GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Fakultät für Physik

Master's Thesis

Analyse von Protein Dynamik mit Markov Modellen

Analysis of protein dynamics with Markov models

prepared by

Max Linke

from Gera

at the Max Planck Institute for Biophysical Chemistry - Computational Biophysics

Thesis period:	7th February 2014 until 4th August 2014
First referee:	Hon.-Prof. Dr. Helmut Grubmüller
Second referee:	Prof. Dr. Marcus Müller

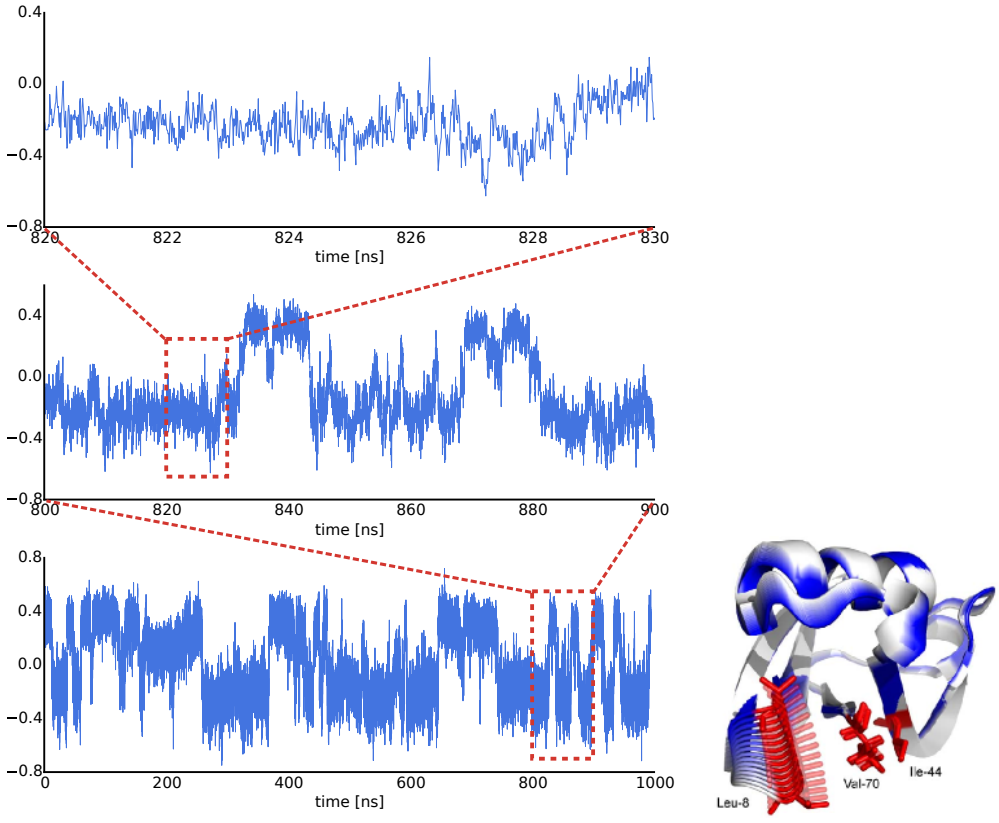
Abstract

Markov State Models (MSM) are a method to find slow dynamics in proteins by approximating the slow dynamics with a Markov chain on a discrete partition of a sub-space of the configuration space. MSMs can extract this slow dynamics information from an ensemble of short simulations. To date MSMs require prior knowledge about the specific protein examined to select a sub-space and a total simulation time in the millisecond range. There have been first steps to use time-lagged independent component analysis (TICA) [1] to automatically find the slow sub-space in a protein. TICA has been recently used [2] with a 30 residue intrinsically disordered peptide kinase inducible domain. We found that TICA is not guaranteed to always find the slow sub-space in a MD-simulation. We could also show that TICA can be used to extract slow dynamics information with MSMs from 100 Ubiquitin simulations with a total simulation time of just 38 μs.

Contents

1. Introduction

Proteins are molecular machines that perform different functions in a cell, for example signal transduction, regulation, transcription and others. How proteins function is an interesting topic. Since protein function is related to their motions and dynamics, we have to find a way to describe the dynamics first to learn about the how they function.



(a) (b)

Figure 1.1.: a) The pincer mode [3] motion from a 1 μs MD-simulation of Ubiquitin at different timescales. From top to bottom the timescale is increased by one order of magnitude in each picture. b) Visualization of the pincer mode in Ubiquitin, Source: [4]

As an example for protein motion consider the pincer mode of Ubiquitin [3], see fig. 1.1a. The pincer mode describes a collective motion of the whole protein that is mainly driven by the movement of the side chain around residue 8, see fig. 1.1b. On short timescales of 10 ns the dynamics are governed by stochastic fluctuations. On a larger timescale of 100 ns the fluctuations stay confined to specific regions for finite times and then quickly transitions into another region, such a transition occurs at 830 ns. On the timescale of 1 μs the dynamics are governed by the jumps between different phase space regions.

