AIP | The Journal of Chemical Physics

Density-based cluster algorithms for the identification of core sets

Oliver Lemke and Bettina G. Keller

Citation: The Journal of Chemical Physics **145**, 164104 (2016); doi: 10.1063/1.4965440
View online: http://dx.doi.org/10.1063/1.4965440
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/145/16?ver=pdfcov
Published by the AIP Publishing

AIP Applied Physics Reviews
NEW Special Topic Sections
NOW ONLINE
Lithium Niobate Properties and Applications: Reviews of Emerging Trends
AIP Applied Physics Reviews
apr.aip.org

Density-based cluster algorithms for the identification of core sets

Oliver Lemke and Bettina G. Keller[a)]
Department of Biology, Chemistry, Pharmacy, Freie Universität Berlin, Takustraße 3, D-14195 Berlin, Germany

(Received 10 August 2016; accepted 5 October 2016; published online 26 October 2016; publisher error corrected 27 October 2016)

The core-set approach is a discretization method for Markov state models of complex molecular dynamics. Core sets are disjoint metastable regions in the conformational space, which need to be known prior to the construction of the core-set model. We propose to use density-based cluster algorithms to identify the cores. We compare three different density-based cluster algorithms: the CNN, the DBSCAN, and the Jarvis-Patrick algorithm. While the core-set models based on the CNN and DBSCAN clustering are well-converged, constructing core-set models based on the Jarvis-Patrick clustering cannot be recommended. In a well-converged core-set model, the number of core sets is up to an order of magnitude smaller than the number of states in a conventional Markov state model with comparable approximation error. Moreover, using the density-based clustering one can extend the core-set method to systems which are not strongly metastable. This is important for the practical application of the core-set method because most biologically interesting systems are only marginally metastable. The key point is to perform a hierarchical density-based clustering while monitoring the structure of the metric matrix which appears in the core-set method. We test this approach on a molecular-dynamics simulation of a highly flexible 14-residue peptide. The resulting core-set models have a high spatial resolution and can distinguish between conformationally similar yet chemically different structures, such as register-shifted hairpin structures. *Published by AIP Publishing.* [http://dx.doi.org/10.1063/1.4965440]

I. INTRODUCTION

In recent years, Markov state models (MSMs) have developed into an extremely useful tool for the analysis of complex molecular dynamics. These models are parametrized from molecular-dynamics simulation (MD) data by discretizing the conformational space and counting the observed transitions between pairs of states. MSMs have been used to investigate such diverse processes as protein folding,[1,2] protein misfolding,[3] ligand binding,[4] allostery,[5] amyloid formation,[6–8] and solvent-dependent conformational dynamics.[9] Once a sufficiently accurate MSM has been obtained, the model yields insight into long-lived conformations (also called metastable sets), the kinetic exchange rates between them, and the hierarchy in the free-energy landscape.[10] Yet, the actual construction of a MSM from MD data is still difficult, because the accuracy of a MSM, i.e., whether or not it faithfully represents the slow conformational dynamics of the system, depends sensitively on the discretization of the conformational space. Often a large number of states is required to achieve an acceptable approximation error, while on the other hand the statistical error increases when more states are added to the model.

The approximation error due to the discretization depends both on the number of states as well as on the exact choice of the state boundaries. For example, if a single state covers two minima in the potential energy landscape of the molecule, the transitions between these minima are not resolved by the corresponding MSM. Even if a state boundary is introduced between the minima, trajectories which leave minimum one, cross the boundary, but immediately return to minimum one before actually visiting the center of minimum two will generate two transition counts between these two minima and the resulting MSM will underestimate the actual transition time scale (recrossing problem). This problem becomes worse if the boundary is not positioned exactly on top of the energy barrier but somewhat closer to either of the minima. One cannot completely avoid this approximation error but, from an analysis of the transfer operator of the dynamics, it is known how the discretization affects this error.[11] In an optimal discretization, the state boundaries are chosen such that the dominant eigenfunctions of the transfer operator can be well represented. This often requires a high resolution in the transition areas between long-lived conformations, because the dominant eigenfunctions vary in these regions. On the other hand, a lower resolution can be afforded within the long-lived conformations because in these regions of the conformational space the eigenfunctions are often relatively constant. Unfortunately, it is difficult to translate this knowledge into a discretization algorithm because it requires prior knowledge of the long-lived conformations and the transition regions between them.

The insight into the approximation error has nonetheless led to several new methods which improve the definition of states, including discretization methods based on diffusion maps,[12] adaptive discretization schemes,[13,14] and methods which first identify an optimal low-dimensional subspace

[a)]Electronic mail: bettina.keller@fu-berlin.de and then construct the discretization in this subspace.[15,16] An alternative strategy to improve the discretization—the variational approach to molecular dynamics[17–19]—abandons the use of discrete states and instead uses functions of the conformational space. Because one can choose functions which smoothly vary in the transition regions, very few basis functions can be sufficient to achieve a highly accurate model of the conformational dynamics. In fact, to achieve a comparable approximation error with a conventional MSM of the same molecule, the number of states can be more than an order of magnitude larger than the number of basis functions in the variational model. However, similar to conventional MSMs, the basis set has to be chosen such that the dominant eigenfunctions can be well represented as a linear combination of the basis functions.[19]

Core-set models[20,21] are a discretization method which has been proposed early in the discussion and which uses committor functions as basis functions. Committor-functions assign the value zero to one region of the conformational space and the value one to another region, and smoothly interpolate in between.[20,22–25] Choosing these regions to be in the core of a long-lived conformation (hence the name of the method), the committor functions have a similar functional form as the dominant transfer operator eigenfunctions and are therefore excellent basis functions. In contrast to the variational approach, the committor functions do not need to be known analytically, but the model can be parametrized from a transformation of the MD trajectory to so-called milestoning processes.[20,22–25] Although the core-set model is an elegant way to transfer the knowledge on the approximation error into an algorithm, the method has not been used frequently. This presumably has two reasons. First, one needs prior knowledge of the long-lived conformations to define the cores. This is somewhat easier than finding optimal states for an MSM since no knowledge on the transition region is required, but it is still not trivial. Second, the cores need to be sufficiently metastable. This is easily fulfilled for molecules with a few highly populated long-lived conformations but might not be the case in molecules with complex molecular dynamics and multiple metastable conformations with different relative populations.

In an earlier publication,[26] we have developed the common-nearest neighbor algorithm (CNN), a density-based cluster algorithm, and shown that it accurately identifies the long-lived conformations of a molecule without the construction of a MSM. In particular, the CNN algorithm identified the center of a long-lived conformation as a cluster but categorized the data points on the rims of the conformation as noise points. This is precisely the property of a good core set. The study additionally showed that geometric cluster algorithms which base their cluster criterion purely on a distance to a cluster medoid cannot reliably identify the long-lived conformations of a molecule. We thus propose to use a density-based cluster algorithm to identify the cores for a core-set model. We test three different density-based cluster algorithms: our own CNN algorithm,[26] the DBSCAN algorithm,[27] and the Jarvis-Patrick algorithm[28] which is implemented in the MD simulation package GROMACS.[29] Cores with different probability densities are identified by a hierarchical clustering procedure in which the density parameters of the cluster algorithm are iteratively re-adjusted. The metastability of the cores is ensured by slightly relaxing a mapping parameter in the construction of the milestoning processes. We test this approach on a two-dimensional model system and on the alanine dipeptide. Then we use it to construct a core-set model of a 14-residue peptide, which forms several different hairpin structures as well as a wide range of random coil structures. Highly accurate kinetic models for this type of peptides are notoriously difficult to construct because the peptides have a vast accessible conformational space with only marginally metastable conformations. Moreover, different hairpin-conformations are structurally very similar but are different from a chemical point of view. Our core-set model resolves two quickly interconverting hairpin structures which only differ by a register-shift in their hydrogen bond pattern.

II. THEORY

For the convenience of the reader, we summarize the derivation of the core-set models in Sections II A–II C. For a detailed discussion see Refs. 20–23 and 30. The density-based cluster algorithms are introduced in Section II D.

A. Molecular dynamics

The state space Ω of a molecular system contains all position and momentum coordinates of the system. We assume that the molecular system is in contact with a thermal bath and that the dynamics in this state space is ergodic, Markovian, and time-homogeneous. This ensures that the molecular-dynamic process samples a unique stationary probability density $\pi(\mathbf{x})$ with $\mathbf{x} \in \Omega$. We furthermore assume that the dynamic process is reversible with respect to $\pi(\mathbf{x})$. A realization of such a process $\mathbf{x}_t \in \Omega$, a so-called trajectory, can be obtained from thermostatted molecular dynamics simulations.

Next, consider an ensemble of identical molecular systems which are distributed according to some initial probability density $p_{t=0}(\mathbf{x})$ which differs from the stationary probability density, i.e., $p_{t=0}(\mathbf{x}) \neq \pi(\mathbf{x})$. As the molecular dynamic processes of each of the systems evolve, the ensemble probability density changes and gradually relaxes towards the stationary probability density: $\lim_{t\to\infty} p_t(\mathbf{x}) = \pi(\mathbf{x})$. The time-evolution of the probability density is governed by a propagator $\mathcal{P}(\tau)$,[31,32]

$$p_{t+\tau}(\mathbf{y}) = \mathcal{P}(\tau)p_t(\mathbf{x}) = \int p(\mathbf{x},\mathbf{y};\tau)p_t(\mathbf{x})d\mathbf{x}, \tag{1}$$

where the transition density $p(\mathbf{x},\mathbf{y};\tau)$ represents the conditional probability density of finding the molecular system at time $t+\tau$ in the state $\mathbf{y}$, given that it has been in $\mathbf{x}\,d\mathbf{x}$ at time t. For practical reasons, one however does not use the propagator for the construction of a Markov state model but the closely related transfer operator $\mathcal{T}(\tau)$,[31,32]

$$u_{t+\tau}(\mathbf{y}) = \mathcal{T}(\tau)u_t(\mathbf{x}) = \frac{1}{\pi(\mathbf{y})}\int p(\mathbf{x},\mathbf{y};\tau)\pi(\mathbf{x})u_t(\mathbf{x})d\mathbf{x} \tag{2}$$

with

$$\begin{aligned} p_t(\mathbf{x}) &= \pi(\mathbf{x})u_t(\mathbf{x}),\\ u_t(\mathbf{x}) &= \pi^{-1}(\mathbf{x})p_t(\mathbf{x}). \end{aligned} \tag{3}$$

The transfer operator transports functions $u_t(\mathbf{x})$ in time. As time goes to infinity, $u_t(\mathbf{x})$ converges to a constant function: $\lim_{t\to\infty} u_t(\mathbf{x}) = \lim_{t\to\infty}\pi^{-1}(\mathbf{x})p_t(\mathbf{x}) = 1$, independent of the stationary probability distribution of the system.

