Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

T. Galliat P. Deuflhard R. Roitzsch F. Cordes

Automatic Identification of Metastable Conformations via Self-Organized Neural Networks

00-51 (December 2000)

Automatic Identification of Metastable Conformations via Self-Organized Neural Networks

T. Galliat, P. Deuflhard, R. Roitzsch, and F. Cordes

Abstract. As has been shown recently, the identification of *metastable* chemical conformations leads to a Perron cluster eigenvalue problem for a reversible Markov operator. Naive discretization of this operator would suffer from combinatorial explosion. As a first remedy, a pre-identification of essential degrees of freedom out of the set of torsion angles had been applied up to now. The present paper suggests a different approach based on neural networks: its idea is to discretize the Markov operator via self-organizing box maps. The thus obtained box decomposition then serves as a prerequisite for the subsequent Perron cluster analysis. Moreover, this approach also permits exploitation of additional structure within embedded simulations. As it turns out, the new method is fully automatic and efficient also in the treatment of biomolecules. This is exemplified by numerical results.

Keywords. Biochemical conformations, cluster analysis, molecular dynamics, Hybrid Monte-Carlo methods, Markov operator, Perron cluster analysis, Self-Organizing Maps

Introduction

The analysis of biomolecular structure and function is one of the real challenges of scientific computing nowadays. The key concept to characterize *structure* has become the characterization in terms of *geometrical conformations,* often just called conformations in literature. The present paper advocates that *function,* the most interesting aspect of biomolecules, should be rather characterized by what has been called *metastable conformations*. Any type of conformations consists of sets of possible molecular states. In geometrical conformations such sets are defined via the geometric similarity of different states. In metastable conformations such sets are defined via the high probability of the molecule to stay in such a set, once it is in such a set. From the point of view of dynamical systems the totality of all possible states is called *invariant set*. As an extension of that term, *almost invariant sets* turn out to be equivalent to metastable conformations in molecular systems.

In [3] a first attempt had been made to identify metastable conformations on the basis of the so-called Perron-Frobenius operator. That approach, though principally opening the door to the new concept of conformation dynamics, had been more or less restricted to toy molecules. Since then the Perron-Frobenius operator in phase space has been replaced by a different Markov operator T in position space [21,22]. This new operator has much nicer theoretical properties. Its numerical evaluation is done via Hybrid Monte-Carlo (HMC) methods; like classical Monte-Carlo, HMC also suffers from possible *trapping* in local potential wells. In order to overcome this unwanted effect, an adaptive temperature version has been worked out [9]. More recently an intelligent adaptation of temperature within the Boltzmann distribution has led to a hierarchical uncoupling/coupling method to be described in this same volume (see [10]). Once a moderate number k of spatial boxes has been determined, the Markov operator can be dicretized in the form of a stochastic (k, k)-matrix. The identification of metastable conformations then boils down to the numerical solution of an eigenvalue cluster problem, the so called Perron cluster eigenvalue problem. An efficient algorithm for *Perron cluster analysis* has been published in [5].

The actual determination of a moderate size of spatial boxes, however, turns out to be a hard problem in itself. In fact, naive decomposition of position space to discretize the Markov operator T would lead to a combinatorial explosion of the number of boxes, the unwanted curse of dimension. As a first remedy, *essential degrees of freedom (DOF)* in the spirit of [2] - on the basis of torsion angles only - led to a treatment of small molecules [22]. Unfortunately, for larger molecules, this technique appeared to be not robust enough. The topic of the present paper is to suggest an alternative spatial decomposition technique based on self-organized neural networks. First attempts have been published in [13]. Here we want to report about progress beyond that paper.

The present paper is organized as follows. In Section 1 we recall the main computational issues for the discretization of the above mentioned Markov operator to obtain a *stochastic matrix*. These are: (a) a spatial deomposition to obtain a moderate number k of boxes, (b) Hybrid Monte Carlo (HMC) methods for the approximation of the entries of the (k, k)-matrix, and (c) temperature or, more general, parameter embedding to avoid trapping in the HMC process. In Section 2 we present our new extension from the point concept of Kohonen's self-organizing maps (SOM) to the set concept of *self-organizing box maps* (SOBM). In order to be able to interpret the results of the neural cluster algorithms, we work out the concept of *discriminating variables* in Section 3. Finally, in Section 4, the algorithm is exemplified as part of the whole *Perron cluster* algorithm at several molecular systems: First, we give results for the simple n-pentane molecule, where everything is known and can therefore be compared in detail with other sources – see [10] also in this volume. Second, we present results for a potential anti-AIDS drug, an HIV protease inhibitor. Third, we report about the occurrence of metastable conformations within virtual screening of a molecular database.

1 Discretization of Markov operator

The present section discusses several computational issues in connection with the already mentioned Markov operator T as suggested by [21]. That operator is obtained from the Perron-Frobenius operator by some momenta averaging based on the Boltzmann distribution f_0 for given heat bath temperature. It may be interpreted as the transfer operator of an underlying Markov chain as shown schematically in Figure 1.

Fig. 1. Markov chain associated with Markov operator T.

Hybrid Monte-Carlo (HMC) methods. This Markov chain can be realized in the following 3-step process [9]:

- random choice of momenta from a Gaussian distribution,
- deterministic propagation of the molecular system by the flow Φ_V^τ with potential V and over *short* time τ,
- acceptance or rejection of new configurations by an appropriate transition kernel K of the underlying Markov process [8] e.g. Metropolis-Hastings.

Spatial decomposition. The evaluation of the operator T requires a spatial decomposition as a prerequisite. Let Θ be a covering of the position space Ω with k pairwise disjoint partitions Θ_s. Let $\Gamma(\Theta_1), ..., \Gamma(\Theta_k)$ denote associated fibers in phase space $\Gamma = \Omega \times \mathbf{R}^{3N}$, where N is the number of atoms in the molecule. For the corresponding characteristic functions we write $\chi_{\Gamma(\Theta_s)}$. On this basis the transfer operator can be discretized to yield the *transition matrix* S with entries

$$S_{sl} = \frac{\int_{\Gamma(\Theta_s)} \chi_{\Gamma(\Theta_l)}(\Phi_V^\tau x) f_0(x) dx}{\int_{\Gamma(\Theta_s)} f_0(x)}. \tag{1}$$

Each entry S_{sl} is the probability of a transition from subspace Θ_s to Θ_l during time τ.

Let us now restrict Ω to be the space spanned by q *torsion angles*. Without loss of generality we assume that each angle ranges within $[-\pi, \pi]$. For an introduction to the analysis of cyclic data see [11]. By P_ρ we denote the probability distribution on Ω which is uniquely determined by the probability density function $\rho := f_0$ associated with the Markov Operator T.

