Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

MARCUS WEBER, SUSANNA KUBE, ALEXANDER
RIEMER, ALEXANDER BUJOTZEK

Efficient Sampling of the Stationary Distribution of Metastable Dynamical Systems

ZIB-Report 07-03 (February 2007)

Efficient Sampling of the Stationary Distribution of Metastable Dynamical Systems*

Marcus Weber, Susanna Kube, Alexander Riemer, Alexander Bujotzek

February 8, 2007

Abstract

This article deals with an efficient sampling of the stationary distribution of dynamical systems in the presence of metastabilities. For such systems, standard sampling schemes suffer from trapping problems and critical slowing down. Starting multiple trajectories in different regions of the sampling space is a promising way out. The different samplings represent the stationary distribution locally very well, but are still far away from ergodicity or from the global stationary distribution. We will show how these samplings can be joined together in order to get one global sampling of the stationary distribution.

AMS MSC 2000: 62H30, 82B80, 65C40,

Keywords: dynamical systems, stationary distribution, rare events, metastability, cluster analysis

*Supported by the DFG Research Center MATHEON "Mathematics for key technologies" in Berlin

Contents

Introduction

Consider general dynamical systems in equilibrium with a unique stationary distribution. Among them, metastable dynamical systems are characterized by the existence of almost invariant subsets in any of which the system remains for "a long time" before it switches to another subset. Such systems play an important role in molecular modeling, where the almost invariant subsets are known as metastable conformations [4]. In case of general dynamical systems, we also speak of metastable clusters.

Since dynamical systems are mostly high-dimensional, information about transition rates, average life times or the stationary distribution cannot be derived analytically but must be obtained from sampling data. Metastable dynamical systems inhibit different time scales, ranging from fast oscillations on the microscopic level to slow transitions between clusters. Consequently, sampling the stationary distribution with a single trajectory is impossible because it will get trapped within the clusters most of the time. Thus, it will take too much time to sample all relevant parts of the sampling space.

To solve this problem, one could start several independent trajectories in different regions of the sampling space such that these sub-samplings represent the global stationary distribution locally well. The goal is to assemble the sub-samplings into one global sampling of the stationary distribution by introducing point-wise weighting factors. This task is associated with the identification of metastable clusters and the calculation of thermodynamical weights. The resulting algorithm is named GLUE and consists of the following four steps:

1. **G**eneration of sampling data: Start different samplings of the dynamical system. Use a method which converges against the stationary distribution according to the law of large numbers. Due to early truncation, this limit is certainly not reached. The samplings are incomplete.

2. **L**ocalization of well-sampled parts: For each sampling, describe the parts of the state space which have a sufficiently high density of sampling points. The result of this step is a meshless discretization of the state space.

3. **U**ncoupling step: Compute a stochastic "mass" matrix on this discretization. The matrix is used to identify the metastable parts of the sampling via Robust Perron Cluster Analysis (PCCA+).

4. **E**xtraction of the global stationary distribution: Within each of the metastable regions, the samplings are rapidly mixing and represent the stationary distribution locally well. Weight the sub-samplings against each other in order to obtain the global stationary distribution.

In the following sections, the four steps of the GLUE algorithm are explained in detail. For each step, we describe the mathematical background and propose a possible algorithmic implementation w.r.t. statistical thermodynamics. Furthermore, an artificial example is presented for illustrative calculations. Each step of the method imposes some prerequisites on the dynamical system. These conditions are mentioned explicitly.

0.1 Notation

We briefly list the main notations used throuout the paper.

Ω	State space.
q	State in Ω.
$\pi(q)$	(Unnormalized) Boltzmann density in $q \in \Omega$.
N	Number of basis functions.
$\{\Phi_i(q)\}_{i=1}^N$	Set of basis functions, $\Phi_i : \Omega \to [0,1]$.
$\{w_i\}_{i=1}^N$	Statistical weights of the basis functions.
$\mathbf{w}$	Vector of the statistical weights of the basis functions.
$\{\overline{q}_i\}_{i=1}^N$	Defining nodes of the basis functions.
n_C	Number of clusters.
$\{c_i\}_{i=1}^{n_C}$	Statistical weights of the clusters.
$\mathbf{c}$	Vector of the statistical weights of the clusters.
$q_i^{(j)}$	i-th sampling point from trajectory j, $q_i^{(j)} \in \Omega$.
$\xi_i^{(j)}$	Weight of sampling point $q_i^{(j)}$, $\xi_i^{(j)} \subset \mathbb{R}_+$.
N_{sam}	Number of generated trajectories.
$BF(j)$	Indices of all basis functions generated from sampling j.

1 Step 1: Generation of sampling data

In this section we describe the conditions that must be satisfied by the sub-sampling trajectories so that the GLUE algorithm will work. The two main aspects are convergence towards the stationary distribution and exploration of all relevant parts of the state space. In many application, this first step of the algorithm can be skipped because the trajectories are already given.

