Generation of Markov state models for the description of protein dynamics from molecular dynamics simulation data

Jan K. Bohrer
Institute of Physics, University of Freiburg, D-79104 Freiburg, Germany

(Dated: 2nd January 2018)

Markov state models can present a concise and at the same time sufficiently precise description of the essential dynamics of protein systems in terms of a transition network between a finite number of discrete metastable states. The model can be analyzed to reveal important physical properties and processes, including the stationary state occupation distribution, lifetime of geometrical structures (which determine the functionality of proteins) and the composition, rate and importance/dominance of possible folding pathways. Although molecular dynamics simulations describe dynamical processes of proteins in microscopic detail, they generate huge amounts of data which have to be interpreted. Markov state models can be used to interpret massive amounts of data and connect data from many (shorter) simulation trajectories. In this work, a general and modular procedure is presented how to build a Markov state model from molecular dynamics simulation data using a sequence of steps, including dimensionality reduction, geometric clustering and dynamic clustering. In the process, certain state of the art example methods for dimensionality reduction (Principal Component Analysis) and clustering (k-means, Density Based Clustering and Perron-Cluster-Analysis) are introduced.

I. INTRODUCTION

Due to progresses in hardware processing power and computational techniques, molecular dynamics (MD) simulations yield a detailed description of the dynamical behaviour of protein systems on microscopic level.[1–8] In the process however, huge amounts of data are generated, which have to be interpreted to extract the meaningful physical information. Markov state models (MSMs) can represent a concise and, at the same time, correct description of the essential dynamics of a protein system by memoryless jump processes between a finite number of discrete states. In the modelling process, it is possible to reduce massive amounts of simulation data (even merged from many single simulation trajectories of the same system) to an interpretable level in terms of a transition network between metastable states.

The way of defining the number and nature of those metastable states determines the level of correctness, applicability and utility of the model. A useful MSM has to be self consistent by conserving the defining Markov assumption (memoryless jump processes) for transition times longer than a minimum *lag time* and contain correct information about the essential physical processes, including stationary state occupation distributions as well as lifetimes of and transition rates between appropriate metastable states. Analyzing the transition network of the Markov state model should reveal important physical processes and properties of the proteins in question, including temporal stability of the associated geometrical atom configurations (which determine the functionality of the protein) and the composition, dynamics and importance of possible folding pathways. Those pathways describe forward and backward transitions from unfolded/*denatured* starting configurations to folded/*native* configurations via a sequence of comparatively long-lived, i.e. metastable, intermediate states.

Many reasonable methods have been developed in various detail to generate Markov state models for a large number of investigated systems[9–16]. A common, general approach consists of a sequence of steps,[14–16] including

1. some sort of dimensionality reduction
2. a geometric clustering method to partition the reduced conformational space into a discrete number (k) of so called *microstates*
3. a dynamic clustering method to further reduce the number of discrete states, yielding a transition network between $\nu < k$ metastable states.

Both, the initial configuration coordinates and the choice of complexity and parameters of the clustering methods should be adapted to the investigated system in question to gain reasonable results. Depending on the investigated system, different quantities of conformational dimensions, microstates and metastable states have proven to be sufficient to describe the essential dynamics in adequate detail.[14–16] Furthermore, it might even be feasible to skip the dynamic clustering step to build a valid and usable Markov state model.[16]

After summarizing the required theory of Markov state models and protein thermodynamics in Sec. II, a general workflow of generating a Markov state model from MD simulation data is presented in Sec. III. In the course, examples of geometric clustering methods (k-means[17] and Density Based Clustering[16]) and of a dynamic clustering method (Perron-Cluster-Analysis[9,10,13]) are introduced.

II. THEORY

A. Markov state models

A Markov state model constitutes a description of the dynamics of a stochastic system using the following assumptions:

The realizations of the stochastic system are restricted to be in one of a set of k discrete states $\{i\}$. The occupation probability to be in state i at time t is denoted $p_i(t)$ and the probability for a transition from state j to state i during *lag time* τ

is given by the transition matrix element $T_{ij}(\tau)$, as illustrated by a basic example in Fig. 1. The stochastic dynamics can be expressed by time homogeneous Markov processes, i.e. the occupation probability state vector $\boldsymbol{p}(t+\tau)$ at time $t+\tau$ is determined exclusively by the state vector $\boldsymbol{p}(t)$ at the previous time t, which is called *memoryless* behavior:

$$p_i(t+\tau) = \sum_j T_{ij}(\tau)\, p_j(t) \tag{1}$$

$$\boldsymbol{p}(t+\tau) = \boldsymbol{T}(\tau)\, \boldsymbol{p}(t) \qquad \text{(matrix form)}. \tag{2}$$

The time evolution during time $n\tau$ is then given by

$$\boldsymbol{p}(t+n\,\tau) = \boldsymbol{T}^n(\tau)\, \boldsymbol{p}(t)\,. \tag{3}$$

To conserve total probability ($\sum_i p_i(t) = 1$) the transition matrix $\boldsymbol{T}$ must be column stochastic:

$$\sum_i T_{ij} = 1\,. \tag{4}$$

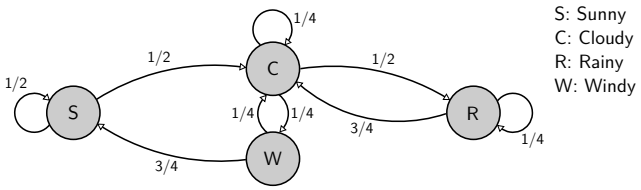


Figure 1. Basic example of a Markov state model: The weather is coarsely divided to be in one of four states: Sunny (S), cloudy (C), windy (W) or rainy (R). Given that the weather on some day is in a certain state j, the connection arrows express the probability that the weather of the following day will be in a certain state i.