We will call such a confined region state, in the literature theses states are also called conformations. That proteins can have different states is known from photo-relaxation experiments on myoglobin [5]. Proteins can switch between states through external influences or by thermal fluctuations.

This means that slow dynamics in a protein can be described by a jump process between different states. We will model the jump process with conditional probabilities for a transition between states to happen in a given time τ_{MSM}. As it can be seen in Fig. 1.1a the dynamics in a state are very quick and it is reasonable to assume that the system will have no memory about where it came from after a jump. This is called the Markov property. Models that use the Markov property and conditional probabilities are called Markov models. To build a Markov Model from MD-simulations the conformation space is discretized into a micro state clustering and the transition probabilities are estimated by counting observed transition between states in the simulation [6], [7]. These models of protein dynamics are called Markov State Models (MSM).

That the systems fulfills the Markov property is just an assumption and not necessarily true for a arbitrary discretization. This leads to a systematic error in the estimated MSM. The movements describing the slow dynamics experience the steepest changes in the transition regions. It has been shown that the systematic error can be made arbitrarily small by improving the estimates for the steep changes, which means placing more states in transition regions between metastable conformations [8]. Because it is generally not known where the transition regions are a fine micro state clustering of the phase space is chosen. Today a Voronoi tessellation of cluster centers calculated with clustering algorithms like k-means are common [9].

The conformation space of a protein in all atom detail has $3N$ dimensions, N being the number of atoms. In high dimensions the data points become sparse and clustering methods, that rely on distance measures, have problems because the

distance between the farthest and nearest points to a reference tends towards 0 [10]. The problem of sparse data can theoretically be overcome with more sampling, e.g. more or longer simulations, but this is impractical in most cases. Because atom bonds are very rigid a lot of these degrees of freedom will be correlated and contain redundant information. We will use sub-spaces that have a minimal correlation between different degrees of freedom. For MSMs dihedral angles [11] or RMSDs [12] have been used. Finding a suitable sub-space usually requires experience and insight into the specific protein that is studied. There exist methods that can automatically find sub-spaces based on different criteria.

A common method for dimension reduction in MD-simulations that does not require any prior information is Principal Component Analysis (PCA) [13]. PCA aims to find a subset of n dimensions that explains most of the variation in a system. PCA is a linear transformation to convert the spacial coordinates into a set uncorrelated variables called principle components. By definition the first principle component has the largest variance, the second the second largest variance and so on. When building a MSM using the first n components with the largest variance we automatically assume that large amplitude motions are most important in a MSM. This assumption does not need to be true for the slow processes we are interested in.

Another method for dimension reduction is Time-lagged Independent Component Analysis (TICA) [14]. TICA aims to finds a sub-space with the slowest motions. This linear transformation converts the original coordinates into a set of uncorrelated variables but instead of maximizing for the largest variance it maximizes the values of the autocorrelation at a predefined lag-time τ_{TICA}. In a Markovian system the autocorrelation function shows an exponential decay and the slowest process would have the highest autocorrelation value for any lag-time τ_{TICA}.

1.1. Project Outline

Markov State Models have been used on different proteins like the G-protein-coupled receptor β_2AR [15] or GB1 hairpin [16] but they require a large amount of simulation time, 2.15 ms for the G-protein-coupled receptor and 0.7 ms the for the GB1 hairpin.

We want to find out if Markov State Models can be used to find slow dynamics in Ubiquitin with a much smaller amount of simulation time, 38 μs, and without prior knowledge with the help of PCA and TICA. For TICA we also checked if it is

always guaranteed to find the slow sub-space in a MD-simulation.

In Chapter 2 we review why it is sufficient to consider the conformation space for MSMs and the theoretical background for MSMs. We will show how MSMs are used to find the slow dynamics in a protein. Here we also show how TICA and PCA are used to construct low dimensional sub-spaces.

Chapter 3 introduces the methods used to estimate a MSM from a set of MD-simulations and the clustering algorithms k-center and k-means that are used to discretize the sub-space. We will also discuss how the free parameter τ_{TICA} for TICA can be chosen.

In Chapter 4 we will present the the results that show that TICA is not guaranteed to find the slow sub-space in a MD-simulation. We will also present the analysis of 100 Ubiquitin MD-simulations with MSMs. Each simulation is 380 ns for a total simulation time of just 38 μs.

We will discuss the relevance of the results from a broader perspective in chapter 5 and give an outlook on future work in chapter 6.