1.1 Mathematical background

Sampling procedure. In general, the sampling procedure used for the first step of GLUE is an iterative method which generates a *chain* of points $q_i \in \Omega$, $i = 1, \ldots, N$,

$$q_1 \to q_2 \to q_3 \to \ldots \to q_N. \tag{1}$$

In the most general setting, the order of the sampling points in (1) is meaningless, only their distribution is required. However, the present algorithm is capable of incorporating dynamic information about the system if such is reflected in the chain's transitions. This case occurs if the chain (1) is a *reversible Markov chain* [2] with fixed time step τ. Then the GLUE algorithm has a slight modification in Step 3, see Section 3.1.

A prerequisite for the first step of the GLUE algorithm is the following.

Prerequisite 1.1 *The sampling procedure converges against a unique stationary distribution of sampling points in* Ω.

Due to early truncation, this limit will not be reached. However, we do not require the single sub-trajectories to cover the whole sampling space. Only the ensemble of all trajectories must represent the stationary distribution sufficiently. This requirement is described next.

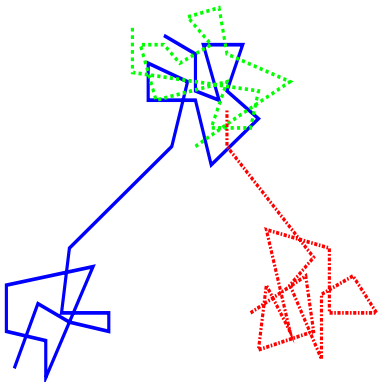


Figure 1: The union of three incomplete trajectories samples the complete state space sufficiently. *Solid line:* A trajectory which samples different conformations locally well. Due to rare transitions the relative weights of the conformations are sampled incorrectly. *Dashed line:* A trajectory which samples only one conformation correctly and does not reach the whole conformational space. *Dash-dotted line:* A trajectory which samples one conformation correctly and another conformation insufficiently.

Completeness. In the present context, $N_{sam} \in \mathbb{N}$ samplings have been generated via a procedure which meets Prerequisite 1.1. Thus, we have a set of sampling points $q_i^{(j)} \in \Omega$. The upper index (j) refers to the sampling $j = 1, \ldots, N_{sam}$, and the lower index i denotes the i-th step of the corresponding sampling, $i = 1, \ldots, N_j$. Clearly, in order to derive all relevant information from these N_{sam} samplings, they should cover the sampling space sufficiently.

Prerequisite 1.2 *For every relevant part of the state space Ω, there is at least one sampling $j \in \{1, \ldots, N_{sam}\}$ which covers this part of the space sufficiently.*

An example for this intuitively formulated Prerequisite 1.2 is given in Figure 1.

1.2 Algorithmic details

A valid example for a sampling procedure that meets Prerequisite 1.1 is the Hybrid Monte Carlo (HMC) method [6]. The trajectory generated by HMC is a reversible Markov chain which converges against the Boltzmann distribution (or the canonical ensemble) in position space Ω. Prerequisite 1.2 is not easy to ensure. A simple heuristic, which may be sufficient for many cases, is to run a high temperature HMC sampling first. From these sampling points, one can then choose a predefined number of points as possible starting points for the sub-samplings at a low temperature. For example, one can choose the centroids of a K-means clustering [11, 12, 7].

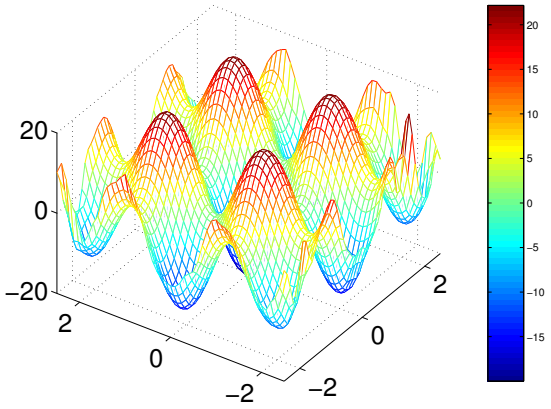


Figure 2: Artificial example: Potential energy surface.

1.3 Artificial example

Consider a two-dimensional dynamical system given by Hamiltonian differential equations w.r.t. positions q and momenta p,

$$\dot{q} = p, \qquad \dot{p} = -\nabla V(q).$$

Let be given a potential energy function $V : \mathbb{R}^2 \mapsto \mathbb{R}$ in the form

$$V(q) = \begin{cases} f(q_1) + f(q_2), & \text{if } |q_1| < 3 \wedge |q_2| < 3 \\ f(q_1) + g(q_2), & \text{if } |q_1| < 3 \wedge |q_2| \geq 3 \\ g(q_1) + f(q_2), & \text{if } |q_1| \geq 3 \wedge |q_2| < 3 \\ g(q_1) + g(q_2), & \text{if } |q_1| \geq 3 \wedge |q_2| \geq 3 \end{cases}$$