The transfer operator is self-adjoint[17,33] with respect to a weighted scalar product

$$\langle u|v\rangle_\pi = \int_\Omega u(\mathbf{x})\pi(\mathbf{x})v(\mathbf{x})d\mathbf{x}. \tag{4}$$

Hence its eigenvalues $\lambda_k(\tau)$ and eigenfunctions $r_k(\mathbf{x})$ are real-valued. The eigenfunctions form a basis of Ω. Furthermore, its eigenvalues lie in the interval $\lambda_k(\tau) \in [-1,1]$.[32,33] As a consequence the time-evolution of the probability density can be formulated as a superposition of the eigenfunctions with time-dependent coefficients[32,34–36]

$$p_{t=n\tau}(\mathbf{x}) = \sum_{k=1}^{\infty} a_k\lambda_k^n(\tau)r_k(\mathbf{x}) \approx \sum_{k=1}^{N} a_k\lambda_k^n(\tau)r_k(\mathbf{x}). \tag{5}$$

Since $|\lambda_k(\tau)| \leq 1$, the coefficients decay exponentially and the slow dynamics can be approximated by a superposition of the dominant first N eigenvectors. These dominant eigenvalues and eigenvectors also contain a wealth of information on the dynamics of the individual system.[32,35,36] We are thus interested in finding the dominant eigenvalues and eigenvectors of the transfer operator by solving

$$\mathcal{T}(\tau)r_k(\mathbf{x}) = \lambda_k(\tau)r_k(\mathbf{x}). \tag{6}$$

Unfortunately, Eq. (6) cannot be solved analytically for any realistic molecular system. One has to resort to approximation techniques which involve a discretization of the transfer operator.[33,37]

B. Discretization of the transfer operator

The eigenfunctions are approximated by expanding them in a finite basis $\{\psi_i(\mathbf{x})\}_{i=1}^n$,

$$r(\mathbf{x}) \approx \sum_{i=1}^{n}\tilde{c}_i\psi_i(\mathbf{x}), \tag{7}$$

where the basis functions span a subspace of the full molecular state space $D \subset \Omega$,

$$D := \operatorname{span}\{\psi_1,\ldots,\psi_n\}. \tag{8}$$

The expansion coefficients $\mathbf{c}^\top = (c_1,c_2,\ldots,c_n)$ in Eq. (7) can be obtained by a discretization of Eq. (6),[20,21]

$$\tilde{\mathbf{c}}^\top\mathbf{P}(\tau)\mathbf{M}^{-1} = \lambda\tilde{\mathbf{c}}^\top, \tag{9}$$

with

$$\mathbf{P}(\tau) : P_{ij}(\tau) = \frac{\langle\chi_i|\mathcal{T}(\tau)\chi_j\rangle_\pi}{\langle\chi_i,\mathbb{1}\rangle_\pi} \tag{10}$$

and

$$\mathbf{M} : M_{ij} = \frac{\langle\chi_i|\chi_j\rangle_\pi}{\langle\chi_i|\mathbb{1}\rangle_\pi}. \tag{11}$$

The functions $\{\chi_i(\mathbf{x})\}_{i=1}^n$ are scaled with respect to the basis set $\{\psi_i(\mathbf{x})\}_{i=1}^n$ as

$$\psi_i(\mathbf{x}) = \frac{\chi_i(\mathbf{x})}{\langle\chi_i|\mathbb{1}\rangle_\pi} \Leftrightarrow \chi_i(\mathbf{x}) = \langle\chi_i|\mathbb{1}\rangle_\pi\chi_i(\mathbf{x}). \tag{12}$$

This discretization is equivalent to the Galerkin discretization of the transfer operator or the variational approach to molecular dynamics,[17–19,37] in which matrices $\mathbf{S} : S_{ij} = \langle\chi_i|\chi_j\rangle_\pi$ and $\mathbf{C}(\tau) = C_{ij}(\tau) = \langle\chi_i|\mathcal{T}(\tau)\chi_j\rangle_\pi$ appear (see the Appendix). Given the analytical form of the basis functions $\{\chi_i(\mathbf{x})\}_{i=1}^n$, the matrix elements S_{ij} and $C_{ij}(\tau)$ can be estimated from an MD simulation of the underlying dynamic process $\mathbf{x}_t$. Conventional Markov state models[13,32,38,39] can be regarded as a special case of the variational approach in which the basis functions are characteristic functions which partition the state space Ω into discrete states.[18] In the core-set approach, the basis functions $\{\chi_i(\mathbf{x})\}_{i=1}^n$ are committor functions (see Section II C). These functions are, however, typically not known analytically. Thus, we are in the somewhat difficult situation of trying to find a matrix representation of an operator which is not known analytically with respect to a basis set which is not known analytically either. Fortunately, one can show that the matrix elements of $\mathbf{P}(\tau)$ and $\mathbf{M}$ can be estimated from milestoning processes derived from MD trajectories.[21–25]

C. Core sets, committor functions, milestoning processes

We define n disjoint core sets $B_1,B_2,\ldots,B_n$ with $B_i \cap B_j = \emptyset$ for all $i \neq j$. In contrast to the states in conventional Markov state models, these core sets do not fully partition the state space $\bigcup_{i=1}^n B_i \subset \Omega$, i.e., there are regions $I = \Omega \setminus \bigcup_{i=1}^n B_i$ in the state space which are not assigned to any of the core sets (Fig. 1(a)). Associated to each core set B_i is a committor function $q_i(\mathbf{x})$, which is defined by the following equations:

$$\begin{cases} \mathcal{L}q_i(\mathbf{x}) = 0 & \mathbf{x} \in I\\ q_i(\mathbf{x}) = 1 & \mathbf{x} \in B_i\\ q_i(\mathbf{x}) = 0 & \mathbf{x} \in B_j\ \forall j \neq i \end{cases}, \tag{13}$$

where $\mathcal{L}$ is the generator of the dynamics

$$\frac{d}{dt}p_t(\mathbf{x}) = \mathcal{L}p_t(\mathbf{x}) \tag{14}$$

associated to the propagator (Eq. (1)) by

$$p_{t+\tau}(\mathbf{x}) = \mathcal{P}(\tau)p_t(\mathbf{x}) = \exp(\tau\mathcal{L})\,p_t(\mathbf{x}). \tag{15}$$

The committor function $q_i(\mathbf{x})$ can be interpreted as the probability that the trajectory which is at position $\mathbf{x}_t$ at time t will reach the set B_i first before it reaches any of the other sets $B_{j\neq i}$. Thus, $q_i(\mathbf{x})$ assumes the value one within B_i, the value zero within any other core set and values between zero and one in the space in between the core sets (Eq. (13)). Fig. 1(b)

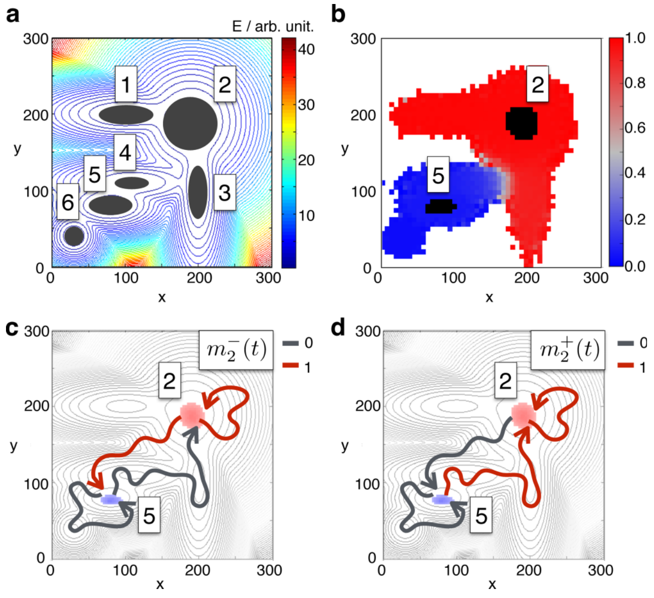


FIG. 1. Core sets, committor function, and milestoning processes. (a) Potential energy function with six minima. The gray areas represent the core sets for each minimum. (b) Committor function of core set 2 ($q_2(x, y)$) using only core set 2 and 5 for the definition of the committor function. (c) Backward milestoning process for core set 2. (d) Forward milestoning process for core set 2.

shows the committor function $q_2(\mathbf{x})$ in the two-dimensional potential energy function of Fig. 1(b), where we have however only used two core sets, B_2 and B_5, to define the committor function. To solve Eq. (13) and to obtain the committor functions, one needs an analytical representation or a matrix representation of the generator or, alternatively, of the transfer operator.

Alternatively, one can define backward and forward milestoning processes, $m_i^-(t)$ and $m_i^+(t)$, for each of the core sets. Milestoning processes are projections of the trajectory $\mathbf{x}_t$ that depend on the history and the future of the trajectory. They assume the value 1 whenever the trajectory is in core set B_i and the value 0 whenever the trajectory is in one of the other core sets. In the intervening space I, the backward milestoning process also assumes the value 1 if the last core set the trajectory has visited was B_i, i.e., the process assumes the value 1 as soon as it hits B_i and only switches to 0 when it reaches another core set (Fig. 1(c)),

$$m_i^-(t) = \begin{cases} 1 & \text{if } \mathbf{x}_t \in B_i \\ 1 & \text{if } \mathbf{x}_t \in I \text{ and last came from } B_i \\ 0 & \text{otherwise} \end{cases}. \tag{16}$$

The forward milestoning process assumes the value 1 whenever the next core set to be visited by the trajectory is B_i (Fig. 1(d)),

$$m_i^+(t) = \begin{cases} 1 & \text{if } \mathbf{x}_t \in B_i \\ 1 & \text{if } \mathbf{x}_t \in I \text{ and will go to } B_i \text{ next} \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

The matrix elements of $\mathbf{M}$ and $\mathbf{P}(\tau)$ (Eqs. (10) and (11)) can be estimated as the (time-lagged) correlation functions between the backward and forward milestoning processes

$$P_{ij}(\tau) = \frac{\langle q_i | \mathcal{T}(\tau) q_j \rangle_\pi}{\langle q_i | \mathbb{1} \rangle_\pi} = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau} m_i^-(t) m_j^+(t+\tau) \tag{18}$$

and

$$M_{ij} = \frac{\langle q_i | q_j \rangle_\pi}{\langle q_i | \mathbb{1} \rangle_\pi} = \frac{1}{T-\tau} \sum_{t=0}^{T-\tau} m_i^-(t) m_j^+(t), \tag{19}$$

where $\mathbf{P}(\tau)$ and $\mathbf{M}$ are defined with respect to the basis set of the committor functions $\{q_i(\mathbf{x})\}_{i=1}^n$. Both matrices are stochastic matrices. The matrix elements M_{ij} represent the probability that the process will visit core B_j next, given that the last core which has been visited was B_i. The matrix elements $P_{ij}(\tau)$ represent the probability that, after an interval $[t, t+\tau]$, the process will visit B_j next, given that the last core which has been visited at time t was B_i. Possible visits to other cores in the interval $[t, t+\tau]$ do not affect $P_{ij}(\tau)$.[37] That is,

$$\begin{aligned} M_{ij} &= \mathbb{P}[m_j^+(t) = 1 | m_i^-(t) = 1], \\ P_{ij}(\tau) &= \mathbb{P}[m_j^+(t+\tau) = 1 | m_i^-(t) = 1]. \end{aligned} \tag{20}$$

The core-set discretization yields a small discretization error if the core sets are sufficiently metastable, such that the process leaves the intervening space I on a faster time scale than the fastest process of interest. This condition is difficult to test. As a proxy we therefore ensure that the largest element in each row of $\mathbf{M}$ is the diagonal element ($M_{ii} > M_{ij}\, \forall j$) or even that $\mathbf{M}$ is diagonally dominant ($M_{ii} > \sum_{j \neq i} M_{ij}$).