Denoting the sorted eigenvalues by λ_i and sorted eigenvectors by $\boldsymbol{\alpha}_i$,

$$\boldsymbol{T}\boldsymbol{\alpha}_i = \lambda_i\, \boldsymbol{\alpha}_i \quad \text{with} \quad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k, \tag{5}$$

one can infer from column stochasticity[18] (Eq. (4)) that there exists at least one eigenvalue $\lambda_1 = 1$, such that $\boldsymbol{T}\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_1$, corresponding to a stationary state $\boldsymbol{p}_s = c_1\boldsymbol{\alpha}_1$. Moreover, $\lambda_1 = 1$ is the eigenvalue with the largest absolute value.[18] The theorem by Frobenius and Perron states that any initial distribution will converge to a stationary distribution for sufficiently long times:[18]

$$\lim_{t\to\infty} \boldsymbol{p}(t) = \lim_{n\to\infty} \boldsymbol{T}^n \boldsymbol{p}(0) = \boldsymbol{p}_s \quad \text{with} \quad \boldsymbol{T}\boldsymbol{p}_s = \boldsymbol{p}_s\,. \tag{6}$$

This behaviour will be further analyzed in terms of a decomposition into eigenvectors in Sec. II C.

In the following, the physical systems are treated in thermodynamic equilibrium, such that detailed balance holds,[13] $T_{ij}\, p_{s,j} = T_{ji}\, p_{s,i}$, and there is only one stationary state corresponding to the equilibrium distribution:[12] $\boldsymbol{p}_s = \boldsymbol{p}_{\text{eq}}$ and $\lambda_i < 1\,(i \neq 1)$.

B. Rate equation, lifetimes and transition times

A description equivalent to a Markov process is given by a rate equation,[18]

$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{p}(t) = \boldsymbol{K}\,\boldsymbol{p}(t)\,, \tag{7}$$

which defines the flow of probability per unit time from state j into state i by means of the elements k_{ij} of matrix $\boldsymbol{K}$, called *transition rates*. Comparing the formal integration of the rate equation

$$\boldsymbol{p}(t+\tau) = \mathrm{e}^{\tau\boldsymbol{K}}\boldsymbol{p}(t) \tag{8}$$

with the Markov time evolution (2) yields

$$\boldsymbol{T}(\tau) = \mathrm{e}^{\tau\boldsymbol{K}} \approx \mathbb{1} + \tau\boldsymbol{K} \tag{9}$$

with transition rates

$$k_{ij} \approx \frac{1}{\tau}(T_{ij} - \delta_{ij}) \tag{10}$$

and *transition times* $\tau_{ij} = k_{ij}^{-1}$.

Of special interest are the decay rates $k_i := -k_{ii} > 0$, since the corresponding *mean lifetimes* $\tau_i = k_i^{-1}$ express the average time, the ensemble realization will remain in state i before leaving to any other state. States with relatively large lifetimes are called *metastable states*, while states with short lifetimes are called *transition states*.

C. Markov process timescale separation

The eigenvectors $\boldsymbol{\alpha}_i$ of the $k \times k$ matrix $\boldsymbol{T}$ build a (not necessarily orthogonal) basis in k-dimensional vector space.[13] Thus, any state vector can be decomposed at any time as a linear combination of the eigenvectors with coefficients $c_i(t)$:

$$\boldsymbol{p}(t) = \sum_{i=1}^{k} c_i(t)\, \boldsymbol{\alpha}_i \tag{11}$$

Starting with an arbitrary state vector $\boldsymbol{p}(t=0) =: \boldsymbol{p}_0 = \sum c_i(0)\boldsymbol{\alpha}_i$ and $c_i := c_i(0)$, we can look at the time evolution

$$\boldsymbol{p}(n\tau) = \boldsymbol{T}^n\,\boldsymbol{p}_0 = \sum_{i=1}^{k} c_i\,\boldsymbol{T}^n\boldsymbol{\alpha}_i = \sum_{i=1}^{k} c_i\,\lambda_i^n\,\boldsymbol{\alpha}_i\,. \tag{12}$$

With $\lambda_1 = 1$ and *implied timescale*[13]

$$t_i := -\frac{\tau}{\ln\lambda_i} \Leftrightarrow \lambda_i = \exp\left[-\frac{\tau}{t_i}\right] \tag{13}$$

follows

$$\boldsymbol{p}(n\tau) = c_1\boldsymbol{\alpha}_1 + \sum_{i=2}^{k} c_i \exp\left[-\frac{n\tau}{t_i}\right]\boldsymbol{\alpha}_i\,. \tag{14}$$

Thus, the contributions of the eigenvectors to the state vector decay exponentially with implied timescales $t_i > 0$. In the limit of long times $n\tau$, only the contribution of the first eigenvector remains, which can be identified with the stationary state, $c_1\boldsymbol{\alpha}_1 = \boldsymbol{p}_s$, consistent with the Frobenius-Perron Theorem (Eq. (6)). This leads to a decomposition into $\nu - 1$ slower processes (t_i large, i.e. λ_i closer to one) and $k - \nu$ faster processes (t_i small, i.e. λ_i closer to zero):

$$\boldsymbol{p}(n\tau) = \boldsymbol{p}_s + \sum_{i=2}^{\nu} c_i \exp\left[-\frac{n\tau}{t_i}\right]\boldsymbol{\alpha}_i + \sum_{i=\nu+1}^{k} c_i \exp\left[-\frac{n\tau}{t_i}\right]\boldsymbol{\alpha}_i\,. \tag{15}$$

Taking only the first $\nu < k$ eigenvectors with the largest eigenvalues, enables a time scale separation of fast (*intrastate*) and slow (*interstate*) processes and thus a dynamic clustering of the k microstates into fewer (ν) metastable states, which constitutes the approach for the Perron-Cluster-Analysis presented in Sec. III C.

D. Chapman-Kolmogorov test for Markov state models

A Markov state model should conserve the Markov property for any timescale longer than a minimum lag time τ. This can be verified by calculating the transition matrix $\boldsymbol{T}(\tau)$ for lag time τ and for a longer lag time $n\tau$ directly from MD data and examining if the *Chapman-Kolmogorov equation*,[18]

$$\boldsymbol{p}(t+n\tau) = \boldsymbol{T}^n(\tau)\,\boldsymbol{p}(t) = \boldsymbol{T}(n\tau)\,\boldsymbol{p}(t) \Leftrightarrow \boldsymbol{T}(n\tau) = \boldsymbol{T}^n(\tau)\,, \tag{16}$$

is valid within statistical errors.