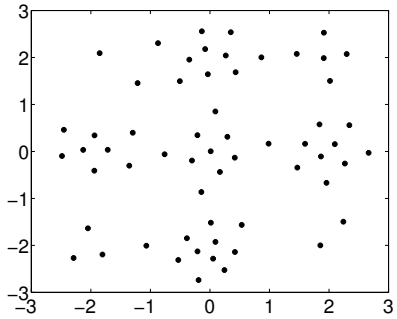
where

$$f(x) = -10\cos(3x) + x^2, \quad g(x) = -10\cos(3x^2) + x^2 + 1000(x-3)^2(x+3)^2.$$

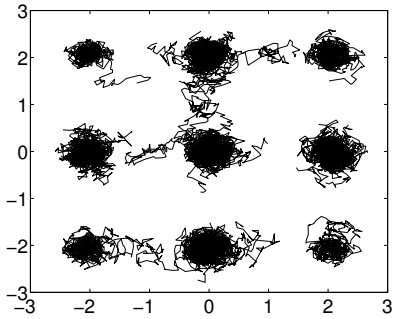
This potential has nine well-separated local minima, each of which induces a conformation, see Figure 2. By adding large penalty terms for $|q_{1,2}| \geq 3$, the domain of interest is restricted to $[-3, 3]^2$. Trajectories will almost never leave this domain, because the potential energy is too high outside.

For a first exploration of the sampling space, we start a high-temperature ($T = 1000K$) HMC trajectory in $q = (0, 0)$. For the proposal step, we apply 60 steps of the Verlet algorithm with time step $\tau = 0.001$[1]. 15000 sampling points are generated. Then we apply K-means with a predefined number of groups, $k = 60$, and choose the centroids as starting points for the sub-trajectories, see Figure 3(a). The sub-trajectories are generated by HMC sampling at $T = 300K$ and have length 500. Some of the trajectories cross a barrier between different basins of attraction, but most of them stay near the starting point, see Figure 3(b). Thus, the overlap is too small to allow the application of bridge sampling techniques.

[1] In our artificial example the variables of interest are dimensionless.



(a) Starting points from high temperature HMC sampling.



(b) Sub-trajectories started from the points in Figure 3(a).

Figure 3: Artificial example: Step 1 - Generation of sampling data. The starting points for 60 sub-trajectories are selected by K-means clustering of a trajectory from high-temperature HMC sampling.

2 Step 2: Localization of well-sampled parts

Given the sub-trajectories, the task is to identify the regions which have a sufficiently high density of sampling points. If the sub-samplings cover all relevant parts of the sampling space, they especially comprise the conformations. Therefore, we place a predefined number of nodes within the well-sampled regions, which will induce the meshless discretization in position space.

2.1 Mathematical background

Assume we have selected some nodes $\bar{q}_i$ from the sub-samplings. The selection rules will be described below. These nodes are the starting points for a discretization.

Meshless approach. Given the nodes $\bar{q}_i$, we define meshless basis functions $\Phi_i : \Omega \to [0,1]$, $i = 1, \ldots, N$, of the following form:

$$\Phi_i(q) = \frac{\exp(-\alpha\, d^2(q, \bar{q}_i))}{\sum_{j=1}^{N} \exp(-\alpha\, d^2(q, \bar{q}_j))}. \tag{2}$$

The parameter $\alpha > 0$ is a shape parameter and $d : \Omega \to \mathbb{R}$ denotes a distance function explained below. By definition, the basis functions form a partition of unity,

$$\sum_{i=1}^{N} \Phi_i(q) = 1, \quad \forall\, q \in \Omega.$$

This property makes them particularly accessible to reweighting techniques, such as those first introduced in [15], which became part of the software code ZIBgridfree [17].

The basis functions depend on a distance function $d : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$, which measures the distance between a sampling point q and a node $\bar{q}_i$. If the basis

functions are to be used for umbrella sampling [14, 15], the distance function must have some special properties. More precise, it must be a Euclidean distance in some lower-dimensional space. Here, we only assume the following:

Prerequisite 2.1 *For any two points $q_1, q_2 \in \Omega$ there is a non-negative distance-like function $d(q_1, q_2)$ (see 2.2).*

Node selection. As a result of the cluster algorithm, which will be described in Section 3.2, the basis functions are assigned to the conformations. The algorithm is based on the assumption that, for each conformation, there exists at least one basis function which completely belongs to this conformation. That assumption leads to the following condition for the node selection.

Prerequisite 2.2 *Every conformation must be resolved by at least one basis function, i.e. there must exist at least one node per conformation.*

For each sub-trajectory, there exists at least one region where the sampling point density is high. These high-density regions form a superset of the clusters we are looking for. Consequently, if we can ensure that the selected nodes cover all high-density regions of the sub-trajectories, they will certainly also cover the metastable conformations.