D. Density-based cluster algorithms

Let X be a data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where each data point $\mathbf{x}_i$ is a point in a high-dimensional space. A cluster $C = \{\mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l, \ldots\}$ is a subset of data points which are located in a region with high data point density and which is separated from other clusters by regions with low data point density (Fig. 2(a)). A direct estimate of the data-point density would involve a discretization of the data space into volume elements and counting the number of data points per volume element, which is only feasible for low-dimensional spaces. Density-based cluster algorithms circumvent the direct estimation of the data-point density by determining whether two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are density-reachable. Typically, a data point $\mathbf{x}_i$ becomes a member of a cluster $C = \{\mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l, \ldots\}$ if it is density-reachable from at least one of the cluster members. The cluster is expanded until none of the so far unassigned data points are density-reachable from any of the cluster members. A generic algorithm for a density-based cluster algorithm is

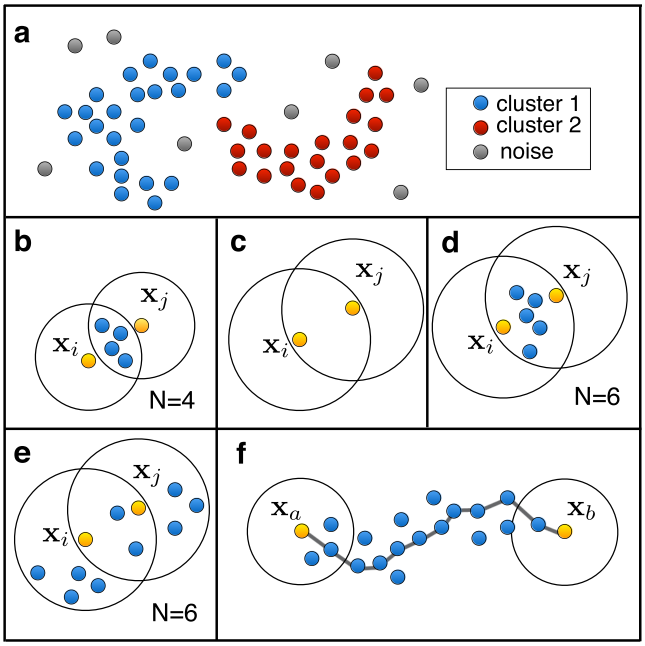


FIG. 2. Density-based clustering. (a) Data set with non-convex clusters and noise points. (b)-(e) Various criteria to decide whether $\mathbf{x}_i$ and $\mathbf{x}_i$ are density reachable (see Section II D): (b) $\mathbf{x}_i$ and $\mathbf{x}_j$ share at least N neighbors (blue points), (c) $\mathbf{x}_i$ and $\mathbf{x}_j$ are in each others neighbor list, (d) $\mathbf{x}_i$ and $\mathbf{x}_j$ share at least N neighbors (blue points) and are in each other's neighbor list, (e) $\mathbf{x}_i$ and $\mathbf{x}_j$ are in each other's neighbor list and each of the data points has at least N neighbors. (f) $\mathbf{x}_a$ and $\mathbf{x}_b$ are density connected but not density reachable.

ALGORITHM 1. Density-based cluster algorithm

```
while clustering is not complete do
    initialize cluster $C$
    while unassigned data points can be added to $C$ do
        loop over unassigned data points $\mathbf{x}_i$ and cluster members $\mathbf{x}_j$
        if $\mathbf{x}_i$ and $\mathbf{x}_j$ meet a DensityCriterion then add $i$ to $C$
    end
    if $C$ has more than $M_{\min}$ members then
        add $C$ to list of clusters
    else
        classify $C$ and all remaining unassigned points as noise
        clustering is complete
    end
end
```

The criterion which determines whether two data points are density-reachable varies from algorithm to algorithm. Usually, for each data point a list of neighbors within a neighborhood R is generated and the number of shared neighbors determine whether two data points are density-reachable. For example, two data points can be density-reachable from each other if

1. they share at least N neighbors (Fig. 2(b)), or
2. they are in each other's neighbor list (Fig. 2(c)), or
3. if 1 and 2 are fulfilled (Fig. 2(d)), or
4. if 2 is fulfilled and each of the data points has at least N neighbors (Fig. 2(e)).

If R is a radius around that data point, condition 1 is a rough estimate of the data point density as the number of data points in the overlap region of the two neighborhoods. The estimate is however not very precise since the actual volume of the overlap region is never determined. If the neighborhood parameter R represents a radius, the maximum distance between density connected data points is $d_{\max}(\mathbf{x}_i,\mathbf{x}_j) = 2R$. Condition 2 reduces this distance to $d_{\max}(\mathbf{x}_i,\mathbf{x}_j) = R$. Note that two cluster members $\mathbf{x}_a$ and $\mathbf{x}_z$ are not necessarily directly density-reachable from each other but that they are at least density-connected. Density-connected means that there is a sequence of data points $\mathbf{x}_1,\mathbf{x}_2,\ldots\mathbf{x}_n$ (with $\mathbf{x}_1 = \mathbf{x}_a$ and $\mathbf{x}_n = \mathbf{x}_z$) in which each data point $\mathbf{x}_{i+1}$ is density-reachable from the previous data point $\mathbf{x}_i$ ((Fig. 2(f))). Data points which cannot be assigned to any cluster are called noise data points. We compare three different density-based cluster algorithms: the Common-Nearest-Neighbor-algorithm (CNN),[26] Density-Based Spatial Clustering of Applications with Noise (DBSCAN),[27] and the Jarvis-Patrick-algorithm (JP).[28] All three algorithms can identify clusters of arbitrary shape (Fig. 2(a)) and can distinguish noise points.

In the **CNN algorithm**,[26] the neighborhood parameter R is a radius. Two data points are density-reachable and belong to the same cluster if they fulfill condition 1. Optionally, condition 3 can be applied which reduces the run time of the algorithm.

Also in the **DBSCAN algorithm**,[27] the neighborhood parameter R is a radius. However, two data points are density-reachable if condition 4 is fulfilled. This amounts to an estimation of the data point density in the neighborhood of each data point rather than in the overlap region between pairs of data points. Applying only this condition yields clusterings with many noise points and very few cluster members. The algorithm therefore differentiates between core points, border points, and noise points. Data points which have at least N nearest neighbors are called core points. Border points are data points which are not core points (i.e., have less than N neighbors) but are closer than R to at least one of the core points in the data set (i.e., are members of the neighborlist of at least one core point). Border points can be assigned to only one cluster. This definition introduces an ambiguity. For example, if a border point $\mathbf{x}_b$ has two core points $\mathbf{x}_i$ and $\mathbf{x}_j$ in its neighbor list, and the cluster of $\mathbf{x}_i$ and $\mathbf{x}_j$ is not density-connected, then $\mathbf{x}_b$ can be assigned to either the cluster of $\mathbf{x}_i$ or the cluster of $\mathbf{x}_j$. But the two clusters cannot be joined via $\mathbf{x}_b$. To which cluster $\mathbf{x}_b$ is assigned depends on the implementation. Data points which are neither core points nor border points are called noise points.

In the **JP algorithm**,[28] the neighborhood parameter R does not represent a radius but a predefined number of nearest neighbors. This results in a fixed size of the neighborlist with a variable volume of the neighborhood. Two data points are density-reachable if they fulfill condition 1 or, optionally, condition 3. In the original publication, the two data points whose neighborhoods are compared did not add to the count of shared neighbors, whereas we decided to include them in this count (to be consistent with the CNN algorithm). That is, our implementation executed with neighborhood parameter R and N nearest neighbors yields the same results as the original algorithm executed with R and $N-2$.

1. Implementation

All three cluster programs read a distance matrix and extract from this matrix a neighbor list for each data point. The neighbor list can be further simplified. In the CNN algorithms, all data points which have less than N neighbors are immediately classified as noise points since they certainly cannot share N neighbors with any other data points. Furthermore, these noise points are removed from the neighbor lists of all other data points. If condition 3 is enforced in the CNN algorithm or the JP algorithm, each data point can only be density-reachable from a data point in its neighbor list. This reduces the search-time for potentially density-reachable points drastically. If condition 3 is not enforced, we keep a second list of potentially density-reachable data points, i.e., all data points within $2R$. Since in the DBSCAN algorithm only core-points (i.e., points which have at least N neighbors) can be used to expand a cluster, the neighbor lists of all non-core points are deleted. In contrast to the CNN algorithm, the non-core points are however not removed from the neighbor list of the core points since they still can enter a cluster as a border point. Finally, we initialize the clustering on the data point with the highest number neighbors, which speeds up the run-time of the algorithm and ensures the reproducibility of the results for DBSCAN. All programs for the density-based clustering and the construction of the core-set model are reported in the supplementary material.

2. Choice of parameters

The parameters for the CNN and DBSCAN algorithm are chosen following the approach in Ref. 26. Histogram of the distances in the distance matrix is plotted and R is set to the value of the first maximum of this distribution. The parameter N is varied until clusters of sufficient size are obtained. In the JP algorithm R is varied until clusters of sufficient size are obtained.

III. METHODS

A. Two-dimensional data set

The two-dimensional Boltzmann-distribution $p(x,y) \propto Z(\beta)^{-1}\exp(-\beta V(x,y))$ $(Z(\beta) = \int \exp(-\beta V(x,y))dxdy$ is the partition function and we set $\beta = 2$, Fig. 1(a)) was sampled using a Markov-Chain-Monte-Carlo algorithm.[40] The potential energy function (Fig. 3(a)) was