Definition 1. We call $\Theta := \{\Theta_1, \ldots, \Theta_k\}$ a *Voronoi tessellation* of Ω with partitions Θ_s, if

$$\bigcup_{s=1}^{k} \Theta_s = \Omega \quad \text{and} \quad \Theta_p \cap \Theta_s = \emptyset \quad \text{for all } p, s \in \{1, \ldots, k\}.$$

The quality of the discretization of the operator T depends on how well the corresponding Voronoi tessellation approximates the topology of Ω with respect to P_ρ. One possibility to measure the approximation quality is given by

Definition 2. Let Θ be a Voronoi tessellation of Ω with k partitions. Then we call

$$\vartheta[\Theta] := \sum_{s=1}^{k} \int_{x,y \in \Theta_s} \operatorname{dist}(x,y) \rho(x) \rho(y) dx dy \tag{2}$$

the *decomposition error* of Θ with respect to ρ, where $\operatorname{dist} : \Omega \times \Omega \to \mathbf{R}_0$ is a suitable distance measure on Ω, e.g., the distance on the q-dimensional unit circle:

$$\operatorname{dist}(x,y) = \sqrt{\sum_{i=1}^{q} (\sin(x_i) - \sin(y_i))^2 + (\cos(x_i) - \cos(y_i))^2} \tag{3}$$

for $x, y \in \Omega$, where x_i and y_i denote the values of the ith torsion angle.

It is obvious that from a theoretical point of view one is interested in the computation of a Voronoi tessellation so that the corresponding decomposition error is minimized with respect to some preassigned number k of partitions. In passing we note that this is a well-known problem also in fields like signal processing [15] and general cluster analysis [6,7]. In most practical applications such an optimal tessellation is usually not sufficient, because the corresponding rules how to decide whether a configuration $x \in \Omega$ is inside a partition Θ_s or not, are computationally too complex. Obviously such rules are much simpler, if we require that the Θ_s are merely *boxes* in Ω.

Definition 3. We call a subset $B \subset \Omega$ a *box* in Ω, if there exist intervals $I(l_i, r_i) \subset \mathbf{R}$ for $i = 1, \ldots, q$, with $B = \bigotimes_{i=1}^{q} I(l_i, r_i) := (I(l_1, r_1), \ldots, I(l_q, r_q))$. For $l_i \leq r_i$ we allow $I(l_i, r_i) = [l_i, r_i]$, $I(l_i, r_i) =]l_i, r_i]$, $I(l_i, r_i) = [l_i, r_i[$ and $I(l_i, r_i) =]l_i, r_i[$. If $l_i > r_i$ we define that $I(l_i, r_i)$ represents the complementary interval $[-\pi, \pi] \setminus I(r_i, l_i)$. We set $\mathrm{BOX}(\Omega) := \{B \mid B \,\mathrm{box\,in}\, \Omega\}$.

Uniform box-decomposition. For $i = 1, \ldots, q$ we choose $\mathrm{m}_i \in \mathbf{N}^+$ and real values $b_{i,1}, \ldots,$ b_{i,m_i} such that $-\pi \leq b_{1,m_i} \leq \ldots \leq b_{i,m_i} \leq \pi$. Then the intervals $I_{i,j} := [b_{i,j}, b_{i,j+1}]$ for $j = 1, \ldots, m_i - 1$ and $I_{i,m_i} := [-\pi, \pi] \backslash]b_{i,1}, b_{i,m_i}]$ are pairwise disjoint and build a covering of $[-\pi, \pi]$. We can easily compute $k = \prod_{i=1}^{q} m_i$ boxes $\varTheta_s$ by simple combination of one interval after the other per each torsion angle. It is then easy to check that $\varTheta_{\mathrm{uniform}} := \{\varTheta_1, \ldots, \varTheta_k\}$ builds a Voronoi tessellation of $\varOmega$. Therefore we call $\varTheta_{\mathrm{uniform}}$ a uniform k-box-decomposition of $\varOmega$. Obviously there are two main problems involved:

First, to optimize $\varTheta_{\mathrm{uniform}}$ with respect to P_ρ, i.e., to minimize the decomposition error according to Eq. (2), one has to determine optimal boundaries $b_{i,1}, \ldots, b_{i,m_i}$ for each variable. This would lead to an NP-hard combinatorial problem.

Second, remember that the transition matrix corresponding to operator T is of dimension $k \times k$; then, with increasing q, the number k grows exponentially for a uniform box-decomposition, even if we only allow two intervals per variable, i.e., if we set $m_i := 2$ for $i = 1, \ldots, q$. For example, a relative small molecule with only 20 torsion angles would thus lead to $k > 10^6$.

Approximate box-decomposition. In order to avoid the curse of dimension, it seems to be a reasonable idea to fix the parameter k at a suitable level and to compute a Voronoi tessellation of $\varOmega$ with k partitions $\varTheta_s \in \mathrm{BOX}(\varOmega)$ and minimal decomposition error. However, the actual realization of such an "optimal" box decomposition is very expensive. Moreover, practical experience shows that the decomposition error usually is very bad compared with the error for arbitrary partitions. We therefore suggest to fix the parameter k, but to relax the above definition of box-decomposition:

Definition 4. We call ($\varTheta$,$\varDelta$) an *approximate k-box-decomposition* of $\varOmega$ with respect to ρ, whenever $\varTheta := \{\varTheta_1, \ldots, \varTheta_k\}$ is a Voronoi tessellation of $\varOmega$ with a nearly optimal decomposition error according to Eq. (2) and $\varDelta$ is a set of k boxes $\varDelta_1, \ldots, \varDelta_k \in \mathrm{BOX}(\varOmega)$, such that $\mathrm{overlap}_{P_\rho}(\varDelta) \approx 0$ and $0 < \mathrm{overlay}_{P_\rho}(\varTheta, \varDelta) \leq 1$.

Herein we use the terms *overlap* and *overlay* in the following way:

Definition 5. Let $M := \{M_1, \ldots, M_k\}$ be a set of k subsets of $\varOmega$ with $P_\rho(M_s) > 0$ for $s = 1, \ldots, k$. Let $\varTheta$ be a Voronoi tessellation of $\varOmega$ with k partitions $\varTheta_s$. Then the overlay of $\varTheta$ and M with respect to P_ρ is given by

$$\mathrm{overlay}_{P_\rho}(\varTheta, M) := \sum_{s=1}^{k} P_\rho(M_s \cap \varTheta_s), \tag{4}$$

whereas the overlap of M with respect to P_ρ is defined by

$$\mathrm{overlap}_{P_\rho}(M) := \sum_{s=1}^{k} \frac{P_\rho(M_s \cap \bigcup_{p \neq s} M_p)}{P_\rho(\bigcup_{p=1}^{k} M_p)}. \tag{5}$$

For an interpretation, the value $\mathrm{overlay}_{P_\rho}(\varTheta, \varDelta)$ indicates the covering quality for a given $\varTheta$. If $\mathrm{overlay}_{P_\rho}(\varTheta, \varDelta) = 1$ then we call ($\varTheta$,$\varDelta$) a fully covering k-box-decomposition. Note that, if additionally $\varTheta$ has an optimal decomposition error with respect to ρ, $\varDelta$ is an optimal box-decomposition in the above strict sense. Suppose now that ($\varTheta$,$\varDelta$) is an approximate k-box decomposition of $\varOmega$, than the following simple rules describe the Voronoi partitions $\varTheta_s$:

$$\text{IF } \forall\, i = 1, \ldots, q \quad x_i \in \varDelta_{s_i} := [l_{s_i}, r_{s_i}] \text{ THEN } x \in \varTheta_s.$$

It is obvious that these description rules are not complete. For a good covering, i.e., one with $\text{overlay}_{P_\rho}(\Theta, \Delta) \approx 1$, this might be no problem. But otherwise one has to define additional rules such as how to deal with the configurations $x \in \Omega$ that can not be assigned via Δ.