2.2 Algorithmic details

Nodes selection. Each sub-sampling should induce at least one basis function represented by a node $\bar{q}_i$. The number of basis functions should correspond to the number of high-density regions of the sub-sampling. Such regions can be identified by geometric cluster algorithms like K-means. Alternatively, one can perform local density estimation [13]. Since the points selected via local density estimation cannot be ensured to cover all high-density regions of the sub-samplings, geometric cluster analysis should be preferred.

Distance function. The term "distance-like" in Prerequisite 2.1 needs further explanation. In our applications, we describe the geometric configuration of molecules by the values of certain dihedral angles, the so-called "essential degrees of freedom" [1, 8]. Then the distance is the Euclidean metric in $\mathbb{R}^{n_d}$, where n_d denotes the number of dihedral angles. Other coordinates are possible as well, for example the $3s$-dimensional cartesian coordinates of a molecular system consisting of s atoms. Since rotation and translation of the molecule do not change its "state" from a chemical point of view, an alignment algorithm like Kabsch's algorithm [9, 10] must be applied before.

2.3 Artificial example

Look at the example from section 1.3 again. We want to define two basis functions per sub-trajectory, i.e. we have to select two nodes per trajectory. For each trajectory, we randomly pick 20 points in that trajectory. Then we compute the mean distance of each of the 20 points to the other 19 points. Afterwards, we select the two points with the lowest mean distances. The locations of these 120 points are shown in Figure 4. They directly define the basis functions $\{\Phi_s\}_{s=1}^{N}$ with $N = 120$. For the construction of the Φ_s we set $\alpha = 20$.

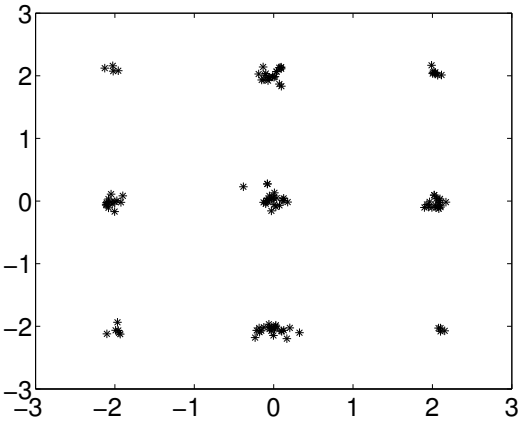


Figure 4: Artificial example: Step 2 - Node selection. The nodes define the set of basis functions. They were selected as the points with smallest mean distance among randomly chosen points from the sub-trajectories.

3 Step 3: Uncoupling step

The aim of step 3 is to determine the number of metastable clusters and to assign the basis functions to these clusters. This requires the construction of a mass matrix or a transition matrix, which is the starting point for cluster analysis via PCCA+.

3.1 Mathematical background

At this stage of the algorithm, we have selected N nodes $\overline{q}_1, \ldots, \overline{q}_N \in \Omega$. Each node is associated with a sampling. To make explanations clearer, let us introduce an upper index $j \in \{1, \ldots N_{sam}\}$ for the basis function $\Phi_i(q)$ as well as for the nodes $\{\overline{q}_i\}$. Then $\Phi_i^{(j)}(q)$ means that the node of basis function i, $\overline{q}_i^{(j)}$, stems from sub-trajectory j.

Since there is no one-to-one correspondence between sub-trajectories and metastable conformations, the clustering cannot be based on the sub-trajectories, but must be based on the basis functions. The input data for solving this cluster problem is a stochastic $n \times n$ matrix that describes the static or dynamic overlap between the basis functions. A stochastic matrix is a non-negative matrix with row sums 1. As mentioned in Section 1.1, the algorithm works slightly differently depending on the sampling procedure (1). The general case leads to a stochastic "mass" matrix S that characterizes the static overlap between the basis functions measured w.r.t. the stationary distribution. The entries of S are given by

$$S(i,j) = \frac{\int_\Omega \Phi_i(q)\, \Phi_j(q)\, \pi(q)\, dq}{\int_\Omega \Phi_i(q)\, \pi(q)\, dq}.$$

The integral is evaluated via Monte Carlo integration,

$$S(i,j) = \frac{\sum_{l=1}^{N_k} \Phi_i^{(k)}(q_l^{(k)})\, \Phi_j(q_l^{(k)})}{\sum_{l=1}^{N_k} \Phi_i^{(k)}(q_l^{(k)})}. \tag{3}$$

Note that we only take into account the contribution of sampling points from the trajectory k that gave rise to basis function i.