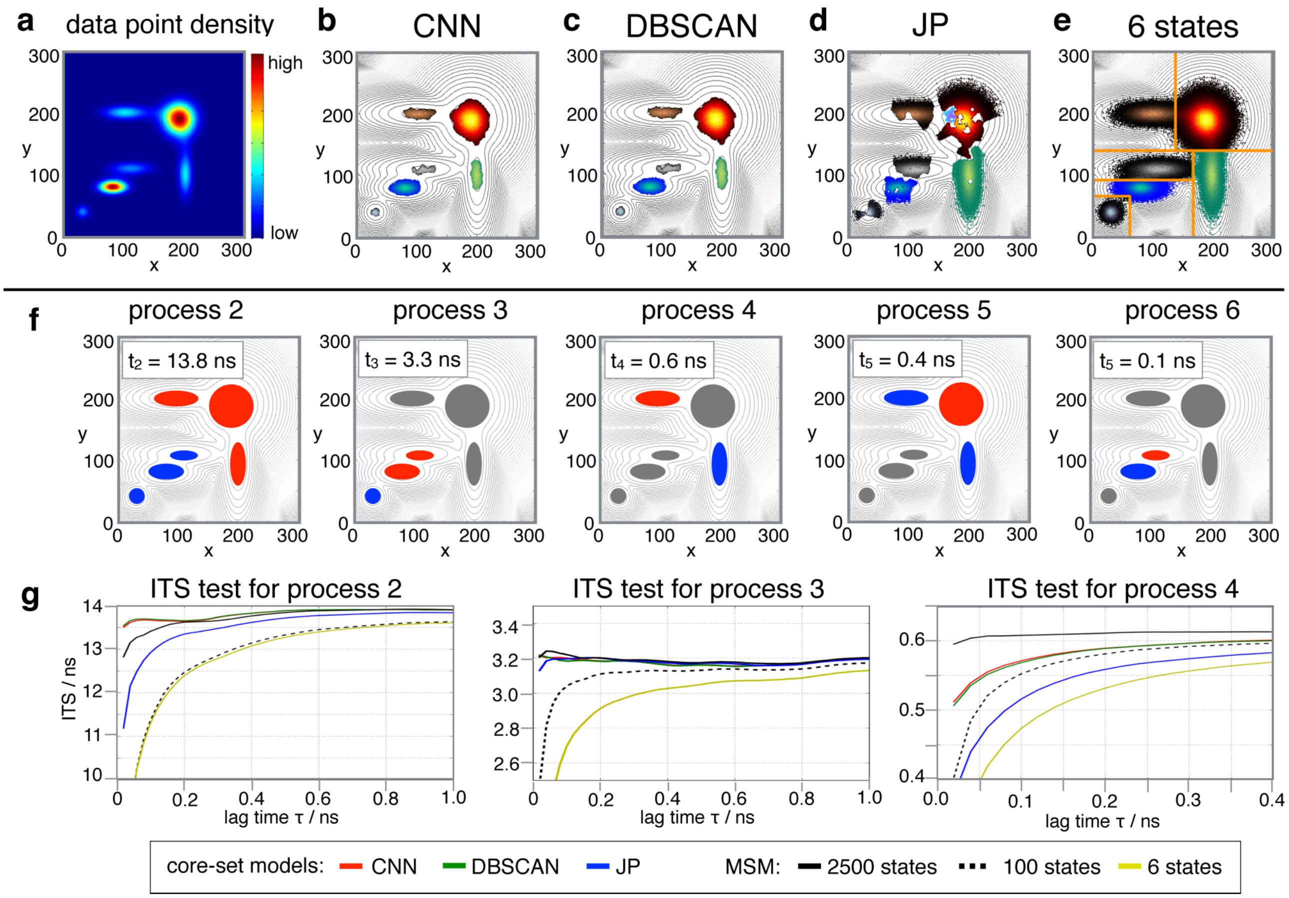


FIG. 3. Core-set models for the dynamics in the two-dimensional energy surface. (a) Stationary probability density. (b)-(d) Cluster results for the density based cluster algorithms CNN, DBSCAN, and JP. For JP, only the largest seven out of 19 clusters are shown. (e) Crisp discretization into six states. (f) Slow kinetic processes as identified by the core-set models and the MSMs. Negative values of the corresponding eigenfunction are shown in blue, positive values in red, and values close to zero in gray. (g) Implied time scale test for processes 2, 3, and 4.

TABLE I. Parameters of the two-dimensional potential energy function (Eq. (21)).

i	c_1	μ_{xi}	σ^2_{xi}	μ_{yi}	σ^2_{yi}
1	1	30	100	40	85
2	2	80	400	80	85
3	2	190	550	190	550
4	1.2	100	1000	200	100
5	1.2	200	100	100	1000
6	1	110	1000	110	85

$$V(x,y) = -\log\left[\sum_{i=1}^{6} c_i \exp\left(-\frac{(x-\mu_{xi})^2}{2\sigma^2_{xi}}\right)\exp\left(-\frac{(y-\mu_{yi})^2}{2\sigma^2_{yi}}\right)\right] \tag{21}$$

with parameters reported in Table I. Trial positions $\mathbf{r}_{\text{trial}} = (x_{\text{trial}}, y_{\text{trial}})$ were drawn from a two-dimensional normal distribution $\mathcal{N}(\mathbf{r}_i, \sigma = 20)$ centered at the position $\mathbf{r}_i = (x_i, y_i)$ of the current iteration and accepted with a probability $p_{\text{acc}} = \min\{1, \exp(-\beta \cdot \Delta V)\}$, where $\Delta V = V(x_{\text{trial}}, y_{\text{trial}}) - V(x_i, y_i)$. For the kinetic analysis, the iteration index was interpreted as the time of the trajectory. The sampling was performed over 10^7 iterations and the positions were saved to file every 10th iteration. The average acceptance rate was 0.83. The full program is reported in the supplementary material.

B. Molecular dynamics simulations

1. Terminally capped alanine

The MD simulations of the terminally capped alanine have been reported previously.[41] The simulations were performed in an NVT ensemble in explicit water using the GROMACS simulation package 4.5.5,[29] the AMBER ff99SB-ILDN force field,[42] and the TIP3P water model.[43] The temperature was restrained to 300 K using the V-rescale thermostat[44] ($\tau_t = 0.01$ ps). A cubic box with a box length of 2.72 nm was used and periodic boundary conditions were applied in all three directions. The equations of motion were integrated using the leap-frog integrator with a time step of 2 fs. Covalent bonds to hydrogen atoms were constrained using the LINCS algorithm[45] (lincs_iter = 1, lincs_order = 4). Lennard-Jones interactions were cut off at 1 nm. The electrostatic interactions were calculated using a Particle-Mesh Ewald (PME) summation[46] with a real space cutoff of 1 nm, a Fourier grid spacing of 0.1, and an interpolation order of 4. Solute coordinates were written to file every 1 ps. Five independent trajectories of 200 ns each were produced, yielding a total simulation time of 1 μs.

2. Hairpin peptide

For the hairpin peptide RGKITVNGKTYEGR we have performed all-atom molecular dynamics simulations in explicit water in an NVT ensemble. We used charged termini, protonated the arginine and lysine residues, and deprotonated the glutamic acid residue. The structure was energy minimized (emtol = 100.0 (kJ/mol)/nm, nsteps = 5000) and solvated in a cubic box with a box length of 7.08 nm. Three chlorine anions were added to obtain an uncharged box. The simulation box was energy minimized and equilibrated for 100 ps. From the equilibration run, we extracted eight starting structures for the production runs. The same force field, water model, and simulation parameters as for Ac-A-NHMe were used, except for the time constant in the V-rescale thermostat which was set to $\tau_t = 0.1$ ps. Solute coordinates were written to file every 1 ps. We obtained eight trajectories with a length of 860–980 ns with a total simulation length of ca. 7.4 μs.

C. Density based clustering and core set models

The clustering was performed on a subset of all frames in the simulated trajectories (Table II), which were extracted at regular intervals. In the two-dimensional data set, the Euclidean distance served as a distance measure between pairs of data points. In the molecular systems, the pairwise distance was calculated as the backbone RMSD between two structures i and j after rotational and translational fit. We used *pyRMSD*[47] (Version 4.2.1) in combination with the QCP-OMP-Calculator[48] for the calculation of the RMSD values. The clustering was performed as described in Section II D with parameters as reported in Table II. The minimum number of members per cluster were 50 for the two-dimensional data set, 20 for Ac-A-NHMe, and 20 for the hairpin peptide.

For the milestoning trajectories, the MD trajectories were mapped onto the n clusters by checking whether a given frame $\mathbf{x}_t$ would qualify as a member of the cluster C_i based on the cluster algorithm and the cluster parameters which generated the cluster. That is, for clusters obtained using a hierarchical cluster procedure, the parameters for the mapping varied from cluster to cluster. For the hairpin peptide, we used a relaxed mapping criterion by increasing the cluster parameter R associated to each cluster by 10% and by reducing the parameter N by one. Note that the clusters were not increased during the mapping, but for each frame $\mathbf{x}_t$ the density criterion for a cluster C_i was

TABLE II. Cluster parameters and number of frames used for the clustering.

Parameter	CNN	DBSCAN	JP
	2D model		
# frames	10 000	10 000	10 000
R/arb. unit	4	4	30
N	20	25	24
	Alanine dipeptide		
# frames	5 005	5 005	5 005
R/nm	0.01	0.01	4
N	20	30	2
	β-hairpin peptide		
# frames	19 950	…	…
R/nm	0.2-0.08	…	…
N	2	…	…

evaluated based on the same set of members of C_i. This yielded a cluster trajectory $c_t \in [0, 1, 2, \ldots, n]$ which then was converted to n forward and n backward milestoning trajectories (Eqs. (16) and (17)). The elements of matrices $\mathbf{P}(\tau)$ and $\mathbf{M}$ were estimated as time-lagged correlation functions of these milestoning processes (Eqs. (18) and (19)). The PCCA+ analysis[49] of the matrix $\mathbf{P}(\tau)\mathbf{M}^{-1}$ was performed using the MSM analysis package EMMA[50,51] and a threshold of 0.6 for the "fuzzy" PCCA-cluster assignment.

D. Markov state models

For the conventional Markov state models, we constructed microstate trajectories and estimated the transition matrix using in-house scripts. The microstate definitions are shown in Figs. 3(e) and 4(d). Additionally, we used regular discretizations with $10 \times 10 = 100$ microstates and with $50 \times 50 = 2500$ microstates for the two-dimensional data set. For Ac-A-NHMe, a regular discretization with $36 \times 36 = 1296$ microstates was constructed in the space of the ϕ- and ψ-backbone torsion angle. We used a moving lag time window to count the transitions and enforced detailed balance by symmetrizing the count matrix. The lag time τ was chosen based on the implied time scale test,[38] which tests whether the implied time scale

$$t_i = \frac{-\tau}{\ln[\lambda_i(\tau)]} \tag{22}$$

does not vary as a function of the lag time τ. The dominant processes were characterized by a PCCA+ analysis[49] using the MSM analysis package EMMA.[50,51]

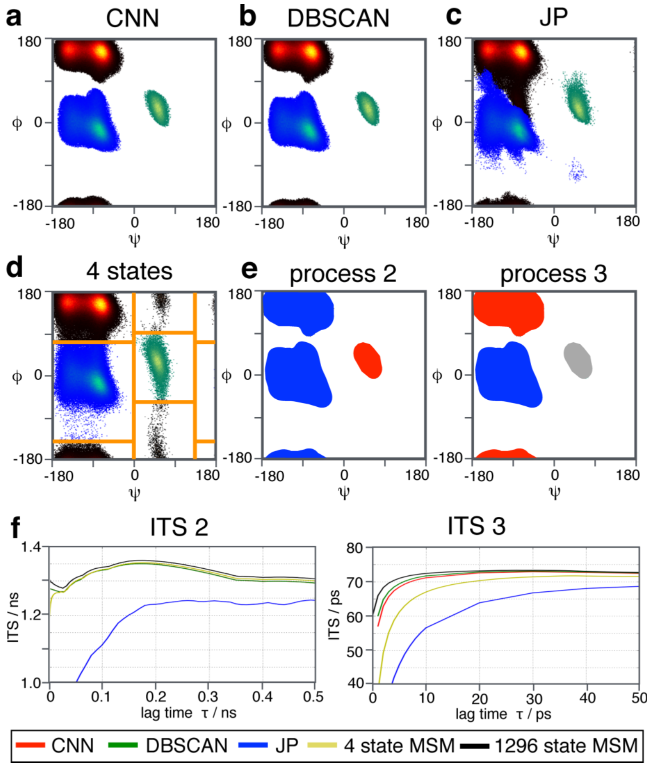


FIG. 4. Core-set models for alanine dipeptide (Ac-A-NHMe). (a)-(c) Cluster results for the density based cluster algorithms CNN, DBSCAN, and JP shown in the Ramachandran plane of alanine dipeptide. (d) Crisp discretization into four states. (e) Slow kinetic processes as identified by the core-set models and the MSMs. Negative values of the corresponding eigenfunction are shown in blue, positive values in red, and values close to zero in gray. (f) Implied time scale test for processes 2 and 3.

