5000 plain MD simulations at 300K. It is found that even the closest misfolded minimum B has not been reached yet. On the contrary, using extended DM-d-MD algorithm, many interesting metastable states are already found at such an early stage of the simulation. Typical configurations in state B along the pathway between folded and unfolded states indicate that the helical turn at the N-terminus breaks first during the unfolding. State E corresponds to a hairpin structure, whereas state D corresponds to a curved hairpin structure. More information about the main pathways leading to unfolded structures can be found by looking at the density of points (Figure 9). For example, we find that only one single pathway (from state C to state F) leads to the completely unfolded structure (state G) whereas it is seen that points in state E are unlikely to reach states F or G without going through the C-D minimum first. Similarly to alanine dipeptide, the sampling significantly slows down when larger values of Δt are used. Typically, when $\Delta t \simeq 1.5$ ps, only a small part of the energy landscape (from minimum A to minimum C) is covered after 4000 CPU-hours. Like in the case of alanine dipeptide, convergence to equilibrium is measured via a set of σ parameters defined as the difference in free energy between representative minima of the free energy landscape. More explicitly, we note $\sigma_{AB} = F_B - F_A$, *i.e.*, the difference in free energy between minima B and A and so on. On the top of Figure 10, it is shown that the σ values become all stable after a simulation time $t = 4$ ns suggesting that equilibrium has been reached. On the bottom of Figure 10, we have plotted the free energy landscape at that time, which corresponds to 7000 CPU-hours using 5000 trajectories. Minima A, B, C, and D which correspond to folded and misfolded regions can be identified like at $t = 2.1$ ns (top of Figure 8). New minima are reported in the unfolded region including hairpin structures (E' and F') which slightly differ from basin E identified at 2.1 ns.

4 Conclusion

In this paper, a new sampling strategy, extended DM-d-MD, has been introduced to rapidly explore the free energy landscape of macromolecular systems characterized by high free

energy barriers. Based on LSDMap, a multidimensional reduction technique, our method provides a way to periodically sample MD trajectories to cover the widest possible region of the free energy landscape at a given time, including critical transition regions. The algorithm is combined with a reweighting scheme ensuring that information about Boltzmann distributions is kept despite the biased dynamics. We observe that Boltzmann distribution is achieved much faster than plain MD simulations. As a case in point, our present algorithm was applied to two test systems of biological interest, alanine dipeptide and alanine-12. For alanine dipeptide, we report a speedup of 1 order of magnitude both in the sampling of critical regions and in the recovery of Boltzmann distribution. For alanine-12, characterized by a more complex energy landscape, the speedup is significantly larger as all important minima were sampled after 2048 CPU-hours whereas plain MD simulations do not allow to correctly explore the configuration space within at least 24000 CPU-hours. Equilibrium distribution is found after 7000 CPU-hours. In addition, while widely-used sampling techniques such as transition-path-sampling-like or state-based techniques require direct or indirect information on the system's reaction coordinates to define the states or initiate the sampling [22], extended DM-d-MD is reaction coordinate free and can be used without any previous knowledge of the configurational landscape. Because we are considering multiple independent MD trajectories, our method has been easily parallelized so that the amount of computational time saved is quasi-proportional to the number of cups used. Similarly to the recently proposed DM-d-MD method on which our algorithm is partly based, we expect our method to be more efficient when applied to systems characterized by higher free energy barriers. In this way, we expect that this might open the way to a more exact comparison with experimental results as well as predictions not yet accessible to experiment.

Acknowledgements

This work was supported by NSF (grants CHE-1152344 and CHE-1265929 to C.C.), and the Welch Foundation (C-1570 to C.C.). Simulations were performed on the following shared resources at Rice University: BlueBioU was supported in part by NIH award NCRR S10RR02950 and an IBM Shared University Research (SUR) Award in partnership with CISCO, Qlogic and Adaptive Computing; DAVinCI was supported in part by the Data Analysis and Visualization Cyberinfrastructure funded by NSF under grant OCI-0959097. And also on the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant OCI-1053575.