If the trajectories represent a reversible Markov chain, one can construct a stochastic "transition" matrix P containing the transition probabilities between the basis functions w.r.t. the Markov chain (1). The entry $P(i,j)(\tau)$ is the probability that a trajectory passes from the high-density region covered by basis function i to the high-density region covered by basis function j within some fixed time τ. The time interval τ is given by the time step between two consecutive sampling points. Analytically,

$$P(i,j)(\tau) = \frac{\int_\Omega \Phi_i(q)\, \Phi_j(\Psi^\tau q)\, \pi(q)\, dq}{\int_\Omega \Phi_i(q)\, \pi(q)\, dq},$$

where Ψ^τ denotes a dynamic propagator, for example an all-atom MD simulation method. Monte Carlo integration yields

$$P(i,j)(\tau) = \frac{\sum_{l=1}^{N_k-1} \Phi_i^{(k)}(q_l^{(k)}) \Phi_j(q_{l+1}^{(k)})}{\sum_{l=1}^{N_k-1} \Phi_i^{(k)}(q_l^{(k)})}. \tag{4}$$

The stochastic matrix can be used for the identification of metastabilities. In the following, the matrix will be denoted by P, although the theory is valid for the mass matrix S as well. In the presence of metastable conformations, one can find an appropriate permutation of indices such that the stochastic matrix P becomes nearly block-diagonal [3]. Each block of P corresponds to one metastable conformation. Any cluster algorithm can be used which assigns the basis functions, represented by the rows and columns of the transition matrix, to the blocks.

3.2 Algorithmic details

Robust Perron Cluster Analysis (PCCA+) can be used to identify those basis functions which belong to the same block in P and therefore to the same metastable conformation [5, 15]. The result of PCCA+ applied to a stochastic matrix $P \in \mathbb{R}^{n\times n}$ is a non-negative matrix $\chi \in \mathbb{R}^{n\times n_c}$, where n_c is the number of hidden blocks corresponding to the number of metastable conformations. The entry $\chi(i,j) \in [0,1]$ denotes the degree of membership of basis function Φ_i w.r.t. the j-th metastable conformation. This can be seen as a *fuzzy clustering* result, as opposed to "crisp" clustering methods, which find a permutation of indices according to the blocks.

The left eigenvector $\mathbf{w} = (w_1, \ldots, w_N)^\top$ of the transition matrix P corresponding to the Perron root $\lambda = 1$ contains the statistical weights of the basis functions Φ_i,

$$\mathbf{w}^\top P = \mathbf{w}^\top, \tag{5}$$

where $w_i = \int_\Omega \Phi_i(q)\, \pi(q)\, dq$. The statistical weights $\{c_i\}_{i=1}^{n_C}$ of the clusters are then computed by

$$\mathbf{c} = \chi^\top \mathbf{w}. \tag{6}$$

3.3 Artificial example

Given the basis functions and the sub-trajectories, we compute the mass matrix S according to (3). An image of the matrix is shown in Figure 5. We apply

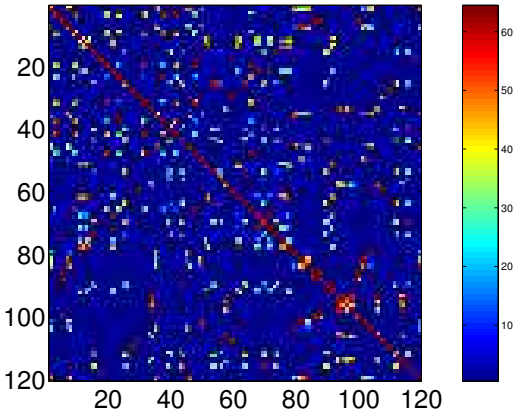


Figure 5: Artificial example: Step 3 - Stochastic mass matrix S.

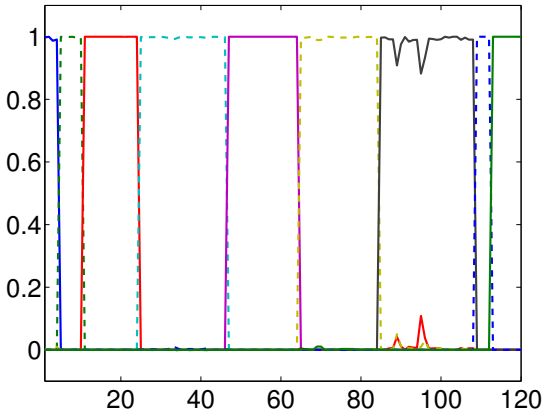


Figure 6: Artificial example: Step 3 - Membership vectors $\{\chi_i\}_{i=1}^9$.

PCCA+ and obtain the following spectrum of S,

$$\{\lambda_i\}_{i=1}^9 = \{1.000,\ 1.000,\ 1.000,\ 1.000,\ 0.9998,\ 0.9992,\ 0.9992,\ 0.9969,\ 0.9967\},$$

followed by a gap to the 10th eigenvalue $\lambda_{10} = 0.9001$. Thus we set $n_C = 9$. The corresponding sorted membership vectors χ are plotted in Figure 6. If the matrix S is reordered according to the membership vectors, the block diagonal structure becomes visible, see Figure 7.