Instead of boxes, one might also think of approximating the Voronoi partitions by using more complex geometrical objects. But the price to be paid for a possibly better approximation quality is a more complex description.

Parameter embedding. Metastability goes intimately with the undesirable effect of *trapping* within any Monte-Carlo simulation. In order to avoid such an occurrence, one may embed the given problem into a family of problems with flow $\Phi_V^{\tau;s}$ in terms of an embedding parameter $s \in [0, 1]$ such that, at $s = 0$, only a few metastable subsets need to be identified, whereas at $s = 1$ a rich structure of conformations might arise. Two types of embedding are in quite common use: *temperature embedding* and *potential embedding*. Upon examining the equations of motion, one immediately sees that, in the context of HMC, temperature embedding can be realized by the following flow:

$$\Phi_V^{\tau;s} = \Phi_{sV}^{s^{-2}\tau}, \tag{6}$$

which requires a scaling of both the potential and the time step of propagation [21].

Any kind of embedding stimulates the idea of a hierarchical algorithm consisting of the following steps:

1. simulate the molecular system for a specific parameter (say, high temperature), which causes the flow to overcome specific energy barriers,
2. identify metastable subsets,
3. increase the parameter (say, lower the temperature), but restrict the simulation to one of the metastable subsets. Go to 1.

This algorithm will generate a hierarchy of subsets that can be sampled independently at each level. The restriction of an HMC-simulation to a given metastable subset C_l requires only a slight modification of the Markov kernel K to K_l [8,10]. The additional rule is that any configuration outside the subset C_l will be rejected. Detailed balance still holds for this modified Markov kernel, so that K_l is still reversible [8]. Since C_l is metastable, only a few rejections will be expected with respect to the new rule. Moreover, trapping should thus be avoided, since energy barriers towards all other metastable subsets can be ignored. A further exploitation of this embedding structure is given in [10], where an uncoupling/coupling technique has been suggested and worked out.

A schematic diagram of such a hierarchy is given in Figure 2. As can been seen there, each cluster needs to be described by appropriate boundaries. To save computer time over the whole simulation, one is interested in simple descriptions in terms of a minimal necessary set of variables. Reduction of variables is a typical task for statistical methods, like e.g. discriminant analysis. As already mentioned in the Introduction, however, we describe the configurational space in terms of cyclic torsion angles. As conventional discriminant analysis is usually not ready to work for cyclic data, we will introduce the new concept of *discriminating variables* in Section 3 and suggest a heuristical algorithm for their automatic determination.

□

Fig. 2. Hierarchical scheme of clustering combined with parameter embedding. The numbers denote metastable conformations at different levels of the hierarchical embedding scheme.

2 From Self-Organizing Maps to Box Maps

In the following we will work out details of our recently developed Self-Organizing Box Maps (SOBM) algorithm [12], an extension of the traditional SOM algorithm. The SOBM algorithm permits to compute approximate k-box decompositions of Ω.

2.1 Point concept: Self-Organizing Maps

The computation of a Voronoi tessellation so that the corresponding decomposition error according to Eq. (2) is minimized for given parameter k is often too expensive. Therefore one often only tries to find a codebook $W := \{W_1, \ldots, W_k\} \subset \Omega$ such that the distances between the configurations from the sampling and their nearest codebook vector W_s are minimized, i.e., such that the *distortion value*

$$\hat{\vartheta}[W] := \sum_{s=1}^{k} \int_{x \in \Theta_{W_s}} \mathrm{dist}(x, W_s)\rho(x)dx \tag{7}$$

is minimized, where

$$\Theta_{W_p} := \{x \in \Omega \,|\, \mathrm{dist}(x, W_p) = \min_{s=1,\ldots,k} \mathrm{dist}(x, W_s)\}. \tag{8}$$

If we can assure a unique assignment of each vector $x \in \Omega$ in Eq. (8), then obviously $\Theta_W := \{\Theta_{W_1}, \ldots, \Theta_{W_k}\}$ is a Voronoi tessellation of Ω with a nearly minimal decomposition error.

But even the exact optimization of Eq. (7) is often too expensive: rather one has to use one of several heuristic algorithms that compute an approximate solution [20,17,1]. One powerful method is KOHONEN'S Self-Organizing Maps (SOM) algorithm. Although it has been shown [18] that the codebook vectors produced by the basic SOM algorithm in general do not exactly coincide with the optimum of Eq. (7) the algorithm usually produces fast and good solutions even for high-dimensional Ω. It can be easily adapted to the case of cyclic data and has the feature of topology approximation. Further the decomposition results are rather robust under small changes of the number k.

We first give a short general description of the SOM algorithm—for an exhaustive presentation see [19]—before we focus on the adaptations that are necessary to use it with cyclic data. For q torsion angle variables, each map is formed by a q-dimensional input-layer that is fully connected with the usually two-dimensional Kohonen layer, which is a neural $n \times m$ grid G with rectangular or hexagonal topology and $k = nm$ grid neurons. The coordinate tuple of each neuron s on the grid is denoted by $z_s \in G$ and each neuron s is uniquely related to a q-dimensional codebook vector W_s. After a suitable initialization of the codebook vectors, the SOM is trained in L time steps by a repeated presentation of vectors of the q-dimensional input space Ω according to a probability distribution P_ρ. For each presented input vector the SOM computes the so called winner neuron and its neighboring neurons on the grid and adapts the related codebook vectors, such that the distance to the input vector is reduced. To achieve convergence, the rate of the distance reduction $\alpha : \{0,\dots,L\} \to [0,1]$ — called learning rate — and the width of the neighborhood of the winner neuron, the so called neighborhood radius function $\gamma : \{0,\dots,L\} \to \mathbf{R}_0^+$, shrink to zero with time.

By construction it is obvious that after a suitable number of training steps the codebook vectors that are related to neighboring neurons on the grid, are neighboring in the input space according to the chosen distance function. Therefore the codebook vectors not only determine — via Eq. (8) — a Voronoi tessellation of Ω, but also approximate the topology of the input space via the neighborhood structure of the grid.

In the following we describe the initialization of the codebook vectors, the definition of the winner neuron together with its grid neighborhood and the specification of the codebook adaptation rule for cyclic input data.

Initialization. We suggest to choose the initial values $W_1(0),\dots,W_k(0)$ as approximately P_ρ-distributed random vectors with $W_s(0) \in \Omega$ for $s = 1,\dots,k$.

Winner neuron and grid neighborhood. Let $x = (x_1,\dots,x_q) \in \Omega$ be an arbitrary input vector and $W_1,\dots,W_k \in \Omega$ the actual codebook vectors of the SOM. Then we call neuron $p \in \{1,\dots,k\}$ the *winner neuron* for input x, if

$$\mathrm{dist}(x, W_p) = \min_{s\in\{1,\dots,k\}} \mathrm{dist}(x, W_s)\,. \tag{9}$$

Note that Eq. (9) is equivalent to $x \in \Theta_{W_p}$, if Θ_{W_p} is defined according to Eq. (8). In the case of more than one neuron, which match Eq. (9), various strategies are used to assure uniqueness. Sometimes the winner is chosen randomly, but usually the one with the lowest index is taken as the actual winner.