E. Internal and reaction coordinates

A molecular dynamics simulation of a biomolecule will generate a trajectory $\boldsymbol{x}(t)$ of the $3N$ Cartesian atom-coordinates. The molecular configurational state can equivalently be described by the internal coordinates *atom-atom-distances* r, *bond angles* θ, and *dihedral angles* ϕ, shown in Fig. 2.

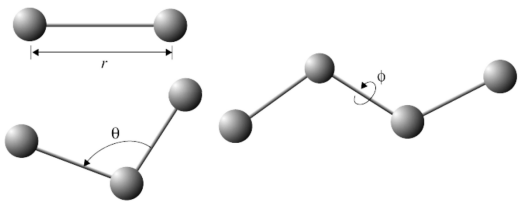


Figure 2. Internal coordinates $\{r, \theta, \phi\}$ used for force field calculations during molecular dynamics simulations of biomolecules. Adapted from Comp. and Biophys. group, Univ. of Illinois.[19]

Analysis of the time evolution of the internal coordinates has shown that the essential geometric structure of most proteins is given by the torsion angles (ϕ, ψ),[12,20–23] shown in Fig. 3.[23] An important underlying reason is the stiffness of atom-atom bonds and bond angles, leading to fast and small vibrations of r and θ around the equilibrium values.[21,23] In contrast, the torsion angles may show comparatively slow dynamics with a wide range of possible values.[21,23] Thus, it is a promising approach to reduce the dimensionality of the system by using only torsion angles for further analysis.[20–22].

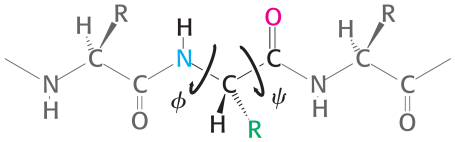


Figure 3. The torsion angles (ϕ, ψ) represent the essential geometric structure of most proteins. Taken from Berg, Tymoczko and Stryer.[23]

From the initially chosen coordinates, one can further deduce a reduced set of variables, called *reaction coordinates*, which capture most of the information about the dynamics of the system.[20–22] One method for this kind of dimensionality reduction is principle component analysis (PCA).

1. Principal component analysis (PCA)

Starting with a set of data $\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_M\}$ in coordinate space $\boldsymbol{q} = (q_1, q_2, \ldots, q_d)^T \in D_{\text{init}} \subseteq \mathbb{R}^d$, PCA is used to extract the linear combinations of coordinates, which show the largest variance within the given data. This is done by diagonalizing the symmetric covariance matrix $\sigma_{ij} = \langle (q_i - \langle q_i \rangle)(q_j - \langle q_j \rangle) \rangle$ and order its orthonormal set of uncorrelated eigenvectors $\{\boldsymbol{v}_i\}$ by the size of their eigenvalues s_i, $(s_1 \geq s_2, \ldots, \geq s_d)$. The first f *principal components* V_i denote the projections

$$V_i = \boldsymbol{v}_i \cdot \boldsymbol{q} \quad (i \leq f)\,. \tag{17}$$

For most protein systems, already a small number $f \leq 10$ of principal components is sufficient to capture most of the essential dynamics.[16,20,22]

F. Free energy and dimensionality reduction

Given a canonical (N, V, T) ensemble, the probability density for coordinates in full configuration space $\boldsymbol{x} \in \mathbb{R}^{3N}$ is given by the canonical distribution

$$P(\boldsymbol{x}) \propto \frac{1}{Z} \mathrm{e}^{-\beta U(\boldsymbol{x})} \quad \text{with} \quad \beta = \frac{1}{k_{\mathrm{B}} T}\,, \tag{18}$$

where $U(\boldsymbol{x})$ is the potential and Z is the partition function. The free energy $F := -k_{\mathrm{B}} T \ln Z = F(N, V, T)$ is a constant of the system, which is minimized by the canonical distribution, leading to a maximum of the total combined entropy of system and surrounding heat bath.

Projecting on a single reaction coordinate $V(\boldsymbol{x})$ leads to the marginal probability density of this coordinate

$$P(V) = \int \mathrm{d}^{3N}\boldsymbol{x}\; \delta(V(\boldsymbol{x}) - V)\, P(\boldsymbol{x})\,. \tag{19}$$

We can thereby define a free energy dependent on the coordinate V, i.e. given that $V(\boldsymbol{x})$ has a certain value V:

$$F(V) = -k_\mathrm{B}T \ln P(V) \; + \; \text{const.} \tag{20}$$

Relative to a reference value V_0, the probability density of V may then be expressed in terms of the free energy $F(V)$,

$$P(V) = P(V_0)\, \mathrm{e}^{-\beta\, \Delta F(V)} \quad \text{with} \quad \Delta F(V) := F(V) - F(V_0)\,, \tag{21}$$

as illustrated in Fig. 4. Choosing the reference V_0 such that $P(V_0)$ is maximal and setting $F(V_0) = 0$ then results in $\Delta F(V) = F(V)$ and $F(V) \geq 0$ for $(V \neq V_0)$.

This concept of the variable dependent free energy can be expanded to a set of f reaction coordinates, leading to a *free energy landscape* $F(V_1, V_2, \ldots, V_f)$, where local minima ("valleys") and local maxima ("barriers") correspond to probability maxima and minima, respectively.

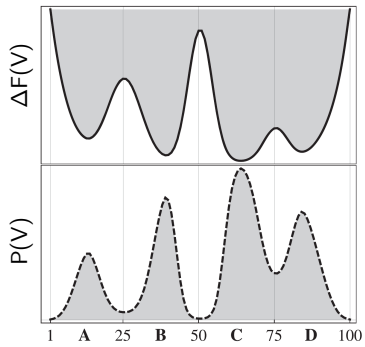


Figure 4. Free energy landscape $\Delta F(V)$ and probability density $P(V)$ dependent on a single reaction coordinate V. Adapted from Prinz *et al.*[13]

A larger number of reaction coordinates generally increases the detail of the energy landscape, as shown in Fig. 5.[14,24] It should however be noted, that the initial choice of coordinates, the method of determining reaction coordinates and the amount and quality of the sampled simulation data has essential influence on the significance of the resulting free energy function.[21,22]

The energy landscape does not only describe the stationary probability distribution, but contains information about the transition probability from one "valley" into another "valley", which decreases exponentially with barrier height. Thus, the single trajectories will remain for longer times in the local minima (metastable states), before conducting single, rare transitions into other local minima, as illustrated in Fig. 6. We thus find a timescale separation between fast intrastate motion and slow interstate motion.