4 Step 4: Extraction of the global stationary distribution

The goal of step 4 is to assign weights $\{\xi_i^{(j)}\}$ to all sampling points $\{q_i^{(j)}\}$ such that they represent the global stationary distribution.

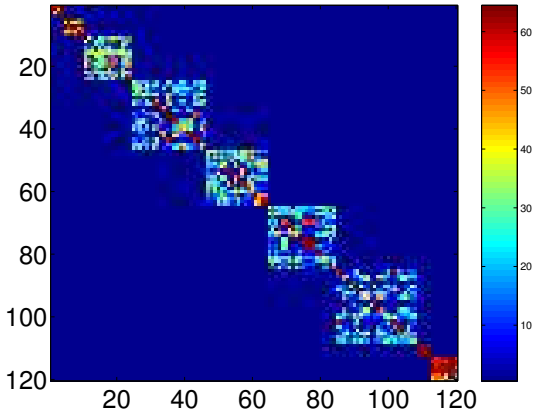


Figure 7: Artificial example: Step 3 - Reordered mass matrix.

4.1 Mathematical background

Equipped with the correct weights $\{\xi_i^{(j)}\}$, the sampling points can be used to compute observables,

$$\langle A \rangle = \int_\Omega A(q)\,\pi(q)\,dq \approx \sum_{j=1}^{N_{sam}} \sum_{i=1}^{N_j} A(q_i^{(j)})\,\xi_i^{(j)}.$$

The statistical weights of the points are given by

$$\xi_i^{(j)} = \sum_{s \in BF(j)} w_s \frac{\Phi_s(q_i^{(j)})}{\sum_{k=1}^{N_j} \Phi_s(q_k^{(j)})}. \tag{7}$$

Thus, the sum of the weights of all points from a trajectory j equals the sum of the weights of the basis functions which are induced by this trajectory. For reasons of computational efficiency, we do not consider the sum over all possible trajectories. The amount of statistical information we loose is small because the term $\Phi_s(q_i^{(j)})$ is expected to be small for $s \notin BF(j)$.

Equation (7) strongly depends on the statistical weights $\{w_s\}_{s=1}^N$ of the basis functions. Since the computation of $\mathbf{w}$ as eigenvector of the transition matrix is ill-conditioned [16], some of the weights w_s are incorrect. Therefore, we propose a reweighting strategy below, for which we need the following requirement.

Prerequisite 4.1 *The probability density function $\pi : \Omega \to \mathbb{R}$ is known pointwise except for a normalization constant.*

Prerequisite 4.1 is valid for the HMC method [6], which converges against the Boltzmann density. This density is known pointwise except for a normalization constant. For $q \in \Omega$, the Boltzmann density function is proportional to $\exp(-\beta V(q))$, where $\beta = (k_B T)^{-1}$ with Boltzmann constant k_B and fixed temperature T, and $V : \Omega \to \mathbb{R}$ is a known potential energy function.

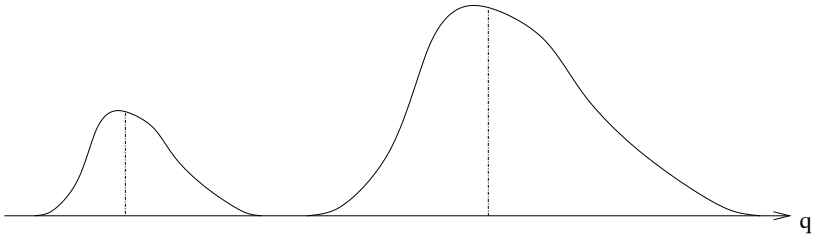


Figure 8: Artificial example: Step 4 - Rescaling of partial densities via pointwise density estimation. The value of the partial density in a single point determines the scaling factor as long as the density has locally been sampled correctly.

4.2 Algorithmic details

The relative statistical weight of an arbitrary point $q \in \Omega$ is given by the potential energy function, $\hat{\pi}(q) = \exp(-\beta V(q))$. It can also be approximated via local density estimation based on the sampling data. The value $\widetilde{\pi}(q)$ obtained from local density estimation will be wrong as long as the weights $\mathbf{w}$ of the basis functions and thus the weights $\xi_i^{(j)}$ of the sampling points are incorrect. However, the difference between $\hat{\pi}(q)$ and $\widetilde{\pi}(q)$ can be used to correct the weights $\mathbf{w}$ of the basis functions. The idea is the following. We assume that the partial densities were computed correctly from the eigenvalue problem [16]. By partial densities, we refer to the vector $\mathbf{w}$ restricted to basis functions belonging to the same cluster. Only the the scaling factors are wrong, which is equivalent to the fact that the cluster weights are incorrect. However, if we know the ratio $\hat{\pi}(q)/\widetilde{\pi}(q)$ in a single point q, we can use this ratio for reweighting, see Figure 8. The actual value of the density, $\widetilde{\pi}(q)$, can be estimated especially well in a point near a minimum of the potential energy surface because these regions are likely to be sampled sufficiently. The algorithmic realization is the following.