To determine the neighboring neurons of the winner neuron, one has to specify a grid distance function $\eta : G \times G \times \mathbf{R}^+ \to [0,1]$. Usually one uses either the bubble grid distance

$$\eta_{\mathrm{bubble}}(z_s, z_p, \gamma) := \begin{cases} 0 \text{ if } \left\|z_s - z_p\right\|_2 \le \gamma \\ 1 \text{ else,} \end{cases}$$

or the Gaussian grid distance

$$\eta_{\mathrm{gaussian}}(z_s, z_p, \gamma) := 1 - \exp\left(-\frac{\left\|z_s - z_p\right\|_2^2}{2\gamma^2}\right)$$

where γ denotes the actual neighborhood radius and $\| \,\|_2$ the two-dimensional Euclidean distance. A neuron s belongs to the neighborhood of winner neuron p if $\eta(z_s, z_p, \gamma) < 1$. If we choose $\eta_{gaussian}$, then the neighborhood of each neuron covers *all* grid neurons.

Codebook adaptation rules. Let neuron p be the winner neuron for input $x(t) = (x_1(t), \dots, x_q(t)) \in \Omega$ at time $t \in \{0,\dots,L-1\}$ and $W_1(t),\dots,W_k(t) \in \Omega$ the actual codebook vectors. Further let $\alpha(t)$ and $\gamma(t)$ be two time-dependent linear or log-linear functions, respectively, decreasing to zero, with $\alpha(0) \le 1$ and $\gamma(0) \le \frac{\min\{n,m\}}{2}$. Usually $\gamma(t)$ is called the *neighborhood-radius* function, respectively.

Then, new codebook vectors are computed as:

$$W_s(t+1) := W_s(t) + \alpha(t)\,\mathrm{neigh}(z_s, z_p, t)\,(x(t) - W_s(t)),\ s = 1,\dots,k. \tag{10}$$

with $\mathrm{neigh}(\mathrm{z_s}, \mathrm{z_p}, \mathrm{t}) := 1 - \eta(\mathrm{z_s}, \mathrm{z_p}, \gamma(\mathrm{t}))$.

Cyclic boundary transformation (CBT) rules. When dealing with cyclic data, the input vector or the old codebook cector, respectively, may need to be transformed first, before the new codebook vector can be computed according to Eq. (10):

(i) IF $W_{s_i}(t) \geq 0$ AND $x_i(t) < 0$ AND $\mathrm{abs}(W_{s_i}(t)) + \mathrm{abs}(x_i(t)) > \pi$
THEN $x_i(t) := x_i(t) + 2\pi$.

(ii) IF $W_{s_i}(t) < 0$ AND $x_i(t) \geq 0$ AND $\mathrm{abs}(W_{s_i}(t)) + \mathrm{abs}(x_i(t)) > \pi$
THEN $W_{s_i}(t) := W_{s_i}(t) + 2\pi$,

with $\mathrm{abs}(x_i) := \sqrt{x_i^2}$ for $i = 1, \dots, q$. Note that after the new codebook vector has been computed, eventually it must also be transformed such that each component $W_{s_i}(t+1)$ is inside the interval $[-\pi, \pi]$.

2.2 Set concept: Self-Organizing Box Maps

The basic idea of the Self-Organizing Box Maps (SOBM) algorithm is to compute *codebook boxes* $\hat{W}_s := (\hat{W}_{s_1}, \dots, \hat{W}_{s_q}) \in \mathrm{BOX}(\Omega)$ with $\hat{W}_{s_i} := [l_{s_i}, r_{s_i}]$ instead of codebook vectors $W_s = (W_{s_1}, \dots, W_{s_q}) \in \Omega$. This is done is such a way that each codebook box is a good box approximation of its implicitly defined Voronoi partition $\hat{\Theta}_s := \Theta_{\hat{W}_s}$ with respect to ρ.

Definition 6. Let B be a box in Ω and A an arbitrary non-void subset of Ω. Then B is called a *box approximation* of A with respect to ρ, if $P_\rho(B \setminus A) + P_\rho(A \setminus B) \ll 1$.

Obviously, this change of concept induces changes of the SOM algorithm, which we arrange here:

Initialization. Let $W_1(0), \dots, W_k(0)$ be different initial values for the codebook vectors of the traditional SOM, e.g., approximately P_ρ-distributed random vectors with $W_s(0) \in \Omega$ for $s = 1, \dots, k$. For our extended algorithm, we choose $\hat{W}_s(0) := \bigotimes_{i=1}^q [l_{s_i}(0), r_{s_i}(0)]$ with $l_{s_i}(0) = W_{s_i}(0)$ and $r_{s_i}(0) = W_{s_i}(0) + \epsilon \leq \pi$ in terms of a small positive value ϵ, the initial width of the interval, such that $\hat{W}_s \cap \hat{W}_p = \emptyset$ for all $s, p \in \{1, \dots, k\}$.

Winner neuron. We suppose that the problem specific q-dimensional distance function dist(x, y) with $x, y \in \Omega$ can be written as a function of q one-dimensional distance measures $d_i(x_i, y_i)$, which means that $\mathrm{dist}(x, y) := f(d_1(x_1, y_1), \dots, d_q(x_q, y_q))$. Note that many popular distance measures, as e.g., the Euclidean distance, just exhibit this feature. In the case of our suggested distance measure (see Eq. (3)) we have:

$$f(d_1, \dots, d_q) := (\sum_{i=1}^q d_i)^{1/2}$$

$$\text{with} \quad d_i(x_i, y_i) := (\sin(x_i) - \sin(y_i))^2 + (\cos(x_i) - \cos(y_i))^2.$$

Obviously we need a distance measure DIST, that permits to compute the distance between an input vector $x \in \Omega$ and codebook boxes $\hat{W}_s \in \mathrm{BOX}(\Omega)$. For that purpose, we suggest

$$\mathrm{DIST}(x, \hat{W}_s) := f(\hat{d}_1(x_1, \hat{W}_{s_1}), \dots, \hat{d}_q(x_q, \hat{W}_{s_q}))$$

with

$$\hat{d}_i(x_i, \hat{W}_{s_i}) := \begin{cases} 0 & \text{if } x_i \in \hat{W}_{s_i} \\ \min\{d_i(x_i, l_{s_i}), d_i(x_i, r_{s_i})\} & \text{else.} \end{cases}$$

Then the winner neuron p has to match a condition analogous to Eq. (9):

$$\mathrm{DIST}(x, \hat{W}_p) = \min_{s \in \{1,\dots,k\}} \mathrm{DIST}(x, \hat{W}_s)\,. \tag{11}$$

Obviously we can use Eq. (11) to define for each codebook box $\hat{W}_s$ the corresponding Voronoi partition $\hat{\Theta}_s := \Theta_{\hat{W}_s}$ analogously to Eq. (8).