III. GENERATION METHODS FOR MARKOV STATE MODELS

Starting with a sampled N-atom MD trajectory (time sequence) $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_M]$ in $3N$-dimensional Cartesian coordinate space, $\boldsymbol{x}_t \in \mathbb{R}^{3N}$, a Markov state model with ν discrete states and $\nu \times \nu$ transition matrix $\boldsymbol{T}(\tau)$ may be generated by a quite general workflow, consisting of (A) dimensionality reduction, (B) geometric clustering and (C) dynamic clustering, which is presented in the following.

A. Dimensionality reduction

The $3N$-dimensional Cartesian coordinate space should be reduced to f appropriate reaction coordinates $\{V_i\}$ to reduce complexity and enable a statistically significant sampling density for thermodynamic quantities like the free energy $F(V_1, \ldots, V_f)$ defined in Eq. (20). This may be achieved by firstly, choosing appropriate internal coordinates, like dihedral torsion angles, and secondly, applying principal component analysis, as described in Sec. II E.

The choice of initial coordinates is crucial as it may separate the internal motion from the overall motion, thereby uncovering truly stochastic dynamics and timescale separations of processes. While the overall translation can easily be removed, the elimination of the overall rotation is nontrivial for non-rigid (i.e. elastic-bond containing) bodies like biomolecules. Using sin/cosine transformed backbone dihedral angles as a basis for PCA, as suggested by Mu, Nguyen and Stock[21] has proven to be a promising approach to determine reaction coordinates of protein systems.[14,15,20–22]

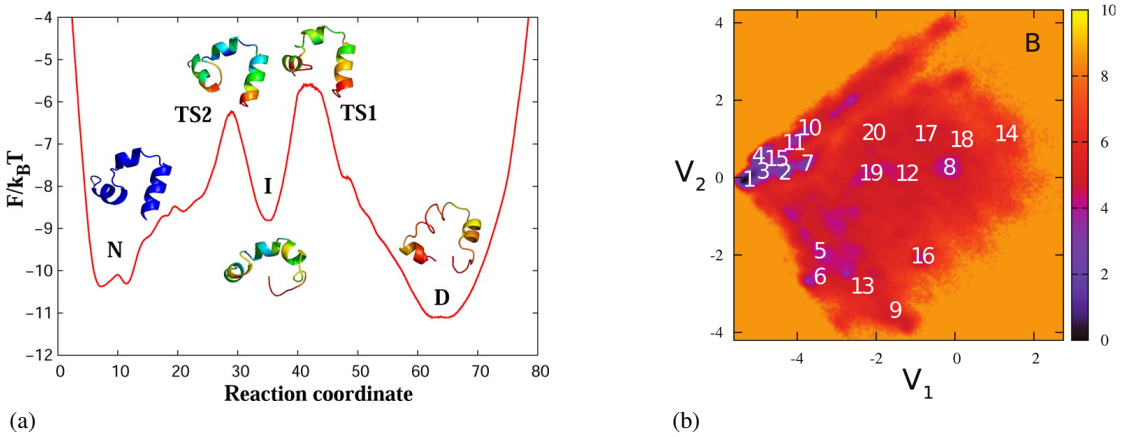
B. Geometric clustering

After dimensionality reduction, we are left with a MD trajectory $[\boldsymbol{V}_1, \ldots, \boldsymbol{V}_t, \ldots, \boldsymbol{V}_M]$ with $\boldsymbol{V}_t \in D_\mathrm{red} \subseteq \mathbb{R}^f$. To build a Markov state model, the continuous configuration space has to be partitioned into a set of k discrete, so called *microstates* $\{i\}$ leading to a discretized trajectory $[i_1, \ldots, i_t, \ldots, i_M]$ with $i_t \in \{1, 2, \ldots, k\}$, as illustrated in Fig. 7.

To define a set of microstates, which can be used as basis for a meaningful and correct Markov state model, many possible *geometric clustering* methods have been proposed[13], of which *k-means*[17] and *Density Based Clustering*[16] are introduced in the following.

The k-means method starts with a predefined number of microstates k and a set of k initial cluster centers $\hat{\boldsymbol{V}}_1, \ldots, \hat{\boldsymbol{V}}_k$. There are different methods to define the initial cluster centers, including Forgy partitioning by randomly picking k points from the data set and more sophisticated methods, like k-means++.[26] Based on the respectively most current cluster centers, the space is partitioned by Voronoi tesselation. The cluster centers are then iteratively moved in coordinate space by an algorithm, which minimizes the cumulative within-cluster variance of the data set.[17]

k-means is a robust and fast geometric clustering method, which is able to produce reasonable results for protein systems.[14,15,20] However, there are a number of disadvantages to be considered. For example, the number of states k is fixed beforehand, such that too many or too few states might be chosen and metastable geometrical states are cut into several



(a) (b)

Figure 5. Free energy landscapes deduced from different MD simulations of the same protein (Villin HP-35 headpiece) for different choices of one (a) and two (b) reaction coordinates V_i. (a) additionally shows the associated geometrical structures of the native (N), denatured (D), intermediate (I) and transition (TS) configuration states. (a) taken from Banushkina and Krivov.[24] (b) taken from Jain and Stock.[14]