References

1 C. D. Snow, H. Nguyen, V. S. Pande and M. Gruebele, *Nature*, 2002, **420**, 102–106.
2 D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan and W. Wriggers, *Science*, 2010, **330**, 341–346.
3 M. A. Rohrdanz, W. Zheng and C. Clementi, *Annu. Rev. Phys. Chem.*, 2013, **64**, 295–316.
4 C. Dellago, P. G. Bolhuis, F. S. Csajka and D. Chandler, *J. Chem. Phys.*, 1998, **108**, 1964.
5 C. Dellago, P. Bolhuis and P. L. Geissler, *Adv. Chem. Phys.*, 2002, **123**, 1–78.
6 T. S. van Erp, D. Moroni and P. G. Bolhuis, *J. Chem. Phys.*, 2003, **118**, 7762.
7 R. J. Allen, C. Valeriani and P. R. ten Wolde, *J. Phys.: Condens. Matter*, 2009, **21**, 463102.
8 F. A. Escobedo, E. E. Borrero and J. C. Araque, *J. Phys.: Condens. Matter*, 2009, **21**, 333101.
9 T. S. van Erp and P. G. Bolhuis, *J. Comput. Phys.*, 2005, **205**, 157–181.
10 J. Rogal and P. G. Bolhuis, *J. Chem. Phys.*, 2008, **129**, 224107.
11 E. Weinan, W. Ren and E. Vanden-Eijnden, *Phys. Rev. B*, 2002, **66**, 052301.
12 V. S. Pande, K. Beauchamp and G. R. Bowman, *Methods*, 2010, **52**, 99–105.
13 P. Májek and R. Elber, *J. Chem. Theory Comput.*, 2010, **6**, 1805–1817.
14 E. Weinan and E. V. Eijnden, *Annu. Rev. Phys. Chem.*, 2010, **61**, 391–420.
15 W. Zheng, M. A. Rohrdanz, M. Maggioni and C. Clementi, *J. Chem. Phys.*, 2011, **134**, 144109.
16 S. Tănase-Nicola and J. Kurchan, *Phys. Rev. Lett.*, 2003, **91**, 188302.
17 S. Tănase-Nicola and J. Kurchan, *J. Stat. Phys.*, 2004, **116**, 1201–1245.
18 A. Mossa and C. Clementi, *Phys. Rev. E*, 2007, **75**, 046707.
19 M. Picciani, M. Athènes, J. Kurchan and J. Tailleur, *J. Chem. Phys.*, 2011, **135**, 034108.
20 M. Picciani, *PhD thesis*, Politecnico di Milano, Université Paris VI - Pierre et Marie Curie, 2012.
21 W. Zheng, M. A. Rohrdanz and C. Clementi, *J. Phys. Chem. B*, 2013, **117**, 12769–12776.
22 M. A. Rohrdanz, W. Zheng, M. Maggioni and C. Clementi, *J. Chem. Phys.*, 2011, **134**, 124116.
23 M. A. Rohrdanz, W. Zheng, B. Lambeth and C. Clementi, *Proceedings of the Conference on Extreme Science and*

Engineering Discovery Environment: Gateway to Discovery, 2013, 4.

24 W. Zheng, A. V. Vargiu, M. A. Rohrdanz, P. Carloni and C. Clementi, *J. Chem. Phys.*, 2013, **139**, 145102.

25 G. M. Torrie and J. P. Valleau, *J. Comput. Phys.*, 1977, **23**, 187–199.

26 R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner and S. W. Zucker, *Proc. Natl. Acad. Sci. U. S. A.*, 2005, **102**, 7426–7431.

27 R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni and B. Nadler, *Multiscale Model. Simul.*, 2008, **7**, 842–864.

28 A. Little, Y.-M. Jung and M. Maggioni, *Proc. AAAI*, 2009.

29 E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edn, 1999.

30 A. Rajan, P. L. Freddolino and K. Schulten, *PLoS One*, 2010, **5**, e9890.

31 B. Keller, X. Daura and W. F. van Gunsteren, *J. Chem. Phys.*, 2010, **132**, 074110.