Codebook adaptation rules. In analogy to the SOM algorithm, the SOBM algorithm has to adapt the codebook *boxes*. This will be done by the following rules:

$$\begin{aligned} l_{s_i}(t+1) := \; & l_{s_i}(t) \\ & + g(l_{s_i}(t), r_{s_i}(t), x_i(t))\; \alpha(t)\, \mathrm{neigh}(z_s, z_p, t)\, (x_i(t) - l_{s_i}(t)) \\ & - \alpha(t)\; c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

$$\begin{aligned} r_{s_i}(t+1) := \; & r_{s_i}(t) \\ & + g(-r_{s_i}(t), -l_{s_i}(t), -x_i(t))\; \alpha(t)\, \mathrm{neigh}(z_s, z_p, t)\, (x_i(t) - r_{s_i}(t)) \\ & + \alpha(t)\; c(l_{s_i}(t), r_{s_i}(t)) \end{aligned}$$

with a linear function $g : [-\pi, \pi]^3 \to [0, 1]$ described in the Appendix and a special function $c : \mathbf{R}^2 \to \mathbf{R}$ independent of the input $x(t)$ (see [12] for a justification of the functions g and c). As shown in [12], the choice

$$c(a, b) := \frac{1}{6}\, \iota[a, b], \tag{12}$$

where $\iota([a, b])$ is the width of the interval $[a, b]$, guarantees that $\hat{W}_s(L)$ is a nearly optimal box approximation of $\hat{\Theta}_s(L)$ if we choose time L large enough.
For cyclic input data we again have to consider CBT-rules, see the Appendix for details.
There will be a problem, if the width of the interval $[l_{s_i}(t), r_{s_i}(t)]$ is nearly 2π. Then one observes sometimes the artifact that left and right boundaries interchange, so that the interval becomes “too small”. In this case the adaptation step has to be skipped and the interval automatically reduces to $[-2\pi + \epsilon, 2\pi - \epsilon]$ as the new value of $\hat{W}_{s_i}(t + 1)$.

One easily checks, that if the SOBM algorithm is successful, i.e., if the computed final codebook boxes $\hat{W}_s$ are good box approximations of the corresponding Voronoi partitions $\hat{\Theta}_s$, then $(\hat{\Theta}\,, \hat{W}) := (\hat{\Theta}_1, \dots, \hat{\Theta}_k\,, \hat{W}_1, \dots, \hat{W}_k)$ is a good approximate k-box-decomposition.

3 Discriminating variables

As usual in data mining problems, the high-variableal configuration space Ω here is also very sparse with respect to ρ, i.e., the subset $\Omega_\rho := \{\omega \in \Omega \,|\, \rho(\omega) > 0\}$ is much smaller than Ω.

Therefore often only very few variables are necessary to separate the given Voronoi partitions Θ_s with respect to ρ, i.e., to separate the sets $\Theta_{s,\rho} := \{\omega \in \Theta_s \,|\, \rho(\omega) > 0\}$. In this section we give a formal definition of discriminating variables with respect to a given Voronoi tessellation. Afterwards we show how an approximate box-decomposition can be used to compute a heuristic solution automatically.

Definition 7. Let $a_1, \ldots, a_q$ denote the q variables (i.e., torsion angles) spanning Ω and let $I \subset \{1, \ldots, q\}$ any index subset. Then we define $A(I) := \{a_i \,|\, i \in I\}$ as the corresponding variable subset and $I^c := \{1, \ldots, q\} \setminus I$ as the complement of I. By $\Omega(I)$ we denote the $|I|$-dimensional subspace of Ω spanned by the variables $a_i \in A(I)$. Further $P_{\rho(I)}$ and $\rho(I)$ denote the projections of P_ρ and ρ on $\Omega(I)$. For any set $M := \{M_1, \ldots, M_k\}$ with $M_s \in \Omega$, we set $M(I) := \{M_1(I), \ldots, M_k(I)\}$ where $M_s(I)$ denotes the projection of M_s on $\Omega(I)$ for $s = 1, \ldots, k$.

Definition 8. Suppose we have a Voronoi tessellation $\Theta := \{\Theta_1, \ldots, \Theta_k\}$ of Ω and a subset of indices $I \subset \{1, \ldots, q\}$.
(a) We call the variable set $A(I^c)$ *redundant* for (Ω, ρ, Θ), if $\Theta_{\rho(I)} := \{\Theta_{1,\rho(I)}, \ldots, \Theta_{k,\rho(I)}\}$ is a Voronoi tessellation of $\Omega_{\rho(I)}$.
(b) We call the variable set $A(I^c)$ *maximally redundant* for (Ω, ρ, Θ), if there exists no subset $J \subset \{1, \ldots, q\}$ such that $A(J^c)$ is redundant for (Ω, ρ, Θ) and $|J| > |I|$.
(c) We call variable a_i of Ω a *univariate discriminating variable* of (Ω, ρ, Θ), if $A(\{i\})$ is not redundant for (Ω, ρ, Θ).
(d) We call the variables $a_i \in A(I)$ *multivariate discriminating variables* of (Ω, ρ, Θ), if $A(I^c)$ is maximally redundant for (Ω, ρ, Θ).

Obviously our task is to find a maximally redundant variable set $A(I^c)$ for Ω, so that we can describe and separate the given Voronoi partitions by rules based only on the corresponding multivariate discriminating variables.

Unfortunately the above definition cannot be directly realized. Besides the fact that it is very expensive to verify whether $\Theta_{\rho(I)}$ is a Voronoi tessellation or not, especially in practical applications one has usually to accept slight overlaps between the partitions $\Theta_{s,\rho(I)}$, if one wants to reduce the number of variables at all. Therefore we give a softer definition which depends on a parameter $\delta \in [0, 1]$, called sensitivity factor:

Definition 9. Let $M := \{M_1, \ldots, M_k\}$ be a set of subsets of Ω with $P_\rho(M_s) > 0$ for each $s \in \{1, \ldots, k\}$ and $I \subset \{1, \ldots, q\}$. Then we call the variable set $A(I^c)$ *δ-redundant* for (Ω, ρ, M), if $\mathrm{overlap}_{P_{\rho(I)}}(M(I)) <= \mathrm{overlap}_{P_\rho}(M) + \delta$. A *maximally δ-redundant* variable subset is defined analogously.

Suppose now we have an approximate k-box-decomposition (Θ,Δ) of Ω with respect to ρ. Let the variable set $A(I^c)$ be δ-redundant for (Ω, ρ, Δ) for a small δ. Since we have $\mathrm{overlap}_{P_\rho}(\Delta) \approx 0$, we have also $\mathrm{overlap}_{P_{\rho(I)}}(\Delta(I)) \approx 0$. If we use the following definition

Definition 10. Let $I \subset \{1, \ldots, q\}$ be any index subset and $\Delta(I)$ be a set of k boxes $\Delta(I)_1, \ldots, \Delta(I)_k \in BOX(\Omega(I))$. Without loss of generality we suppose that $I = \{1, \ldots, j\}$ with $j \leq q$ and that $\Omega \subset \bigotimes_{i=1}^{q} [l_i, r_i]$. We call $\bar{\Delta}(I)_s := \Delta(I)_s \times \bigotimes_{i=j+1}^{q} [l_i, r_i]$ the *extension* of $\Delta(I)_s$ to Ω. Set $\bar{\Delta}(I) := \{\bar{\Delta}(I)_1, \ldots, \bar{\Delta}(I)_k\}$.