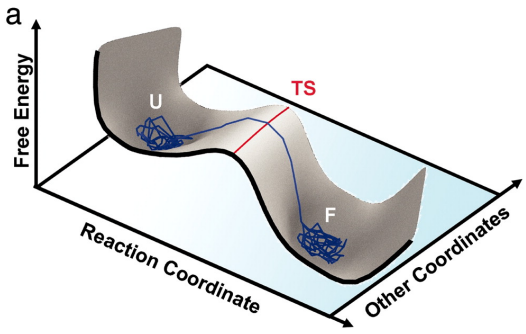


Figure 6. Trajectories show a timescale separation between fast intrastate motion and slow interstate motion, depending on the energy barrier height, which separates the metastable energy minima (in this case denoted by U and F, which corresponds to unfolded and folded conformational states of a protein). Adapted from Cho, Levy and Wolynes.[25]

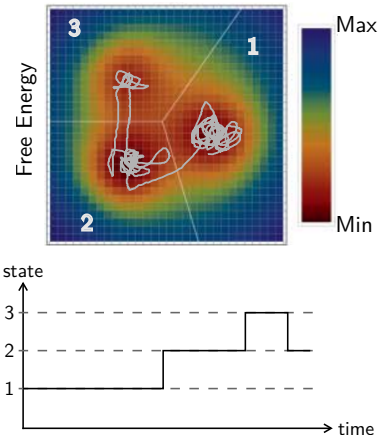


Figure 7. Basic example of a partitioning of the reduced two-dimensional reaction coordinate space into three discrete states leading to a discretized trajectory $i(t)$. The reaction coordinate dependent free energy landscape is color coded and the initial continuous trajectory is shown as gray line. Adapted from Prinz *et al.*[13]

states or combined into one state. Additionally, the initialization of the cluster centers includes a stochastic step, which might cause the algorithm to find only a local optimum of the variance minimization. The algorithm has thus to be repeated several times with different initial conditions and, depending on the case, also with different numbers of clusters k.

Complex protein systems might therefore require more sophisticated geometric clustering methods, like Density Based Clustering, which begins with estimating the free energy landscape from simulation data and aims to partition the coordinate space at free energy barriers (i.e. paths of local maxima of the free energy). The reason for this approach is, that transitions of trajectories across free energy barriers are much rarer than motion along the negative free energy gradient. Thus, all of the coordinate space of a local energy "valley" should rather be clustered into the same metastable state. The resulting challenge of density based clustering is to cut the space not at the geometric middle between two energy minima (which are usually well sampled), but at the local energy maxima (which are usually very insufficiently sampled), as illustrated in Fig. 8.

A robust approach proposed by Sittel and Stock[16], starts with estimating the free energy density by counting the number of local neighbors found in a hypersphere of radius R around every data point $\boldsymbol{V}_i$ of the trajectory. From this neighborhood population $P_R(\boldsymbol{V}_i)$, the local free energy is calculated as

$$F_R(\boldsymbol{V}_i) = -k_B T \ln[P_R(\boldsymbol{V}_i)/P_R^{\max}], \tag{22}$$

where $P_R^{\max} = P_R(\boldsymbol{V}_{\max}) = \max\{P_R(\boldsymbol{V}_i)\}$ and $F_R(\boldsymbol{V}_{\max}) = 0$. The data points are then lumped together into clusters at different threshold levels of the free energy, as shown in Fig. 9: Starting with a first energy threshold $F_1 > 0$, only data points

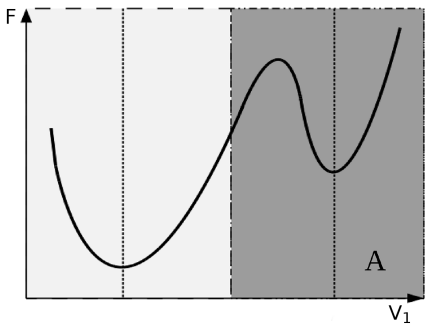


Figure 8. Since transitions across free energy barriers are much rarer than motions in direction of decreasing free energy, clusters should be partitioned at energy barriers to capture the dynamics between metastable states. Cutting the coordinate space in the geometric middle (vertical dashed-dotted line) of the two local energy minima assigns some region "left" of the energy barrier to the "right" cluster (dark gray), which leads to erroneous dynamical behavior. Adapted from Jain and Stock.[14]

with $F(\boldsymbol{V}_i) \leq F_1$ are included in the first level of the lumping process. All other data points are ignored at this level. The selected points are assigned to the same cluster, if they are "close together" in coordinate space, meaning that a point can only belong to one cluster and all points of a certain cluster are separated from all points of a different cluster by at least the lumping distance d_{lump}.

Usually, there will only be one cluster after the first lumping level. At the second level, a higher threshold $F_2 > F_1$ is defined and all data points with $F \leq F_2$ are included in the lumping process. By filling up the energy landscape from below, new clusters will be formed and existing clusters will grow at each level until two clusters are merged going from a certain level to the next level (cf. Fig. 9c $\rightarrow$ Fig. 9d). We can conclude that the local energy barrier must run at this merging point (resp. merging line/hyperarea in higher dimensions) between the two clusters. Proceeding like this until all data points are included in the analysis, a tree structure of growing and merging clusters is generated, as shown in Fig. 9, from which the whole trajectory-visited space can be partitioned into k discrete microstates.

After coordinate space partitioning, the $k \times k$ transition matrix $\tilde{\boldsymbol{T}}(\tau)$ can now be calculated from the MD-trajectory, following Prinz *et al.*[13] Firstly, the continuous space trajectory $[\boldsymbol{V}(t=0), \boldsymbol{V}(t=\tau), \ldots, \boldsymbol{V}(t=M\tau)]$ is projected onto a discrete temporal sequence $[i(0), i(\tau), \ldots, i(M\tau)]$ of the corresponding microstates $i(t) \in \{1, 2, \ldots, k\}$. Secondly, the number $\#\,(i \rightarrow j)$ of direct transitions from state i to state j, i.e. the number of occurrences of the case $[i(t+\tau) = j \,|\, i(t) = i]$, is counted for all $i(t)$ of the trajectory and collected in the *count matrix*

$$\boldsymbol{N}(\tau) = \begin{bmatrix} \#\,(1 \rightarrow 1) & \#\,(2 \rightarrow 1) & \ldots & \#\,(k \rightarrow 1) \\ \vdots & \vdots & \ddots & \vdots \\ \#\,(1 \rightarrow k) & \#\,(2 \rightarrow k) & \ldots & \#\,(k \rightarrow k) \end{bmatrix} . \qquad (23)$$