IV. RESULTS

A. Two-dimensional potential energy surface

Fig. 1 illustrates the performance of the three density-based cluster algorithms on a two-dimensional potential energy surface with six minima. We sampled the dynamics in this potential energy landscape using a Metropolis Markov chain Monte Carlo algorithm.[40] Each of the wells has a different minimum energy such that the data point density varies from well to well (Fig. 3(a)). Nonetheless, the six minima were identified by each of the three cluster algorithms using only a single parameter setting (Figs. 3(b)-3(d)). That is, we did not need to apply hierarchical clustering to account for the variation in data point density. DBSCAN and CNN found exactly six clusters and yield almost identical results. Both algorithms identified tight but well-defined cores in each of the minima (Figs. 3(b) and 3(c)). The JP algorithm extracted 19 clusters of which the seven largest are shown in Fig. 3(d). These JP clusters are larger than those identified by DBSCAN and CNN and have fuzzy boundaries. Also note that minimum 2 is split into two clusters.

We used the clusters as core sets and compared the resulting core-set models with conventional MSMs. Two core-set models were constructed using six cores from the CNN and DBSCAN results. The core-set model based on the JP-results had 19 cores. One conventional MSM consisted of six states which we chose manually to optimally represent the six minima (Fig. 3(e)). We also included two additional MSMs with a regular discretization (10 bins per axis → 100 states and 50 bins per axis → 2500 states). All six models identified the same slow processes and assigned comparable lag times to the processes (Figs. 3(f) and 3(g)). We used the implied-timescale test[21,32,38] as an indicator for the discretization error. In this test, the discretization error is treated as negligible for lag times τ at which the implied scales are approximately constant with respect to the lag time. The 2500-state MSM showed the fastest convergence in all six processes, followed by the six-core core-set models based on the DBSCAN and CNN clustering. The JP-clustering outperformed the 100-state MSM in processes 2, 3, and 6, but yielded a relatively poor convergence for processes 4 and 5. (Data shown for processes 2, 3, and 4.) All three core-set models showed better convergence than the six-state MSM. The kinetic processes corresponding to the implied time scales are depicted in (Fig. 3(f)). Red areas denote positive signs in the dominant eigenvectors, blue areas denote negative signs, and grey areas correspond to values close to zero. The dominant eigenvectors represent the kinetic exchange between regions of opposite signs. Thus, process 2 represents the kinetic exchange across the largest barrier in the system which separates cores 1, 2, and 3 from cores 4, 5, and 6. Process 4 and 5 with similar implied time scales correspond to the equilibration within

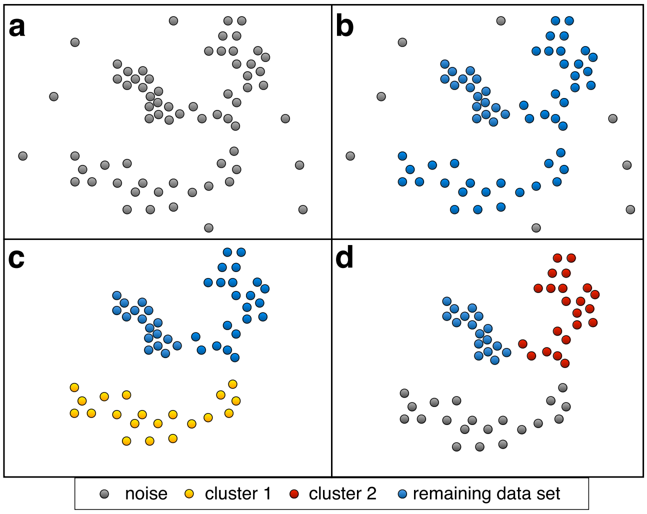
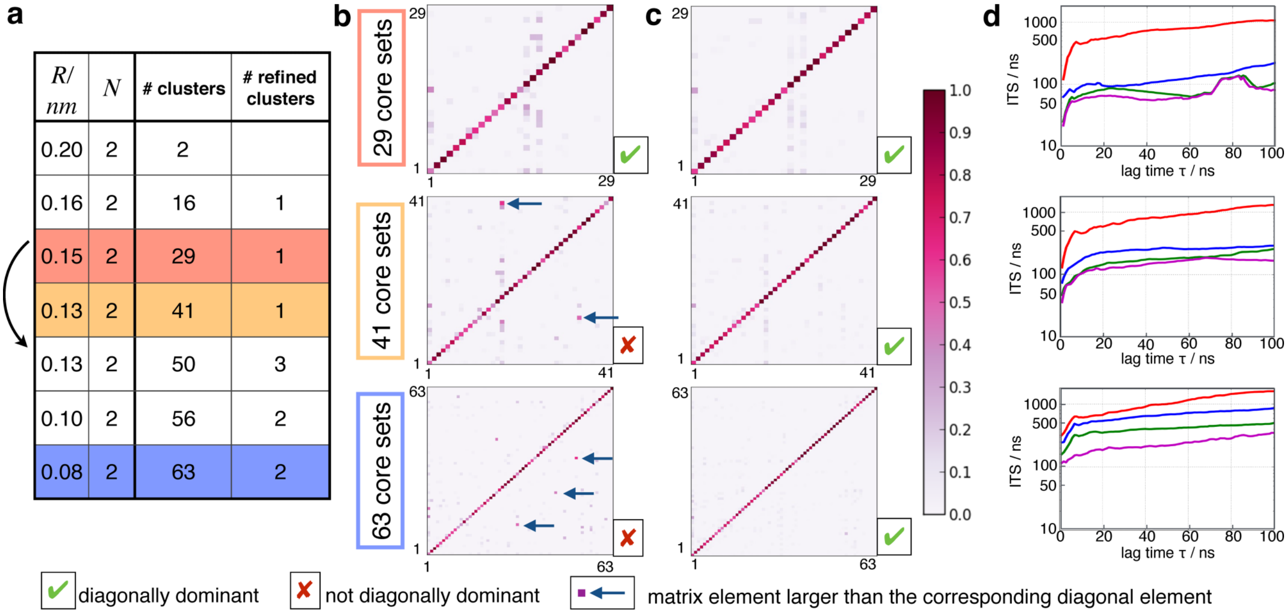


FIG. 5. Hierarchical clustering (see Section IV C). (a) Data set with clusters of different data point densities. (b) Noise points are identified. (c) and (d) The remaining data set is hierarchically subdivided into clusters.

cores 1, 2, and 3, while cores 4, 5, and 6 do not take part in these processes. The equilibration within cores 4, 5, and 6 is mediated by a relatively slow process (process 3, 3.3 ns) between the high-lying core 6 and the more populated cores 4 and 5, and by a faster kinetic exchange between cores 4 and 5 (process 6). The benchmark-test on the two-dimensional energy surface showed that, a core-set discretization with n cores identified by a density-based cluster algorithm yields a considerably lower discretization error than a conventional MSM with n optimally chosen states. In fact, the resolution of the regular discretization had to be increased to 2500 states to obtain a convergence which is comparable to a core-set model with six core sets.

B. Terminally capped alanine

As a molecular system with well-defined metastable states we chose alanine dipeptide (Ac-A-NHMe) which is a commonly used test system.[13,18,41] The slow dynamics of the molecule is well-captured by the dynamics of its ϕ- and ψ-backbone torsion angles (Ramachandran plane). Figs. 4(a)-4(c) show the cluster results for CNN, DBSCAN, and JP. All three algorithms identified three clusters corresponding to the α-helix conformation (blue cluster), the β-sheet conformation (orange cluster), and the L_α-helix conformation (green cluster). As in the 2D model, CNN and DBSCAN yielded very similar results with rather tight clusters, and the JP algorithm yielded larger clusters with less well defined boundaries. We used these clusters to define core-set models and compared the results to two conventional MSMs. The first MSM was constructed on four manually defined states shown in Fig. 4(d), the second MSM was constructed on a regular grid with 36 bins per torsion angle yielding 1296 states in total. The core set models based on the CNN and the DBSCAN clustering as well as both conventional MSMs identified two slow processes (Fig. 4(e)). Process 2 corresponds to a kinetic exchange between the L_α-helix conformation and the other two conformations with an implied time scale of ≈1.3 ns. Process 3 corresponds to a kinetic exchange between the α-helix conformation and the β-sheet conformation with an implied time scale of ≈70 ps. The implied-time scale test shows that the convergence (Fig. 4(f)) of these four models is similar. By contrast, the core-set model based on the JP clustering converges at considerably larger lag times. Moreover, the implied time scales converge to values which are slightly below the reference value of the 1296-state MSM (Fig. 4(f)) and hence do not fully reproduce the reference model.



$R/$ nm	N	# clusters	# refined clusters
0.20	2	2	
0.16	2	16	1
0.15	2	29	1
0.13	2	41	1
0.13	2	50	3
0.10	2	56	2
0.08	2	63	2

FIG. 6. Hierarchical clustering of a β-hairpin peptide and mass matrices of the corresponding core-set models. (a) Sequence of cluster parameters in the hierarchical clustering procedure. (b) Matrices **M** at different levels of clustering. Arrows highlight matrix elements which cause some matrices to be not diagonally dominant. (c) Corresponding matrices **M** with relaxed mapping parameters. (d) Implied time scale test for the core-set models with relaxed mapping parameters. Red: process 2, blue: process 3, green: process 4, purple: process 5.

C. Core-set approach with hierarchical clustering

The results, so far, have shown (i) that, for systems with a small number of well-defined minima in the potential energy function, the CNN and the DBSCAN cluster algorithm reliably identify these minima and (ii) that core-set models based on these clusters show a better convergence in the implied time scale test than conventional MSMs of comparable size and are therefore more accurate. However, many biologically interesting molecules exhibit a large number of only marginally metastable states. Moreover, the probability densities associated to these metastable states can vary considerably. Thus, the challenge is to find a method that identifies states with different probability densities and to assure at the same time that the resulting cores are sufficiently metastable that the matrix **M** remains diagonally dominant.

We approached the first part of the problem by using a hierarchical clustering procedure in which in every round of clustering the parameters was re-adjusted to match the data point density in the remaining data set. Fig. 5(a) shows a data set with clusters of different data point densities which can be distinguished visually but not be separated with a single round of clustering. By clustering with parameters corresponding to a low data point density, noise points are identified (gray points in Fig. 5(b)) and removed from the data set. By readjusting the parameters to a higher data point density, the yellow cluster is split of from the data set (Fig. 5(c)). Re-clustering the blue data set with parameters corresponding to even higher data point densities subdivides the data set into two clusters, shown in red and blue, whereas applying the same parameters to the yellow data sets splits the cluster into noise points (Fig. 5(d)). In practice, one decreases the neighborhood parameter R in small intervals of ΔR while keeping the value N fixed. Reducing R at first leads to smaller clusters because data points at the rims are characterized as noise points. When R is further lowered, eventually one or more clusters are split into smaller clusters.