One easily checks that $\mathrm{overlap}_{P_\rho}(\bar{\varDelta}(I)) \approx 0$ and therefore ($\varTheta$,$\bar{\varDelta}(I)$) is an approximate k-box-decomposition of $\varOmega$ with respect to ρ. Since the value $\mathrm{overlay}_{P_\rho}(\varTheta, \bar{\varDelta}(I))$ depends on the chosen sensitivity factor δ, we have to adjust δ, such that the approximation quality of ($\varTheta$,$\bar{\varDelta}(I)$) is optimal. Then we can describe the Voronoi partitions $\varTheta_s$ based only on the variables that are necessary for a suitable discrimination:

$$\text{IF } \forall\, i \in I \quad x_i \in \varDelta_{s_i} \text{ THEN } x \in \varTheta_s.$$

Note that $\bar{\varDelta}(I)_{s_i} = \varDelta_{s_i}$ for $i \in I$. The computation of a maximal δ-redundant variable set $A(I^c)$ for $(\varOmega, \rho, \varDelta)$ has combinatorial complexity. Therefore we suggest to use the following heuristic for a given δ:

```
Compute $\varphi(i) := \mathrm{overlap}_{P_{\rho(\{i\})}}(\varDelta(\{i\}))$ for each $i \in \{1,\dots,q\}$.
Set $D := \{1,\dots,q\}$ and $I := \{1,\dots,q\}$.
WHILE $D \neq \emptyset$ DO
    Choose $i \in D$ with minimal value $\varphi(i)$.
    Set $D := D \setminus \{i\}$ and $I := I \setminus \{i\}$.
    IF $A(I^c)$ is not $\delta$-redundant for $\varOmega$ with respect to $\varDelta$ THEN $I := I \cup \{i\}$.
WEND
```

Note that $\varphi(i)$ is large for univariate discriminating variables a_i. The optimal sensitivity factor δ has to be determined in an iterative process – to be described now. For simulation efficiency and quality evaluation reasons we want to compute simple rules that describe and separate the clustering $\{C_1,\dots,C_\kappa\}$ of $\varOmega$. As we have seen above this can be easily achieved by using the box concept. Therefore we need to define some box-clustering:

Definition 11. Let $C := \{C_1,\dots,C_\kappa\}$ be a dynamical clustering of $\varOmega$ based on the Voronoi tessellation $\varTheta := \{\varTheta_1,\dots,\varTheta_k\}$, with $C_s = \bigcup_{p\in J_s} \varTheta_p$ and p.w. disjoint index sets $J_s \subset \{1,\dots,k\}$ for $s = 1,\dots,\kappa$. Further let $\varDelta := \{\varDelta_1,\dots \varDelta_k\}$ be a set of boxes in $\varOmega$ such that each $\varDelta_s \in BOX(\varOmega)$ is a box approximation of $\varTheta_s$. Set $C_s^{\varDelta} := \bigcup_{p\in J_s} \varDelta_p$. Then we call $C^{\varDelta} := \{C_1^{\varDelta},\dots,C_\kappa^{\varDelta}\}$ the corresponding *box clustering* with respect to $\varDelta$.

An algorithm to compute a dynamical clustering together with simple descriptions based on a suitable corresponding box-clustering will consist of the following steps:

1. Compute an approximate k-box-decomposition ($\varTheta$,$\varDelta$) of $\varOmega$.
2. Compute a dynamical clustering C based on the Voronoi tessellation $\varTheta$.
3. Compute descriptions based on the corresponding box-clustering $C^{\varDelta}$.

The approximation quality of the box-clustering is given by $\mathrm{overlay}_{P_\rho}(C, C^{\varDelta})$ and depends obviously on the value $\mathrm{overlay}_{P_\rho}(\varTheta, \varDelta)$. Practical experience shows that with increasing q the approximation quality usually becomes worser. As a simple solution to reduce the number of variables and so to improve the approximation quality, one can think about using only the discriminating variables of $\varOmega$ and to compute $C := C(I)$ based on ($\varTheta(I)$,$\varDelta(I)$), where $A(I^c)$ is a maximal δ-redundant variable set for $(\varOmega, \rho, \varDelta)$. Unfortunately, with larger q one has to choose a large δ to reduce the number of variables sufficiently. But then the value $\mathrm{overlap}_{P_{\rho(I)}}(\varDelta(I))$ is usually not longer approximately 0 and therefore $\mathrm{overlap}_{P_{\rho(I)}}(C^{\varDelta(I)})$

is also no longer approximately 0. But then the computed descriptions do not separate the clusters very well. Therefore we suggest an iterative process to compute an optimal value δ, such that $\mathrm{overlay}_{P_{\rho(I)}}(C, C^{\Delta(I)})$ is maximized, while $\mathrm{overlap}_{P_{\rho(I)}}(C^{\Delta(I)})$ is still approximately 0:

```
Compute an approximate $k$-box-decomposition ($\Theta$,$\Delta$) of $\Omega$.
Choose a small sensifity factor $\delta > 0.0001$, e.g., $\delta := \max\{\varphi(i) \,|\, i = 1, \ldots, q\}/2$.
WHILE $\delta > 0.0001$ DO
    Compute a maximal $\delta$-redundant variable set $A(I^c)$ for $(\Omega, \rho, \Delta)$.
    Compute a dynamical clustering $C$ based on the Voronoi tessellation $\Theta(I)$.
    Compute $\mathrm{op}(I) := \mathrm{overlap}_{P_{\rho(I)}}(C^{\Delta(I)})$ and $\mathrm{oy}(I) := \mathrm{overlay}_{P_{\rho(I)}}(C, C^{\Delta(I)})$.
    IF $\mathrm{op}(I) < 0.1$ THEN
        Set $\delta := \delta - \delta/10$.
    ELSE
        Set $\delta_{\mathrm{old}} := \delta$ and $\delta := \delta + \delta/20$.
        WHILE $\delta_{\mathrm{old}} <> \delta$ DO
            Set $\mathrm{op}_{\mathrm{old}} := \mathrm{op}(I)$ and $\mathrm{oy}_{\mathrm{old}} := \mathrm{oy}(I)$.
            Compute a maximal $\delta$-redundant variable set $A(I^c)$ for $(\Omega, \rho, \Delta)$.
            Compute a dynamical clustering $C$ based on the Voronoi tessellation $\Theta(I)$.
            Compute $\mathrm{op}(I) := \mathrm{overlap}_{P_{\rho(I)}}(C^{\Delta(I)})$ and $\mathrm{oy}(I) := \mathrm{overlay}_{P_{\rho(I)}}(C, C^{\Delta(I)})$.
            IF $(\mathrm{op}(I) < \min\{\mathrm{op}_{\mathrm{old}} + 0.03, 0.1\})$ AND $(\mathrm{oy}(I) < \mathrm{oy}_{\mathrm{old}})$ THEN
                Set $\delta_{\mathrm{old}} := \delta$ and $\delta := \delta + \delta/20$.
            ELSE
                Set $\delta := \delta_{\mathrm{old}}$.
            IFEND
        WEND
        Compute a maximal $\delta$-redundant variable set $A(I^c)$ for $(\Omega, \rho, \Delta)$.
        Compute a dynamical clustering $C$ based on the Voronoi tessellation $\Theta(I)$.
        Compute descriptions based on the corresponding box-clustering $C^{\Delta(I)}$.
        Set $\delta := 0$
    IFEND
WEND
```

4 Numerical Results

The above SOBM algorithm is now exemplified within the whole conformation analysis algorithm for molecular systems. All molecules, for which HMC results are presented, were parametrized by the MMFF force field [16]. As noted in (6), the sampling of a thermodynamic distribution at various temperatures within a temperature embedding can be realized by a correlated scaling of time steps and potential [21].