1. The basis functions must be assigned to the conformations. Therefore, a so-called *crisp clustering* is needed, i.e. a matrix $\overline{\chi} \in \mathbb{R}^{n \times n_C}$ with $\overline{\chi}(i,j) = 1$, if Φ_i belongs to conformation j, and $\overline{\chi}(i,j) = 0$ otherwise. If one applies PCCA+ to solve the clustering problem, one can determine $\overline{\chi}$ from χ in the following way. $\overline{\chi}(i,j) := 1$ if and only if $\chi(i,j)$ is the largest entry among the elements of the i-th row $\chi(i,:)$ of χ. In all other cases $\overline{\chi}(i,j) := 0$. Identical entries inside the rows of χ are unlikely to occur due to rounding errors in numerical computations.

2. For each conformation j, we select one representative basis function $\Phi_{j^\star}$, namely the one that contains the most information,

$$j^\star = \mathrm{argmax}_{i,\overline{\chi}(i,j)=1} \left(w_i \sum_{l=1}^{N_k} \Phi_i^{(k)}(q_l^{(k)}) \right). \tag{8}$$

   In other words, we select the basis function with large statistical weight and high density of sampling points, i.e. $\Phi_i^{(k)}(q_l^{(k)})$ is close to 1 for many points.

3. For the representative basis functions $\Phi_{j^\star}^{(k)}$, we search for the sampling point $q_{i^\star}^{(k)}$ from trajectory k with minimum potential energy. This point is briefly denoted by $q_{j^\star}$. The target histogram height is computed as

$$\hat{\pi}_j := \hat{\pi}(q_{j^\star}) = \exp(-\beta V(q_{j^\star})). \tag{9}$$

4. The histogram height from (9) can be rewritten as an integral containing the δ-distribution. That intergral can be approximated replacing the δ-distribution by an exponential kernel with shape parameter α_2 and using Monte Carlo quadrature:

$$\begin{aligned} \tilde{\pi}_j := \tilde{\pi}(q_{j^\star}) &= \int_\Omega \delta(q - q_{j^\star})\, \exp(-\beta V(q))\, dq \\ &\approx \int_\Omega \exp(-\alpha_2\, d_q^2(q, q_{j^\star}))\, \exp(-\beta V(q))\, dq \\ &= \sum_{k=1}^{N_{sam}} \sum_{i=1}^{N_k} \exp(-\alpha_2\, d_q^2(q_i^{(k)}, q_{j^\star}))\, \xi_i^{(k)} \end{aligned} \tag{10}$$

   This is an expression for the actual histogram hight depending on the sampling data. The distance function d_q will in general be different to the distance function d from Section 2.2. More precise, d_q must satisfy $d(q_1, q_2) = 0 \leftrightarrow q_1 = q_2$. A possible distance measure can be the root mean square distance, i.e. the Euclidean distance between aligned molecules.

5. Given the actual and the target histogram heights, we correct the statistical weights $\{w_s\}_{s=1}^N$ of the basis functions $\{\Phi_s\}$. For this purpose, we first identify the cluster index j for which $\overline{\chi}(s, j) = 1$. Then we set

$$w_s^{new} = \frac{w_s \hat{\pi}_j / \tilde{\pi}_j}{Z}, \tag{11}$$

   where Z is a normalization constant such that $\sum_s w_s = 1$.

6. Equivalently, we compute the cluster weights $\{c_j\}_{j=1}^{n_C}$,

$$c_j^{new} = \frac{c_j \hat{\pi}_j / \tilde{\pi}_j}{\sum_{i=1}^{n_C} c_i \hat{\pi}_i / \tilde{\pi}_i} \tag{12}$$

Given the new weights from (11), one can calculate the correct point weights in (7).

Remark 4.1 *Many sampling routines use* umbrella strategies [14]. *In this case the sampling points $q_i^{(j)}$ are not equally weighted. In the more general case, there are weights $\eta_i^{(j)} \geq 0$ for every sampling point $q_i^{(j)}$, which sum up to 1 for each sampling,*

$$\sum_i \eta_i^{(j)} = 1, \quad \forall\, j = 1, \ldots, N_{sam}.$$

The algorithm presented here needs only slight modifications to cover this situation. The additional factor $\eta_i^{(j)}$ must be introduced in every formula which is derived from an integral w.r.t. q.