By normalizing the columns of the count matrix,

$$\tilde{T}_{ij} = \frac{N_{ij}}{\sum_m N_{mj}} , \qquad (24)$$

we finally receive the transition matrix

$$\tilde{\boldsymbol{T}}(\tau) = \begin{bmatrix} P(1 \rightarrow 1) & P(2 \rightarrow 1) & \ldots & P(k \rightarrow 1) \\ \vdots & \vdots & \ddots & \vdots \\ P(1 \rightarrow k) & P(2 \rightarrow k) & \ldots & P(k \rightarrow k) \end{bmatrix} . \qquad (25)$$

It should be noted that the count matrices of different simulation trajectories of the same system can be combined into a single count matrix to gain improved statistical significance.[13] Thus, for Markov state modelling, many short simulation runs can be performed in parallel, instead of one impractical, single long-time run.

For simple protein systems, geometric density based clustering might be sufficient to build the transition matrix for a reasonable Markov state model.[16] In more complex cases however, additional dynamic clustering is necessary to appropriately reduce the number of microstates and capture the essential dynamic between metastable states.[16]

C. Dynamic clustering

Starting from any chosen discretization of the configuration space into k microstates and the corresponding $k \times k$ transition matrix $\tilde{\boldsymbol{T}}$ calculated from the MD trajectory (as described in Sec. III B), *dynamic clustering* can be used to identify the essential dynamic processes of the system and accordingly cluster the microstates into fewer ($\nu < k$) metastable states. Generally, one analyzes the dynamic transition behavior between the microstates to conclude, which of them belong to the same metastable state. A variety of dynamic clustering methods have been developed over the last years,[9–11,13,14] of which the robust *Perron Cluster Cluster Analysis* (PCCA) is introduced in the following as proposed by Deuflhard *et al.*[9,10]

Initially, the eigenvalues λ_i and corresponding eigenvectors $\boldsymbol{\alpha}_i$ of the $k \times k$ transition matrix $\tilde{\boldsymbol{T}}$ are calculated and sorted by their eigenvalues. Figure 10 shows an example of a 100×100 transition matrix with the component structure of its first four eigenvectors and the first twelve eigenvalues λ_i. The implied timescales show a clear separation between the first four and the following eigenvalues. Thus, using Eq. (15),

$$\boldsymbol{p}(n\tau) = \boldsymbol{p}_s + \sum_{i=2}^{\nu} c_i \exp\left[-\frac{n\tau}{t_i}\right] \boldsymbol{\alpha}_i + \sum_{i=\nu+1}^{k} c_i \exp\left[-\frac{n\tau}{t_i}\right] \boldsymbol{\alpha}_i ,$$

with $\nu = 4$, we identify the stationary distribution and three slow processes and take only the first four eigenvectors for further analysis.

In equilibrium, all eigenvectors $\boldsymbol{\alpha}_i$ except for $\boldsymbol{\alpha}_1$ will have components $(\boldsymbol{\alpha}_i)_j$ with positive entries and components $(\boldsymbol{\alpha}_i)_l$ with negative entries,[13] as shown in Fig. 10(b). Let the contribution to the linear combination of $\boldsymbol{p}$ at $t = 0$ be $c_i \boldsymbol{\alpha}_i$ and $c_i \geq 0$ without loss of generality. At the next time step,

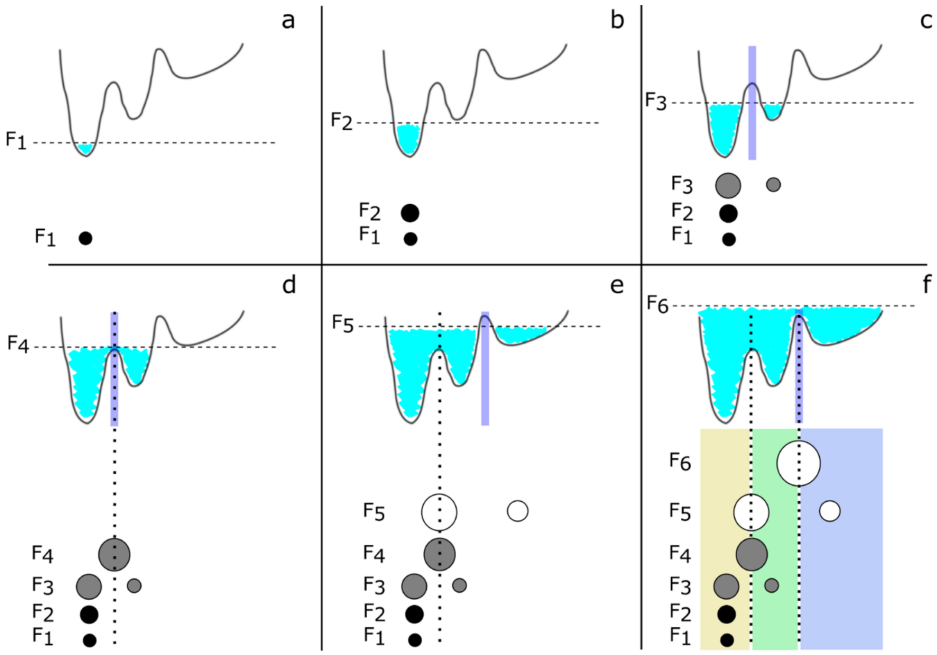
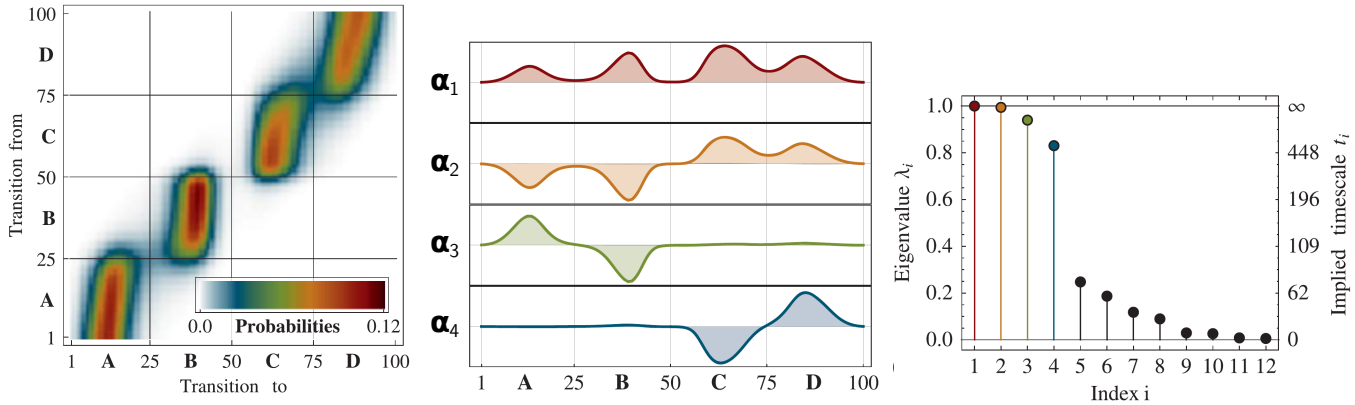