Apart from pentane, the Hybrid Monte Carlo (HMC) simulations were performed with time steps $\tau = 2.24$ femtoseconds (fs) or $\tau = 1.83$fs. Each new configuration is generated by a propagation of the system over 40 time steps. Each simulation consists of 5 independent Markov chains. Every second configuration is stored. Convergence of the HMC-simulation is reached, as soon as the Gelman and Rubin quotient [14] is sufficiently close to the value 1. At least n = 20000 configurations turned out to be necessary for each simulation run.

Although the computation of the approximate box-decomposition can be done automatically, one has to fix some parameters. Obviously the resolution parameter k is the most important one, because the quality of the decomposition depends severely on it. Fortunately, both the SOM and the SOBM-algorithm are quite robust against small changes of k, so that the following iterative strategy may be successful:

1. Choose a small value $k = k_0$.

2. Compute an approximate box-decomposition for k using a hybrid algorithm: First compute k codebook vectors with the SOM algorithm. Then use the codebook vectors to initialize the k codebook boxes of the SOBM and adapt these boxes sufficiently fine.

3. Test whether the number of codebook boxes is large enough to guarantee an acceptable fine decomposition of Ω. Heuristically, we regard k to be large enough, if more than 10% of the final boxes $\hat{W}_s(L)$ contain no vector $x \in \Omega$ with $\rho(x) > 0$.

4. If the number of codebook boxes is too small, increase k and go to step 2.

The use of the above combined SOM and SOBM algorithm speeds up the decomposition process, because the adaptation of boxes needs at least twice as much computing time as the adaptation of points. Throughout our numerical experiments we have set the following parameters:

SOM algorithm: First perform $\max\{n, 5000\}$ ordering steps with $\alpha(0) := 0.9$, $\eta := \eta_{gaussian}$ and $\gamma(0)$ choosen to be half the radius of the selected map size, then perform $\max\{n, 15000\}$ convergence steps with $\alpha(0) := 0.1$, $\eta := \eta_{bubble}$ and $\gamma(0) := 1$.

SOBM algorithm: Use computed SOM codebook vectors as initialization of SOBM codebook boxes, then perform $\max\{5n, 50000\}$ convergence steps with $\alpha(0) := 0.1$, $\eta := \eta_{bubble}$ and $\gamma(0) := 1$.

4.1 Box-decomposition for n-pentane molecule

In what follows, we show the results of our new SOBM algorithm for the simple n-pentane molecule. We have analyzed the configurational space with respect to the two central torsion angles defined by carbon quadrupels. The HMC simulation was performed at temperature $T = 800\ K$ and with simulation time step $\tau = 2.83$fs. Based on the computed box decomposition, our cluster algorithm identifies five possible metastable conformations (for more details about the n-pentane and its dynamical behaviour see [10]). Figure 3 shows the final codebook boxes computed by the SOBM algorithm for each identified cluster. Obviously the overlap between boxes of different clusters is small. To measure the quality of our approximate box-decomposition, we have to check the overlay between the 99 boxes and the implicitly defined

□

Fig. 3. Box-decomposition of n-pentane: Final boxes from SOBM algorithm for the 5 identified clusters (overlay=86.3%, overlap=0.3%). Compare Fig. 4.

□

Fig. 4. Clustering of n-pentane: Metastable conformations based on box-decomposition. Compare Fig. 3.

Voronoi tessellation. In Figure 4 Voronoi partitions for each identified metastable conformation are visualized with respect to the sampling probability density ρ. A comparison with Figure 3 shows that the covering is satisfactory and indeed the computed overlay is 86.3%. But even for such a large overlay one observes rather big "holes", i.e., uncovered areas of the Voronoi partitions. This occurrence may cause problems during the temperature embedding process; if one only uses the simple box description rules it is possible that too many configurations are rejected, which implies that the Markov chain converges poorly. Figure 3 nicely shows the topology approximation feature of the SOBM algorithm: areas of the input space are discretized with different resolutions, i.e. by a different number of boxes. The greater the distributational variation of the sampling configurations, the finer is the resolution.

4.2 HIV-protease inhibitor

□□

Fig. 5. Two conformations of HIV-protease inhibitor: Average configurations for two out of six identified metastable conformations at temperature level $T = 1000\,K$.

The inhibitor VX-478 of the enzyme HIV-protease consists of 70 atoms. Each configuration can be roughly reconstructed by 34 torsion angles and corresponding equilibrium bonds and angles. In order to illustrate the Perron cluster analysis, we present here the results of two levels out of a hierarchical simulation protocol corresponding to a temperature embedding - see Table 1), where eigenvalue spectra, coupling-matrices, overlays and numbers of discriminating dihedrals are arranged.

T[K]	spectrum	coupling matrix	ol [%]	ndv
1500	1.000 0.967 0.870 0.832	0.994 0.006 0.038 0.962	19.8	22
1000	1.000 0.979 0.967 0.915 0.906	0.976 0.024 0.000 0.008 0.982 0.010 0.000 0.036 0.964	17.2	24
	1.000 0.997 0.976 0.948 0.945	0.976 0.022 0.003 0.003 0.995 0.002 0.000 0.001 0.998	20.2	19

Table 1. Hierarchical temperature embedding for HIV-protease inhibitor: overlay (ol), number of discriminating variables (ndv).

The cluster analysis at level $T = 1000\,K$ decomposes each of the two conformations at level $T = 1500\,K$ into 3 conformations clearly indicated by spectral gaps each. The overlay value of all simulations was close to 20.0%. The overlap value was fixed at 0.0%. The number of discriminating dihedrals was found in each SOBM analysis to be roughly 20 out of 34. This number varies because each metastable conformation reflects different energy barriers. Figure 5 shows average configurations for two out of the six identified conformations at $T = 1000\,K$. For comparison the two average configurations are aligned in a plane defined by three common atoms. As it turns out, the different orientation of the functional groups due to electrostatic and due to Lennard-Jones interactions seem to be the main reasons for the observed differences of the conformations.

4.3 Virtual screening project

Our SOBM algorithm has been successfully used within a Virtual Screening (VS) project. For illustration, we here have applied the VS to 200 small molecules with different number of atoms (< 100) i.e. different variables q of the corresponding configurational space Ω. The aim of the project was to explore any metastable conformations of the given molecules at high temperature.

Figure 6 shows the number of clusters identified by Perron cluster analysis as a statistic over all molecules.

We have observed a quite small overlap for nearly all computed box clusterings (top of Figure 7). For smaller molecules also the overlay of the box clusterings is good (bottom of Figure 7). The reason for the partially bad overlay for larger molecules is the fact that the computed δ-redundant variable sets are relatively “too small”, i.e., the ratio of the number of discriminating variables vs. the total number of torsion angles is sometimes “too high” for larger molecules (top of Figure 8). Finally Figure 8 (bottom) shows the average CPU-times for the computation of the discretizations and clusterings on a SUN Ultra E3000 ordered by the number of torsion angles of the analyzed molecules.