An example for a peptide with complex conformational dynamics and a multitude of only marginally metastable states is the 14-residue peptide RGKITVNGKTYEGR.[52] In solution, the peptide forms several different β-hairpin structures as well as a wide range of random coil structures. In our simulations, the peptide was about 45% folded and 55% unfolded. We applied the hierarchical cluster approach in combination with the CNN algorithm to a data set of 19 950 peptide structures. Starting from the initial parameter values ($R = 0.2$ nm and $N = 2$), we reduced R in intervals of $\Delta R = 0.01$ nm. The parameter values at which clusters were split is shown in Fig. 6(a). The initial clustering with parameters $R = 0.2$ nm and $N = 2$ yielded two clusters, of which one was further split into 15 smaller clusters by clustering with parameters $R = 0.16$ nm and $N = 2$. We sequentially split the largest cluster of each clustering with decreasing values of the neighborhood parameter R obtaining clusterings with 29 and 41 clusters. The clustering with 50 clusters was obtained by refining the three largest clusters from the clustering with 29 clusters with $R = 0.13$ nm. Further decreasing the neighborhood parameter to $R = 0.10$ nm and $R = 0.08$ nm subdivided the two largest clusters at each level and eventually led to a very fine clustering with 63 clusters. The minimum number of shared neighbors was kept constant at $N = 2$ throughout the hierarchical cluster analysis.

We used the clusterings with 29, 41, and 63 clusters for the construction of core-set models as described in Sec. III. It is important to point out that, when mapping the MD-trajectory onto the clusters, we did not use global values for the cluster parameters. Instead, a structure was assigned to a core if the structure met the density criterion which originally generated the cluster at the center of the core. That is, each cluster "attracted" frames according to its cluster parameters. The matrices **M** of the corresponding core-set models are shown in Fig. 6(b). The matrix **M** of the model with 29 cores was diagonally dominant. However, in the models with higher number of states, several states were not sufficiently metastable and generated off-diagonal matrix elements which were larger than the corresponding diagonal matrix element. The resulting matrices were not diagonally dominant and the discretization could not be used for a core-set model. We therefore relaxed the mapping criteria, i.e., we set $N = 1$ and

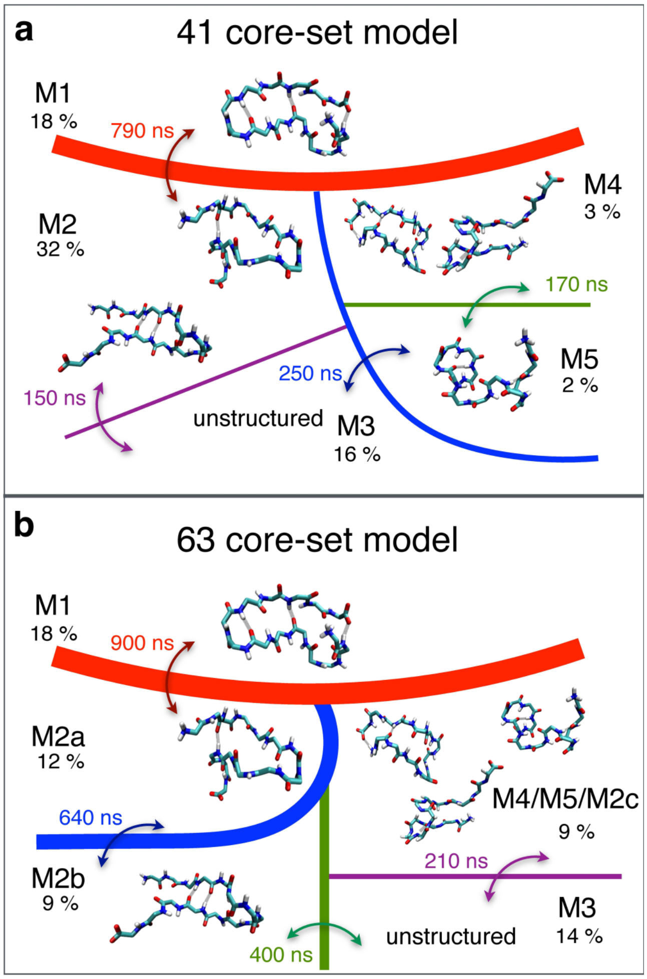


FIG. 7. Kinetic model of the β-hairpin peptide. (a) 41-core-set model at lag time $\tau = 40$ ns. (b) 63-core-set model at lag time $\tau = 40$ ns. The lines represent free-energy barriers as identified by a PCCA+ analysis of the core-set models. The associated implied time scales are shown in the same color as the barrier. Next to the index of each metastable set (M1–M5), the relative population of the set is shown in percentages. The depicted structures are example structures from each metastable-set.

increased each of the neighborhood parameters R by 10%, for these specific states. The relaxed mapping criterion should be chosen large enough that **M** becomes diagonally dominant and small enough that frames are not assigned to more than one core. The resulting matrices **M** are shown in Fig. 6(c). The off-diagonal matrix elements were much closer to zero than in Fig. 6(b) and indeed all three matrices were diagonally dominant. The models with the relaxed mapping parameters were analyzed further.

The implied time scales (Fig. 6(d)) in the resulting three core-set models were well converged at a lag time of about $\tau = 10$ ns and remained reasonably constant up to $\tau = 100$ ns. That is, the region in which the models can be assumed to be Markovian stretched over an order of magnitude in the lag time. In summary, Fig. 6 shows that, using a hierarchical clustering, one can systematically vary the spatial resolution of the core-set discretization. By slightly relaxing the mapping criteria, one can ensure that the mass matrix is diagonally dominant and obtain well-converged core-set models even for very fine discretizations.

D. Dynamic model of the β-hairpin peptide

Figs. 7 and 8 show the structural interpretation of the core-set models with 41 and 63 cores. We applied PCCA+ analysis to group the cores into larger metastable sets (M1–M5). The identified metastable sets account for 60%–70% of all structures in the trajectory, depending on the model and the fuzziness-parameter in the PCCA+ analysis. The remaining structures were random coil structures with no stable hydrogen bond pattern. Remarkably, the structures in all metastable sets were stabilized by an ionic bond from the positively charged side chain of Arg$_1$ to either the carbonyl group at the C-terminus or to the negatively charged side chain of Glu$_{12}$ (Fig. 8). Thus, these two interactions seem to act as a brace which forces the peptide into a loop structure which then gives rise to various β-strand and β-bridge structures. Both models identified the metastable set M1, which is characterized by a very stable β-bridge between Thr$_5$ and Glu$_{12}$. The slowest process in the system is the exchange of this structure with the rest of the structural ensemble with an implied time scale of 790 ns in the 41-core-set model and of 900 ns in the 63 core-set model. The 41 core-set model next identified a relatively large metastable set M2, which consists of hairpin structures. However, the dssp-plot shows that the set covers at least two different types of hairpin structures, whose hydrogen bond patterns seem to be register shifted. For example, the carbonyl group of Lys$_9$ forms hydrogen bonds to the amino groups Ile$_4$ and Thr$_5$, which is not possible simultaneously. Likewise, the amino-group of Tyr$_{11}$ forms hydrogen bonds

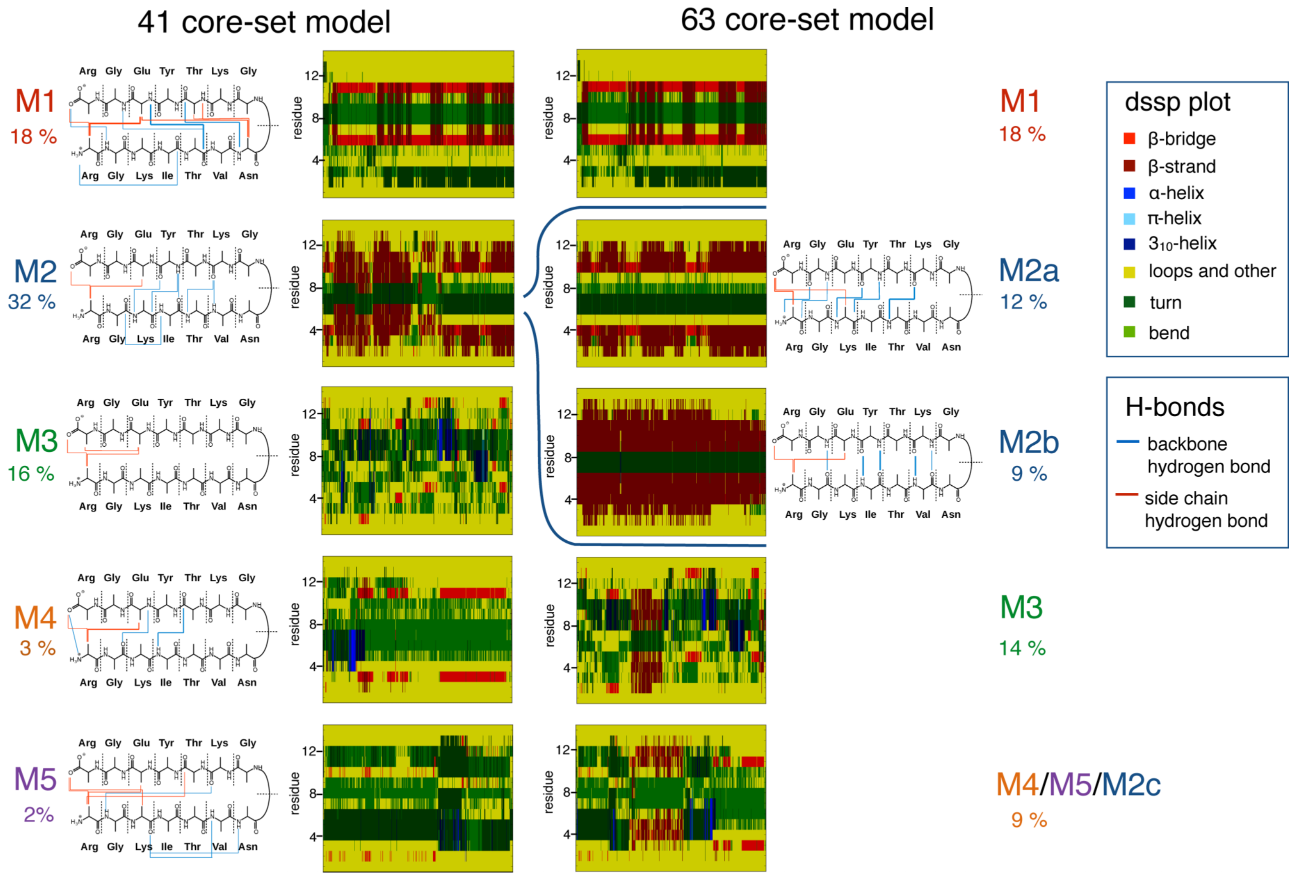


FIG. 8. Structural analysis of the metastable sets as identified by the 41-core-set model and the 63-core-set model, both at lag-time $\tau = 40$ ns. The dssp-plot and the most populated hydrogen bonds for each set of conformations are shown. M1 from the 63-core-set model has a large overlap with M1 from the 41-core-set model. The hydrogen bond pattern is only shown once. Likewise for M3. M4/M5/M2c from the 63-core-set model is a collection of multiple structures with no common hydrogen pattern. The percentages below the metastable-set index denote the relative population of the set. For the dssp-plots, 10.000 frames were extracted from each metastable set.

to the carbonyl groups of Lys_3 and Ile_4. Obviously, the 41 core-set discretization cannot resolve the difference between these two hairpin structures. The model further identified a relatively unstructured metastable set with M3 with various β-bridge contacts and about 16% relative population, a loop structure stabilized by a β-bridge between Lys_3 and Tyr_{11} (M4, 3% relative population), and a turn structure (M5) with about 2% relative population. The kinetic exchange between metastable sets M2, M3, M4, and M5 takes place at time scales of 150–250 ns.