Figure 9. Density based clustering method as proposed by Sittel and Stock.[16] The estimated free energy surface is filled stepwise from the minimum with data points of the MD trajectory, which are then lumped into the same cluster if they are geometrically "close together", such that the formed clusters are separated by at least the minimum "lumping distance". At different levels of the lumping process, only data points with free energy below the level threshold $F \leq F_i$ are included into the analysis. With increasing energy threshold, new clusters are formed (a, c, e), while existing clusters grow (a) $\rightarrow$ (b) $\rightarrow$ (c) until they merge at some energy level (d, f). By this procedure, a tree structure of emerging, growing and joining clusters is generated (f) and the local energy barriers can be identified at the merging points and used to appropriately cut the configuration space into microstates.



(a) (b) (c)

Figure 10. Example for a 100×100 transition matrix. (a) The color coded elements (transition probabilities) already indicate metastable regions of microstates by the approximately block diagonal structure. (b) The component structure of the first four eigenvectors $\boldsymbol{\alpha}_i$ shows the stationary distribution $\propto \boldsymbol{\alpha}_1$ and regions with negative sign and positive sign for $i \neq 1$. The component indices are cut into regions A, B, C, and D at positions of sign switches by the PCCA algorithm. (c) The eigenvalues λ_i show a clear timescale separation between three slow processes ($i = 2,3,4$) and fast processes ($i > 4$). Figures adapted from Prinz *et al.*[13]

the contribution shrinks by a factor of λ_i (with $|\lambda_i| < 1$): $c_i \boldsymbol{\alpha}_i \rightarrow c_i \lambda_i \boldsymbol{\alpha}_i$ (cf. Eq. (12)). Thereby, the states with *positive* components of $\boldsymbol{\alpha}_i$ will *lose* probability (contribution getting less positive) and the states with *negative* components of $\boldsymbol{\alpha}_i$ will *gain* probability (contribution getting less negative). Thus, looking at the time evolution of the contribution of a certain single eigenvector $\boldsymbol{\alpha}_i$, we can infer a flow of probability from some subsets of states into other subsets, depending on the sign structure of the eigenvector components. Looking at $\boldsymbol{\alpha}_2$ in Fig. 10(b), probability flows from regions C and D into regions A and B under the slowest process with the largest implied timescale t_2. Dividing the microstates at in-

dex 50, where the components of $\boldsymbol{\alpha}_2$ switch their sign, will lead to a first clustering into two states. Analogously the sign structure of $\boldsymbol{\alpha}_3$ describes the second slowest process, where probability flows from region A to region B and implies an additional cut at the sign switch at index 25. Lastly, we identify the flow from probability from region D to region C by the structure of $\boldsymbol{\alpha}_4$ and cut at the sign switch at index 75 to obtain a clustering of the microstates in regions A, B, C and D into four metastable states, which were already indicated by the structure of the transition matrix in Fig. 10(a).

By investigating more eigenvectors, a number of processes on shorter timescales will explicitly be included. The number of considered eigenvectors has to be chosen depending on the desired degree of detail and complexity, which is necessary for the system in question.

Although the PCCA approach is justified theoretically, it should be noted that it turned out to be numerically unstable in many cases, because the sign switches of the eigenvector components might not be unambiguously identifiable due to insufficient sampling[10,12,13]. Although the basic idea is still valid, Deuflhard *et al.*[10] proposed an improved method called PCCA+, which is much more robust in the practical application.

After projecting the discrete trajectory sequence $[i(0), i(\tau), \ldots, i(M\tau)]$ of the microstate basis $\{i\} = \{1, 2, \ldots, k\}$ onto the sequence $[j(0), j(\tau), \ldots, j(M\tau)]$ in the new basis $\{j\} = \{1, 2, \ldots, \nu\}$ of metastable states, the final $\nu \times \nu$ transition matrix $\boldsymbol{T}(\tau)$ and the corresponding transition rates and times are calculated as described in Sec. III B and Eq. (10), respectively. The resulting Markov state model can be presented as a transition rate network, as shown in Fig. 11.

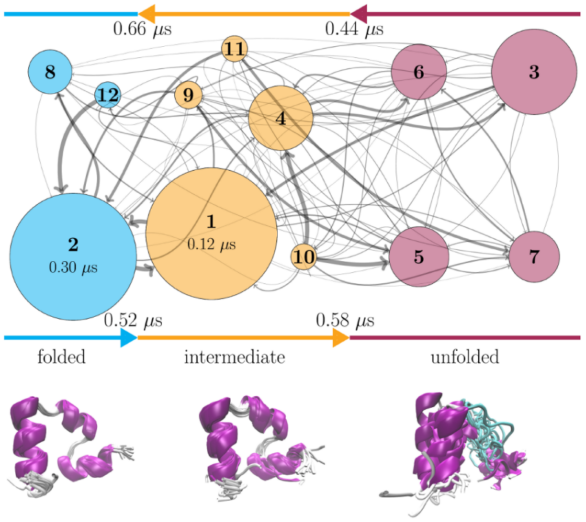


Figure 11. A Markov state model of the Villin HP-35 headpiece is presented as a transition network between 12 identified metastable states illustrated as colored nodes with area sizes corresponding to the stationary occupation ratio of the states. The mean lifetimes of the states with the highest populations are given in nodes 1 and 2. Thick connection arrows represent large transition rates (i.e. small transition times) between states. Blue, yellow, and purple nodes respectively indicate folded, intermediate, and unfolded protein conformations. An overlay of the geometrical structures of the states with the respectively highest populations are shown below the corresponding network regions. Figure taken from Sittel and Stock.[16]