□

Fig. 6. Virtual Screening of 200 molecules: Number of identified metastable conformations

□□

Fig. 7. VS of 200 molecules: Overlap (top) and overlay (bottom) for molecules with different number of torsion angles

□□

Fig. 8. VS of 200 molecules: Top: ratio of discriminating angles vs.total number of torsion angles. Bottom: CPU-time on SUN Ultra E3000

Conclusion

The present paper describes in detail, how self-organized neural networks (SOM) can be utilized and extended (SOBM) to be of crucial importance for the actual computation of metastable conformations within a Perron cluster analysis. The performance of the algorithm as given herein is illustrated by biomolecular examples. The present version of our algorithm appears to be quite efficient in connection with hybrid Monte Carlo methods as worked out in [10]. Efforts to further increase its reliability and speed are already under investigation.

Acknowledgments. The authors thank M.Meyer and ANALYTICON DISCOVERY, POTSDAM, GERMANY for support within the virtual screening project. Two of us (T.G. and F.C.) were partially supported by RISK–CONSULTING, PROF. DR. WEYER, KÖLN, GERMANY and by the EUROPEAN REGIONAL DEVELOPMENT FUND (ERDF).

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 94–105, 1998.
2. A. Amadei, A.B.M. Linssen, and H.J.C. Berendsen. Essential dynamics of proteins. *Proteins*, 17, 1993.
3. P. Deuflhard, M. Dellnitz, O. Junge, and Ch. Schütte. Computation of essential molecular dynamics by subdivision techniques. In [4], pages 98–115.
4. P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, and R.D. Skeel, editors, *Computational Molecular Dynamics: Challenges, Methods, Ideas. Lecture Notes in Computational Science and Engineering*, volume 4. Springer, 1998.
5. P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in nearly uncoupled markov chains. *Linear Algebra and its Applications 315*, pages 39–59, 2000.
6. B.S. Duran and P.L. Odell. *Cluster Analysis*. Springer, Berlin, 1974.
7. U.M. Fayyad, G. Patetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.). *Advances in Knowledge Discovery and Data Mining*. AAAI Press / The MIT Press, California, 1996.
8. A. Fischer. An uncoupling-coupling technique for Markov Chain Monte Carlo methods. Preprint SC-00-04, Konrad–Zuse–Zentrum, Berlin. Available via http://www.zib.de/.
9. A. Fischer, F. Cordes, and Ch. Schütte. Hybrid Monte Carlo with adaptive temperature in mixed–canonical ensemble: Efficient conformational analysis of RNA. *J. Comp. Chem.*, 19(15):1689–1697, 1998.
10. A. Fischer, Ch. Schütte, P. Deuflhard, and F. Cordes. Hierarchical uncoupling-coupling of metastable conformations. ZIB-Report 01-03, Konrad–Zuse–Zentrum, Berlin. Available via http://www.zib.de/bib/pub/pw.
11. N.I. Fisher. *Statistical Analysis of Circular Data*. University Press, Cambridge, 1993.
12. T. Galliat and P. Deuflhard. Adaptive hierarchical cluster analysis by self-organizing box maps. ZIB-Report 00-13 (April 2000), Konrad–Zuse–Zentrum, Berlin. Available via http://www.zib.de/DataMining, 2000.
13. T. Galliat, W. Huisinga, and P. Deuflhard. Self-organizing maps combined with eigenmode analysis for automated cluster identification. In H. Bothe and R. Rojas, editors, *Proceedings of the 2nd International ICSC Symposium on Neural Computation*, pages 227–232. ICSC Academic Press, 2000.
14. A. Gelman and D.B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511, 1992.
15. A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.

16. T.A. Halgren. Merck molecular force field.i-v. *J. Comp. Chem.*, 17(5&6):490–641, 1996.
17. M. Van Hulle. *Faithful Representations and Topographic Maps*. John Wiley Sons, Inc., 2000.
18. T. Kohonen. Comparison of som point densities based on different criteria. *Neural Computation*, (11):2081–2095, 1999.
19. T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.
20. B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
21. Ch. Schütte. *Conformational Dynamics: Modelling, Theory, Algorithm, and Application to Biomolecules*. Habilitation Thesis, Dept. of mathematics und computer science, Free University Berlin, 1998. Available as ZIB-Report SC-99-18 via http://www.zib.de/bib/pub/pw/.
22. Ch. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comput. Phys., Special Issue on Computational Biophysics*, 151:146–168, 1999.

Appendix

Function g within Codebook adaptation rules for SOBM. We have to distinguish between normal and complementary intervals.

Case 1: The interval $\hat{W}_{s_i} := [l_{s_i}, r_{s_i}]$ is an interval with $l_{s_i} < r_{s_i}$. Then we define:

$$g(a,b,\psi) := \begin{cases} 1 & \text{if } \psi \notin [a,b] \wedge d_i(\psi,a) \leq d_i(\psi,b) \\ 0 & \text{if } \psi \notin [a,b] \wedge d_i(\psi,a) > d_i(\psi,b) \\ \frac{b-\psi}{\iota([a,b])} & \text{else.} \end{cases}$$

with $\iota([a,b]) := (b-a)$.

Case 2: The interval $\hat{W}_{s_i}$ is a complementary interval with $l_{s_i} > r_{s_i}$. Then we define:

$$g(a,b,\psi) := \begin{cases} 1 & \text{if } \psi \in [b,a] \wedge d_i(\psi,a) \leq d_i(\psi,b) \\ 0 & \text{if } \psi \in [b,a] \wedge d_i(\psi,a) > d_i(\psi,b) \\ \frac{2\pi+(b-\psi)}{\iota([a,b])} & \text{if } \psi \notin [b,a] \wedge \psi \geq a \\ \frac{b-\psi}{\iota([a,b])} & \text{else.} \end{cases}$$

with $\iota([a,b]) := 2\pi + (b-a)$.

Cyclic interval boundary transformation rules for SOBM. If $\hat{W}_{s_i} := [l_{s_i}, r_{s_i}]$ with $l_{s_i} > r_{s_i}$ or if x_i is not inside the complementary interval $\hat{W}_{s_i}$, i.e., $x_i \in [r_{s_i}, l_{s_i}]$, then we have to consider the earlier defined CBT rules, with $l_{s_i}(t)$ and $r_{s_i}(t)$ instead of $W_s(t)$. But if x is inside the complementary interval $\hat{W}_{s_i}$, i.e., $x_i \notin [r_{s_i}, l_{s_i}]$, one has to consider sligthly different transformation rules, because one has to assure that the boundaries are adapted towards the correct direction:

```
IF $g(l_{s_i}(t), r_{s_i}(t), x_i(t)) > g(-r_{s_i}(t), -l_{s_i}(t), -x_i(t))$ THEN
    Use the CBT rules for the adaptation of $l_{s_i}(t)$.
    IF $x_i(t) > r_{s_i}(t)$ THEN
        First set $x_i(t) := x_i(t) - 2\pi$, afterwards adapt $r_{s_i}(t)$ directly
        (i.e., without further transformation).
    ELSE
        Adapt $r_{s_i}(t)$ directly.
    ENDIF
ELSE
```

```
    Use the CBT rules for the adaptation of $r_{s_i}(t)$.
    IF $x_i(t) < l_{s_i}(t)$ THEN
        First set $x_i(t) := x_i(t) + 2\pi$, afterwards adapt $l_{s_i}(t)$ directly.
    ELSE
        Adapt $l_{s_i}(t)$ directly.
    ENDIF
ENDIF
```