2. Theory

2.1. Markov Property Of Proteins

A system can be described with a Markov model if it fulfills the Markov property. This means the systems has no memory about it's past and it's future is solely determined by the current state. So the probability to go from a point $\boldsymbol{x}$ to a point $\boldsymbol{y}$ in the time τ is given by:

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{y}; \tau) d\boldsymbol{y} &= \mathbb{P}[\boldsymbol{x}(t+\tau) \in \boldsymbol{y} + d\boldsymbol{y} | \boldsymbol{x}(t) = \boldsymbol{x}, \boldsymbol{x}(t-\tau)] \\ &= \mathbb{P}[\boldsymbol{x}(t+\tau) \in \boldsymbol{y} + d\boldsymbol{y} | \boldsymbol{x}(t) = \boldsymbol{x}] \\ &\boldsymbol{x}, \boldsymbol{y} \in \Omega, \tau \in \mathbb{R}_{0+} \end{aligned} \tag{2.1}$$

Where Ω is the complete phase space. A protein fulfills this requirement if the whole phase space including atomic positions and momenta is analyzed. Then the Hamiltonian equations of motion will give the time evolution from any point in phase space. But it is not practical to use this high dimensional phase space. The first dimension reduction is to only use the atomic positions. Leaving out the momenta in the analysis can introduce memory at small timescales due to inertia. To check at what timescales the Markov property is still fulfilled for the coordinates alone it is helpful to model the interactions between the protein and a canonical heat bath by nonlinear Langevin dynamics as shown by Zwanzig [17].

$$m\ddot{x} = -\Delta U(x) - \gamma \dot{x} + F_r(t) \tag{2.2}$$

Where γ is a friction constant, m the atomic masses of the atoms, $U(x)$ the potential energy and F_r a Gaussian distributed random force. When taking the Fourier transform of eq. 2.2 it is possible to compare the left and right hand site

and estimate a timescale when inertia becomes negligible.

$$m\omega^2\tilde{x} = -k\tilde{U}(x) + \gamma\omega\tilde{x} + \tilde{F}_r$$
$$\Rightarrow (m\omega - \gamma)\omega\tilde{x} = -k\tilde{U}(x) + \tilde{F}_r$$

Friction will be the dominant factor if $m\omega \ll \gamma$. To get an estimate of the timescale at which $m\omega = \gamma$ we use the Einstein relation $\gamma = \frac{2}{D}k_bT$ [18] and order of magnitude values for k_bT at room temperature, the diffusion constant of a water molecule in water [19] and the atomic mass of a water molecule.

$$D \approx 10^{-9}\,\mathrm{m^2/s}, k_bT \approx 10^{-21}\,\mathrm{kgm^2/s^2}, m \approx 10^{-26}\,\mathrm{kg}$$
$$\rightarrow \frac{1}{\omega} \approx 10^{-14}s = 10\,\mathrm{fs}$$

This is a very rough estimate. The largest error is likely in the diffusion constant because the diffusion of a single atom in a protein will certainly not be the same as for a water molecule in water. We are in the high friction regime because we are interested in the ns timescale. This means we can restrict our analysis to the conformation space without introducing memory due to inertia.

2.2. Markov State Models

In this thesis we want to use MSM to extract information about the slow dynamics of a protein from a set of MD-simulations. In this section we will show how this information can be obtained from MSM, how MSMs are constructed using one or more MD-simulations and how τ_{MSM} should be chosen. To make the flowing mathematical derivations easier we will make two assumptions.

- Detailed Balance

  In equilibrium the fraction of the system going from point $\boldsymbol{x}$ to a point $\boldsymbol{y}$ in a time τ has to be the same as the fraction going from $\boldsymbol{y}$ to $\boldsymbol{x}$. This is known as detailed balance.

$$\mu(\boldsymbol{x})p(\boldsymbol{x},\boldsymbol{y};\tau) = \mu(\boldsymbol{y})p(\boldsymbol{y},\boldsymbol{x};\tau) \tag{2.3}$$

$\mu(\boldsymbol{x})$ is the stationary distribution and $p(\boldsymbol{x},\boldsymbol{y};\tau)$ is the probability to move from point $\boldsymbol{x}$ to point $\boldsymbol{y}$ in the time τ.