IV. SUMMARY AND CONCLUSION

Markov state modelling leads to a reduced description of the dynamics of a stochastic system by characterizing memoryless jump processes between a number of discrete states. For protein systems, the model can be represented as a transition network between metastable conformational states. This description of the dynamics can be useful to interpret massive amounts of data generated by molecular dynamics simulations, if the modelling is conducted in an appropriate way to identify the significant, essential dynamic processes and the corresponding metastable states.

From a created Markov state model, one can calculate important properties, like free energies, the stationary occupation distribution and lifetimes of the states, as well as transition rates between the states to gain understanding of the thermodynamics and kinetics of the complex protein system. The geometrical structures associated with the states, like folded and unfolded configurations, and their time dependent transformations are of great importance, because they determine the functionality of the protein.

I presented a quite general and modular workflow to generate a Markov state model from molecular dynamics simulation data by a sequence of steps, (i) dimensionality reduction (Sec. III A), (ii) geometric clustering (Sec. III B), (iii) dynamic clustering (Sec. III C), and introduced example methods for each of the steps, namely (i) Principal Component Analysis, (ii) k-means and Density Based Clustering, and (iii) Perron-Cluster-Analysis. It is possible to replace the presented example methods by other methods, which fulfill the same function, showing the generality of the workflow.

Conclusively, it shall be emphasized that the right choice of initial coordinates (e.g. backbone torsion angles) and the parameters used for the described methods in each of the steps, e.g. the number of reduced dimensions and metastable states, is on the one hand crucial to gain a meaningful Markov state model and on the other hand highly dependent on the system in question and the desired applicability of the model. Consequently, the execution of the methods has to be adjusted and the results have to be validated and interpreted for each investigated system and there is generally no standardizable, unambiguous process for optimal Markov state modelling.

REFERENCES

[1] W. van Gunsteren, D. Bakowies, R. Burgi, I. Chandrasekhar, M. Christen, X. Daura, P. Gee, A. Glattli, T. Hansson, C. Oostenbrink, C. Peter, J. Pitera, L. Schuler, T. Soares, and H. B. Yu, Chimia **55**, 856 (2001).
[2] L. D. Schuler, X. Daura, and W. F. Van Gunsteren, Journal of Computational Chemistry **22**, 1205 (2001).
[3] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, and C. Sim-

merling, "Comparison of multiple amber force fields and development of improved protein backbone parameters," (2006), arXiv:0605018 [q-bio].
[4]R. B. Best and G. Hummer, Journal of Physical Chemistry B **113**, 9004 (2009), arXiv:NIHMS150003.
[5]K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw, Proteins: Structure, Function and Bioinformatics **78**, 1950 (2010).
[6]S. Piana, K. Lindorff-Larsen, and D. E. Shaw, Proceedings of the National Academy of Sciences of the United States of America **109**, 17845 (2012).
[7]D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, Science **330**, 341 (2010).
[8]R. García-Fandiño, P. Bernadó, S. Ayuso-Tejedor, J. Sancho, and M. Orozco, PLoS Computational Biology **8** (2012), 10.1371/journal.pcbi.1002647.
[9]P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte, Linear Algebra and its Applications **315**, 39 (2000).
[10]P. Deuflhard and M. Weber, Linear Algebra and Its Applications **398**, 161 (2005).
[11]J. D. Chodera and F. Noé, Current Opinion in Structural Biology **25**, 135 (2014), arXiv:NIHMS150003.
[12]F. Noé, I. Horenko, C. Schütte, and J. C. Smith, Journal of Chemical Physics **126**, 1 (2007).
[13]J. H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schtte, and F. Noé, Journal of Chemical Physics **134** (2011), 10.1063/1.3565032.
[14]A. Jain and G. Stock, Journal of Chemical Theory and Computation **8**, 3810 (2012).
[15]A. Jain and G. Stock, Journal of Physical Chemistry B **118**, 7750 (2014).
[16]F. Sittel and G. Stock, Journal of Chemical Theory and Computation **12**, 2426 (2016).
[17]J. A. Hartigan and M. A. Wong, Applied Statistics **28**, 100 (1979).
[18]N. G. Van Kampen, *Stochastic processes in physics and chemistry*, 2nd ed. (Elsevier Science, 1992).
[19]UIUC, "Internal force field coordinates," Univ. of Illinois at Urbana-Champaign, Comp. and Biophys. Group, `http://www.ks.uiuc.edu/Training/Tutorials/namd/namd-tutorial-unix-html/node25.html`, accessed: 2017-12-09.
[20]F. Sittel, A. Jain, and G. Stock, Journal of Chemical Physics **141**, 014111 (2014).
[21]Y. Mu, P. H. Nguyen, and G. Stock, Proteins **58**, 45 (2005), arXiv:0605018 [q-bio].
[22]A. Altis, M. Otten, P. H. Nguyen, R. Hegger, and G. Stock, Journal of Chemical Physics **128** (2008), 10.1063/1.2945165.
[23]J. M. Berg, J. L. Tymoczko, and L. Stryer, *Biochemistry*, 7th ed. (W H Freeman, 2012).
[24]P. V. Banushkina and S. V. Krivov, Journal of Chemical Theory and Computation **9**, 5257 (2013).
[25]S. S. Cho, Y. Levy, and P. G. Wolynes, Proceedings of the National Academy of Sciences **103**, 586 (2006).
[26]D. Arthur and S. Vassilvitskii, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (2007) pp. 1027–1025, arXiv:1212.1121.