The 63-core-set model split metastable set M2 into two subsets which interconvert at a time scale of about 640 ns. Each of the two sets now has a consistent dssp-plot (Fig. 8). Set M2a is stabilized by a hydrogen bond from the amino group of Thr_5 to the carbonyl group of Lys_9, i.e., a hydrogen bond of $i \rightarrow i + 4$ (α-turn). The other backbone hydrogen bonds in the M2 set are in register to this bond: $\text{Lys}_3 \rightarrow \text{Tyr}_{11}$, $\text{Lys}_3 \leftarrow \text{Tyr}_{11}$, $\text{Arg}_1 \rightarrow \text{Gly}_{13}$, and $\text{Arg}_1 \leftarrow \text{Gly}_{13}$. By contrast, set M2b is stabilized by two backbone hydrogen bonds between Val_6 and Lys_9, i.e., bonds of type $i \rightarrow i + 3$ (β-turn), with all the other backbone hydrogen bonds being in register with this bond: $\text{Ile}_4 \rightarrow \text{Tyr}_{11}$, $\text{Ile}_4 \leftarrow \text{Tyr}_{11}$, and $\text{Gly}_2 \leftarrow \text{Gly}_{13}$. Metastable set M3 in the 63-core-set model has a large overlap with M3 from the 41-core-set model. Performing the PCCA+ analysis on the dominant five eigenvectors on the core-set transition matrix splits the conformational space into five metastable sets. Consequently, the last metastable set of the 63-core-set model combines the quickly interconverting sets M4 and M5 (and some structures from M2) of the 41-core-set model into one metastable set.

The analysis of the core-set models for the β-hairpin shows that using a hierarchical clustering one can define sufficiently many cores to accurately describe the dynamics of even very flexible molecules with a large conformational space. Core-set models based on different levels of clustering are consistent among each other. Increasing the number of cores leads to a splitting of large metastable sets and hence an increase in spatial resolution. Moreover, the comparison between the 41-core-set model and the 63-core-set model shows that a high spatial resolution is not only necessary to decrease the approximation error but also to differentiate between similar structures which differ from a chemical point of view.

V. DISCUSSION AND CONCLUSIONS

We identified the highly populated areas in the conformational space using density-based cluster algorithms and used these clusters as cores in the core-set approach. We have shown that with this strategy one can obtain highly accurate models of the conformational dynamics. In particular, the number of cores needed to achieve a given approximation error is up to an order of magnitude smaller than the number of states in a conventional MSM with comparable approximation error.

We tested three different density-based cluster algorithms. The CNN[26] and the DBSCAN[27] algorithm consistently yielded very similar clusterings with tight clusters centered at the potential energy minima. By contrast, the JP algorithm[28] yielded larger numbers of cluster with more fuzzy boundaries. Core-set models of a terminally capped alanine (Ac-A-NHMe) based on the JP clustering were not in agreement with a 1296-state reference MSM, and no converged core-set model could be constructed from the JP-clustering of the 14-residue peptide (data not shown). These results indicate that the JP algorithm is not well-suited as a starting point for core-set models. In the JP algorithm the neighborhood of a data point is defined by the R nearest neighbors, rather than by a fixed distance R. The effective neighborhood radius grows with decreasing data point density which distorts the estimate of the data point density. As a consequence, the clusters are not defined by a drop in data point density and lead to ill-defined cores.

On the other hand, the CNN and the DBSCAN algorithm are equally suited for the identification of highly populated states in a conformational space and for the construction of core-set models. If the number of data points is relatively low compared to the dimensionality of the data set, as for example in the data set for 14-residue peptide with 19 950 frames, either a low value for the number of common neighbors N or a large value for the neighborhood radius R needs to be chosen for the CNN algorithm. In our experience, choosing a low number of N and adjusting R to the data-point density works better than fixing R to a high value and varying N, possibly because R can be varied continuously whereas N can only assume integer values.

Note that a recently introduced class of density-based cluster algorithms[53,54] is likely to also yield a suitable core-set discretization. In these algorithms, first the number of cluster centers are identified based on an estimate of the local data-point density at each data point[54] and the distance between high-density data points.[53] Then the remaining data points are assigned to the clusters based on the distance to the cluster members and local data point density of the unassigned point. The data-point density between two data points is not estimated. In Ref. 55, a different approach is used to identify core sets. First, on the order of 1000 trial milestone conformations are chosen based on their local probability density. From these trial milestones, a subset of core-set conformations are selected by maximizing a metastability index. This requires a search through the possible subsets of all trial conformations, for which the authors propose an elegant algorithm.

One of the main advantages of the identification of core-sets using density-based clustering is that one can extend the core-set approach to systems which are not strongly metastable. This is important for the practical application of the core-set method because many biologically interesting systems are only marginally metastable. We achieve this by applying the density-based cluster algorithm in a hierarchical manner while monitoring whether the matrix $\mathbf{M}$ remains diagonally dominant. The dominance of the diagonal elements in $\mathbf{M}$ is a measure for the metastability of the cores. This metastability can be further improved by relaxing the parameters which govern the mapping of individual trajectories onto the cores during the construction of the milestoning processes. This approach yields core-set models with a high spatial resolution. For example, we could distinguish between conformationally similar yet chemically different structures, such as register-shifted hairpin structures in a 14-residue peptide. Overall, combining density-based clustering with the core-set approach is an easy to use discretization method for Markov state models which in our test systems improved both the approximation error and the spatial resolution of the models.

SUPPLEMENTARY MATERIAL

See supplementary material for the used scripts for the density-based cluster algorithms as well as all further scripts for the creation of the core sets and the Markov State model. In addition to this, examples are included. Furthermore, the script for the Markov-Chain-Monte-Carlo-sampler, which was used for the creation of the 2D-dataset, is provided.

ACKNOWLEDGMENTS

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through Grant No. CRC 1114. O.L. acknowledges support by the DFG collaborative research center CRC 765. The computer facilities of the Freie Universität Berlin (ZEDAT) are acknowledged for computer time.

APPENDIX: DISCRETIZATION OF THE TRANSFER OPERATOR

The eigenfunctions are approximated by expanding them in a finite basis $\{\chi_i(\mathbf{x})\}_{i=1}^n$,

$$r(\mathbf{x}) \approx \sum_{i=1}^{n} c_i \chi_i(\mathbf{x}) \tag{A1}$$

with the basis functions spanning a subspace of the full molecular state space $D \subset \Omega$,

$$D := \text{span}\{\chi_1, \ldots, \chi_n\}. \tag{A2}$$

The expansion coefficients in Eq. (A1) can be obtained by a Galerkin discretization of eq.[31,33,37] This yields the following generalized eigenvalue problem:

$$\mathbf{C}(\tau)\mathbf{c} = \lambda \mathbf{S}\mathbf{c}, \tag{A3}$$

where the elements of the correlation matrix $\mathbf{C}(\tau)$ are given as

$$C_{ij}(\tau) = \langle \chi_i | \mathcal{T}(\tau)\chi_j \rangle_\pi = \int \chi_i(\mathbf{y})\pi(\mathbf{y})\left[\mathcal{T}(\tau)\chi_j(\mathbf{x})\right] d\mathbf{y} \tag{A4}$$

and the elements of the overlap matrix $\mathbf{S}$ as

$$S_{ij} = \langle \chi_i | \chi_j \rangle_\pi = \int \chi_i(\mathbf{x})\pi(\mathbf{x})\chi_j(\mathbf{x}) d\mathbf{x}. \tag{A5}$$

This discretization is used in the variational approach to molecular dynamics.[17–19] The overlap matrix is symmetric because the scalar product is symmetric, and it is invertible because the basis functions are linearly independent. Thus, to obtain the expansion coefficients, one can also solve the equivalent eigenvalue problem

$$\mathbf{S}^{-1}\mathbf{C}(\tau)\mathbf{c} = \mathbf{T}(\tau)\mathbf{c} = \lambda \mathbf{c}. \tag{A6}$$

The matrix $\mathbf{T}(\tau) = \mathbf{S}^{-1}\mathbf{C}(\tau)$ is the so-called projected transfer operator. The discretization in Eq. (A6) is used in conventional Markov state models.[13,32,34,38]

In the core-set approach[20,21] one uses Eq. (A6) as a starting point to derive a discretization with respect to an alternative basis of D

$$D := \text{span}\{\psi_1, \ldots, \psi_n\} \tag{A7}$$

with

$$\psi_i(\mathbf{x}) = \frac{\chi_i(\mathbf{x})}{\langle \chi_i | \mathbb{1} \rangle_\pi} \Leftrightarrow \chi_i(\mathbf{x}) = \langle \chi_i | \mathbb{1} \rangle_\pi \chi_i(\mathbf{x}). \tag{A8}$$

That is, one approximates the eigenfunctions by expanding them in $\{\psi_i(\mathbf{x})\}_{i=1}^n$

$$r(\mathbf{x}) \approx \sum_{i=1}^{n} \tilde{c}_i \psi_i(\mathbf{x}) = \sum_{i=1}^{n} \frac{\tilde{c}_i}{\langle \chi_i | \mathbb{1} \rangle_\pi} \chi_i(\mathbf{x}), \tag{A9}$$

and seeks the expansion coefficients $\tilde{\mathbf{c}}$. Note that in Eq. (A8) the basis functions are only scaled by a scalar $\langle \chi_i | \mathbb{1} \rangle_\pi$. Thus, both basis sets span the same subspace D and the corresponding expansion coefficients are related by

$$\mathbf{c} = \mathbf{\Pi}^{-1}\tilde{\mathbf{c}} \Leftrightarrow \tilde{\mathbf{c}} = \mathbf{\Pi}\mathbf{c}, \tag{A10}$$

where $\mathbf{\Pi}$ is diagonal matrix

$$\Pi_{ij} = \begin{cases} \langle \chi_i | \mathbb{1} \rangle_\pi & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}. \tag{A11}$$