4.3 Artificial example

First, the statistical weights of the clusters are computed from the matrix S by $\mathbf{w}^\top S = \mathbf{w}^\top$ and (6),

$$\begin{aligned}\{c_i\}_{i=1}^9 &= \{0.4564e\text{-}9,\ 0.8379e\text{-}7, 0.4462e\text{-}5,\ 0.3365e\text{-}6,\ 0.4007,\\ &\quad 0.6668e\text{-}7,\ 0.4703e\text{-}7,\ 0.1956,\ 0.4037\}.\end{aligned}$$

These weights are incorrect because $\mathbf{w}$ is wrong. This is supported by the fact that the weights do not reflect the symmetry of the potential energy surface. After having selected the point with the lowest potential energy within each cluster, we calculate the Boltzmann density in these points according to (9),

$$\begin{aligned}\{\hat{\pi}_i\}_{i=1}^9 &= \{0.0963,\ 0.0970,\ 0.5429,\ 0.5416,\ 0.5425,\ 0.5430,\ 3.0333,\\ &\quad 0.0965,\ 0.0972\} \cdot 10^3.\end{aligned}$$

The actual histogram height, approximated by local density estimation according to (10) with $\alpha_2 = 20$, amounts to

$$\begin{aligned}\{\tilde{\pi}_i\}_{i=1}^9 &= \{0.1958e\text{-}9,\ 0.3458e\text{-}7,\ 0.1874e\text{-}5,\ 0.1274e\text{-}6,\ 0.1613,\\ &\quad 0.2746e\text{-}7,\ 0.1683e\text{-}7,\ 0.8280e\text{-}1,\ 0.1980\}.\end{aligned}$$

Now we can compute the corrected cluster weights by (12),

$$\begin{aligned}\{c_i^{new}\}_{i=1}^9 &= \{0.0152,\ 0.0159,\ 0.0877,\ 0.0970,\ 0.0914,\ 0.0894,\ 0.5745,\\ &\quad 0.0155,\ 0.0134\}.\end{aligned}$$

Indeed, these weights reflect the symmetry of the potential energy surface correctly.

Conclusion

Outlook. The reweighting strategy for sampling points proposed in this paper is very similar to the method introduced in [16]. There, a stable algorithm has been presented for the computation of thermodynamical weights $\mathbf{w} = (w_1, \ldots, w_N)^\top$, which avoids solving the ill-conditioned eigenvector problem $\mathbf{w}^\top S = \mathbf{w}^\top$ numerically. Instead, a resampling technique is used. Since resampling is often impractical or even impossible, a data-based reweighting strategy is useful. In contrast to [16], we did not only describe this reweighting method, but also showed how a meshless discretization can be derived directly from the sampling data. We believe that our method is general and applicable to arbitrary metastable dynamical systems in equilibrium. Concerning the application to biomolecules, the algorithmic implementation is still ongoing work, but results will be presented soon.

Acknowledgement. We want to thank Peter Deuflhard for his contributions to the stability analysis of thermodynamical weight computation.

References

[1] A. Amadei, A. B. M. Linssen, and H. J. C. Berendsen. Essential dynamics of proteins. *Proteins*, 17:412–425, 1993.

[2] P. Brémaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Number 31 in Texts in Applied Mathematics. Springer-Verlag New York, 1999.

[3] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Lin. Alg. Appl.*, 315:39–59, 2000.

[4] P. Deuflhard and C. Schütte. Molecular conformation dynamics and computational drug design. In J. M. Hill and R. Moore, editors, *Applied Mathemetics Entering the 21st Century. Proceedings ICIAM 2003*, pages 91–119, 2004. Invited paper.

[5] P. Deuflhard and M. Weber. Robust Perron Cluster Analysis in Conformation Dynamics. In M. Dellnitz, S. Kirkland, M. Neumann, and C. Schütte, editors, *Lin. Alg. App. – Special Issue on Matrices and Mathematical Biology*, volume 398C, pages 161–184. Elsevier, 2005.

[6] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195(2):216–222, 1987.

[7] J. Hartingan and M. Wang. A K-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[8] S. Hayward and H. J. C. Berendsen. Systematic analysis of domain movements in proteins from conformational change: New results on citrate sythase and t4 lysozyme. *Proteins*, 30:144–154, 1998.

[9] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32:922–923, 1976.

[10] W. Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A34:827–828, 1978.

[11] S. Lloyd. Least squares quantization in pcm. *Bell Telefone Laboratories Paper, Marray Hill*, 1957.

[12] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symposium*, pages 281–297, 1967.

[13] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076, 1962.

[14] G. M. Torrie and J. P. Valleau. Monte Carlo study of a phase-separating liquid mixture by umbrella sampling. *J.Chem.Phys.*, 66(4):1402–1408, February 1977.

[15] M. Weber. *Meshless Methods in Conformation Dynamics*. Doctoral thesis, Department of Mathematics and Computer Science, Freie Universität Berlin, 2006. Published by Verlag Dr. Hut, München.

[16] M. Weber, S. Kube, L. Walter, and P. Deuflhard. Stable computation of propability densities for metastable dynamical systems. ZIB-Report 6-39, Zuse Institute Berlin, 2006.

[17] M. Weber and H. Meyer. ZIBgridfree - adaptive conformation analysis with qualified support of transition states and thermodynamic weights. ZIB-Report 05-17, Zuse Institute Berlin, 2005.