Inserting into Eq. (A6) yields the following matrix equation for the expansion coefficients $\tilde{\mathbf{c}}$:

$$\mathbf{\Pi}\mathbf{S}^{-1}\mathbf{C}(\tau)\mathbf{\Pi}^{-1}\tilde{\mathbf{c}} = \lambda \tilde{\mathbf{c}}, \tag{A12}$$

where the matrices $\mathbf{C}(\tau)$, $\mathbf{S}$, and $\mathbf{\Pi}$ are defined with respect to the original basis $\{\chi_i(\mathbf{x})\}_{i=1}^n$ and are given by Eqs. (A4), (A5), and (A11). In the literature on the core-set method, the transpose of Eq. (A12)

$$\left[\mathbf{\Pi}\mathbf{S}^{-1}\mathbf{C}(\tau)\mathbf{\Pi}^{-1}\tilde{\mathbf{c}}\right]^\top = \tilde{\mathbf{c}}^\top\mathbf{\Pi}^{-1}\mathbf{C}(\tau)\mathbf{S}^{-1}\mathbf{\Pi} = \lambda\tilde{\mathbf{c}}^\top \tag{A13}$$

is typically used ($\mathbf{C}(\tau)$, $\mathbf{S}$, and $\mathbf{\Pi}$ are symmetric matrices), and we will adhere to this convention. Defining the matrix

$$\mathbf{P}(\tau) = \mathbf{\Pi}^{-1}\mathbf{C}(\tau) \qquad \text{with } P_{ij}(\tau) = \frac{\langle \chi_i | \mathcal{T}(\tau)\chi_j \rangle}{\langle \chi_i, \mathbb{1} \rangle_\pi} \tag{A14}$$

and the mass matrix

$$\mathbf{M} = \mathbf{\Pi}^{-1}\mathbf{S} \qquad \text{with } M_{ij} = \frac{\langle \chi_i | \chi_j \rangle}{\langle \chi_i | \mathbb{1} \rangle_\pi}, \tag{A15}$$

Eq. (A13) can be recast as

$$\tilde{\mathbf{c}}^\top\mathbf{P}(\tau)\mathbf{M}^{-1} = \lambda\tilde{\mathbf{c}}^\top. \tag{A16}$$

This discretization is used in the core-set approach.

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

[2] V. S. Pande, K. Beauchamp, and G. R. Bowman, "Everything you wanted to know about Markov state models but were afraid to ask," Methods **52**, 99 (2010).

[3] A. Sirur, D. De Sancho, and R. B. Best, "Markov state models of protein misfolding," J. Chem. Phys. **144**, 075101 (2016).

[4] S. Doerr, M. J. Harvey, F. Noé, and G. D. Fabritiis, "HTMD: High-throughput molecular dynamics for molecular discovery," J. Chem. Theory Comput. **12**, 1845 (2016).

[5] G. R. Bowman, D. L. Ensign, and V. S. Pande, "Enhanced modeling via network theory: Adaptive sampling of Markov state models," J. Chem. Theory Comput. **6**, 787 (2010).

[6] Q. Qiao, G. R. Bowman, and X. Huang, "Dynamics of an intrinsically disordered protein reveal metastable conformations that potentially seed aggregation," J. Am. Chem. Soc. **135**, 16092 (2013).

[7] C. T. Leahy, R. D. Murphy, G. Hummer, E. Rosta, and N.-V. Buchete, "Coarse master equations for binding kinetics of amyloid peptide dimers," J. Phys. Chem. Lett. **7**, 2676 (2016).

[8] M. Schor, A. S. J. S. Mey, F. Noé, and C. E. MacPhee, "Shedding light on the dock–lock mechanism in amyloid fibril growth using Markov state models," J. Phys. Chem. Lett. **6**, 1076 (2015).

[9] J. Witek, B. G. Keller, M. Blatter, A. Meissner, T. Wagner, and S. Riniker, "Kinetic models of cyclosporin A in polar and apolar environments reveal multiple congruent conformational states," J. Chem. Inf. Model. **56**, 1547 (2016).

[10] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande, "Progress and challenges in the automated construction of Markov state models for full protein systems," J. Chem. Phys. **131**, 124101 (2009).

[11] M. Sarich, F. Noé, and C. Schütte, "On the approximation quality of Markov state models," Multiscale Model. Simul. **8**, 1154 (2010).

[12] L. V. Nedialkova, M. A. Amat, I. G. Kevrekidis, and G. Hummer, "Diffusion maps, clustering and fuzzy Markov modeling in peptide folding transitions," J. Chem. Phys. **141**, 114102 (2014).

[13] J. D. Chodera, N. Singhal, V. S. Pande, K. A. Dill, and W. C. Swope, "Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics," J. Chem. Phys. **126**, 155101 (2007).

[14] G. R. Bowman and P. L. Geissler, "Equilibrium fluctuations of a single folded protein reveal a multitude of potential cryptic allosteric sites," Proc. Natl. Acad. Sci. U. S. A. **109**, 11681 (2012).

[15] R. T. McGibbon and V. S. Pande, "Learning kinetic distance metrics for markov state models of protein conformational dynamics," J. Chem. Theory Comput. **9**, 2900 (2013).

[16] G. Pérez-Hernández, F. Paul, T. Giorgino, G. D. Fabritiis, and F. Noé, "Identification of slow molecular order parameters for Markov model construction," J. Chem. Phys. **139**, 015102 (2013).

[17] F. Noé and F. Nüske, "A variational approach to modeling slow processes in stochastic dynamical systems," Multiscale Model. Simul. **11**, 635 (2013).

[18] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, "Variational approach to molecular kinetics," J. Chem. Theory Comput. **10**, 1739 (2014).

[19] F. Vitalini, F. Noé, and B. G. Keller, "A basis set for peptides for the variational approach to conformational kinetics," J. Chem. Theory Comput. **11**, 3992 (2015).

[20] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden, "Markov state models based on milestoning," J. Chem. Phys. **134**, 204105 (2011).

[21] M. Sarich, R. Banisch, C. Hartmann, and C. Schütte, "Markov state models for rare events in molecular dynamics," Entropy **16**, 258 (2013).

[22] W. E and E. Vanden-Eijnden, "Transition-path theory and path-finding algorithms for the study of rare events," Annu. Rev. Phys. Chem. **61**, 391 (2010), https://web.math.princeton.edu/~weinan/.

[23] A. K. Faradjian and R. Elber, "Computing time scales from reaction coordinates by milestoning," J. Chem. Phys. **120**, 10880 (2004).

[24] E. Vanden-Eijnden, M. Venturoli, G. Ciccotti, and R. Elber, "On the assumptions underlying milestoning," J. Chem. Phys. **129**, 174102 (2008).

[25] E. Vanden-Eijnden and M. Venturoli, "Markovian milestoning with Voronoi tessellations," J. Chem. Phys. **130**, 194101 (2009).

[26] B. Keller, X. Daura, and W. F. van Gunsteren, "Comparing geometric and kinetic cluster algorithms for molecular simulation data," J. Chem. Phys. **132**, 074110 (2010).

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD-96 Proceedings* (AAAI, 1996), p. 226.

[28] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," IEEE Trans. Comput. **C-22**, 1025 (1973).

[29] D. V. D. Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen, "GROMACS: Fast, flexible, and free," J. Comput. Chem. **26**, 1701 (2005).

[30] N.-V. Buchete and G. Hummer, "Coarse master equations for peptide folding dynamics," J. Phys. Chem. B **112**, 60576069 (2008).

[31] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard, "A direct approach to conformational dynamics based on hybrid Monte Carlo," J. Comput. Phys. **151**, 146 (1999).

[32] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, "Markov models of molecular kinetics: Generation and validation," J. Chem. Phys. **134**, 174105 (2011).

[33] C. Schütte, "Conformational dynamics: Modelling, theory, algorithm, and application to biomolecules," Habilitation thesis, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1999.

[34] B. Keller, P. Hünenberger, and W. F. van Gunsteren, "An analysis of the validity of Markov state models for emulating the dynamics of classical molecular systems and ensembles," J. Chem. Theory Comput. **7**, 1032 (2011).

[35] J.-H. Prinz, B. Keller, and F. Noé, "Probing molecular kinetics with Markov models: Metastable states, transition pathways and spectroscopic observables," Phys. Chem. Chem. Phys. **13**, 16912 (2011).

[36] B. G. Keller, J.-H. Prinz, and F. Noé, "Markov models and dynamical fingerprints: Unraveling the complexity of molecular kinetics," Chem. Phys. **396**, 92 (2012).

[37] C. Schütte and M. Sarich, "A critical appraisal of Markov state models," Eur. Phys. J.: Spec. Top. **224**, 2445 (2015).

[38] W. C. Swope, J. W. Pitera, and F. Suits, "Describing protein folding kinetics by molecular dynamics simulations. 1. Theory," J. Phys. Chem. B **108**, 6571 (2004).

[39] F. Noé, I. Horenko, C. Schütte, and J. C. Smith, "Hierarchical analysis of conformational dynamics in biomolecules: Transition networks of metastable states," J. Chem. Phys. **126**, 155102 (2007).

[40] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," J. Chem. Phys. **21**, 1087 (1953).

[41] F. Vitalini, A. S. J. S. Mey, F. Noé, and B. G. Keller, "Dynamic properties of force fields," J. Chem. Phys. **142**, 084101 (2015).

[42] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw, "Improved side-chain torsion potentials for the Amber ff99SB protein force field," Proteins **78**, 1950 (2010).

[43] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," J. Chem. Phys. **79**, 926 (1983).

[44] G. Bussi, D. Donadio, and M. Parrinello, "Canonical sampling through velocity rescaling," J. Chem. Phys. **126**, 014101 (2007).

[45] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije, "LINCS: A linear constraint solver for molecular simulations," J. Comput. Chem. **18**, 1463 (1997).

[46] T. Darden, D. York, and L. Pedersen, "Particle mesh Ewald: An N·log(N) method for Ewald sums in large systems," J. Chem. Phys. **98**, 10089 (1993).

[47] V. A. Gil and V. Guallar, "pyRMSD: A Python package for efficient pairwise RMSD matrix calculation and handling," Bioinformatics **29**, 2363 (2013).

[48] D. L. Theobald, "Rapid calculation of RMSDs using a quaternion-based characteristic polynomial," Acta Crystallogr., Sect. A **61**, 478 (2005).

[49] P. Deuflhard and M. Weber, "Robust Perron cluster analysis in conformation dynamics," Linear Algebra Appl. **398**, 161 (2005).

[50] M. Senne, B. Trendelkamp-Schroer, A. S. J. S. Mey, C. Schütte, and F. Noé, "EMMA: A software package for Markov model building and analysis," J. Chem. Theory Comput. **8**, 2223 (2012).

[51] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "PyEMMA 2: A software package for estimation, validation, and analysis of Markov models," J. Chem. Theory Comput. **11**, 5525 (2015).

[52] T. Kortemme, M. Ramírez-Alvarado, and L. Serrano, "Design of a 20 amino-acid, three-stranded β-sheet protein," Science **281**, 253 (1998).

[53] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," Science **344**, 1492 (2014).

[54] F. Sittel and G. Stock, "Robust density-based clustering to identify metastable conformational states of proteins," J. Chem. Theory Comput. **12**, 2426 (2016).

[55] E. Guarnera and E. Vanden-Eijnden, "Optimized Markov state models for metastable systems," J. Chem. Phys. **145**, 024102 (2016).