- Ergodicity

  For $t \to \infty$ a trajectory has to come arbitrary close to any point in phase space, which is equivalent to say that the time average and the average over the complete phase space Ω become equal.

$$\int_\Omega d\boldsymbol{x}\mu(\boldsymbol{x})A(\boldsymbol{x}) = \lim_{t\to\infty}\frac{1}{t}\int_0^t dt' A(\boldsymbol{x}(t')) \tag{2.4}$$

extracting slow dynamics information from a MSM

Markov State Models are describing the time evolution of ensemble probabilities. Considering an ensemble with a probability distribution $p_t(\boldsymbol{x}) \neq \mu(\boldsymbol{x})$. A straight forward description for the time evolution is to use a continuous operator Q and propagate p directly [8].

$$p_{t+\tau_{MSM}}(\boldsymbol{x}) = Q(\tau_{MSM}) \circ p_t(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega} d\boldsymbol{y}p(\boldsymbol{y},\boldsymbol{x};\tau_{MSM})p_t(\boldsymbol{y}) \tag{2.5}$$

It is better to use the transfer operator T [8] instead, because it can be directly estimated from simulations, shown in sec. 2.2, and we can show that the eigenfunctions of T are orthogonal using detailed balance. The transfer operator propagates functions u which are densities normalized with the stationary distribution $u_t = \frac{p_t}{\mu}$.

$$u_{t+\tau_{MSM}}(\boldsymbol{x}) = T(\tau_{MSM}) \circ u_t(\boldsymbol{x}) = \frac{1}{\mu(\boldsymbol{x})}\int_{\boldsymbol{y}\in\Omega} d\boldsymbol{y}p(\boldsymbol{y},\boldsymbol{x};\tau_{MSM})\mu(\boldsymbol{y})u_t(\boldsymbol{y}) \tag{2.6}$$

T is self-adjoint in a Hilbert space with weighted the scalar product $\langle v|w\rangle_\mu = \int d\boldsymbol{x}v(\boldsymbol{x})^* w(\boldsymbol{x})\mu(\boldsymbol{x})$.

$$
\begin{aligned}
\langle Tv|w\rangle_\mu &= \int d\boldsymbol{y}(Tv)^*(\boldsymbol{y})w(\boldsymbol{y})\mu(\boldsymbol{y}) \\
&= \int d\boldsymbol{y}w(\boldsymbol{y})\mu(\boldsymbol{y})\frac{1}{\mu(\boldsymbol{y})}\int d\boldsymbol{x}p(\boldsymbol{x},\boldsymbol{y};\tau_{MSM})\mu(\boldsymbol{x})v(\boldsymbol{x}) \\
&\overset{d.b.}{=} \int d\boldsymbol{y}w(\boldsymbol{y})\mu(\boldsymbol{y})\frac{1}{\mu(\boldsymbol{y})}\int d\boldsymbol{x}p(\boldsymbol{y},\boldsymbol{x};\tau_{MSM})\mu(\boldsymbol{y})v(\boldsymbol{x})\frac{\mu(\boldsymbol{x})}{\mu(\boldsymbol{x})} \\
&= \int d\boldsymbol{x}v(\boldsymbol{x})\mu(\boldsymbol{x})\frac{1}{\mu(\boldsymbol{x})}\int d\boldsymbol{y}p(\boldsymbol{y},\boldsymbol{x};\tau_{MSM})\mu(\boldsymbol{y})w(\boldsymbol{y}) \\
&= \int d\boldsymbol{x}v(\boldsymbol{x})\mu(\boldsymbol{x})(Tw) = \langle v|Tw\rangle_\mu
\end{aligned}
$$

For the third step the detailed balance assumptions was used. This means that all eigenvectors of T are orthogonal and the eigenvalues are in the range of $-1 \leq \lambda_i \leq 1$. It follows from the definition that T and Q have the same eigenvalues and that their eigenvectors Ψ_i and Φ_i differ by the stationary distribution.

$$
\begin{aligned}
Q\Phi_i(\boldsymbol{x}) &= \lambda_i\Phi_i(\boldsymbol{x}) \\
T\Psi_i(\boldsymbol{x}) &= \lambda_i\Psi_i(\boldsymbol{x}) \\
\Phi_i(\boldsymbol{x}) &= \mu(\boldsymbol{x})\Psi_i(\boldsymbol{x})
\end{aligned}
$$

Because T is self-adjoint the eigenfunctions Ψ_i are orthogonal to each other and T can be decomposed into it's eigenfunctions

$$
u_{t+k\tau_{MSM}} = [T(\tau_{MSM})]^k \circ u_t = \sum_{i=1}^{n} \lambda_i^k \langle u_t, \Psi_i\rangle_\mu \Psi_i(\boldsymbol{x}) \tag{2.7}
$$

Every physical system has to reach equilibrium for $t \to \infty$. Ergodicity states that there can be only one eigenvalue that is equal to one $\lambda_i = 1$. If two eigenvalues were exactly one then there would exist two disconnected subsets in Ω and it would be possible to construct a trajectory that does not visit every point in Ω in an infinite time. Then a trajectory could be constructed that stays in only one subset and the time and space averages won't be equal anymore. For Q the corresponding eigenfunction is the equilibrium distribution and for T it is a constant. See fig. 2.1 for a 1D energy landscape with 4 wells and the respective eigenfunctions.

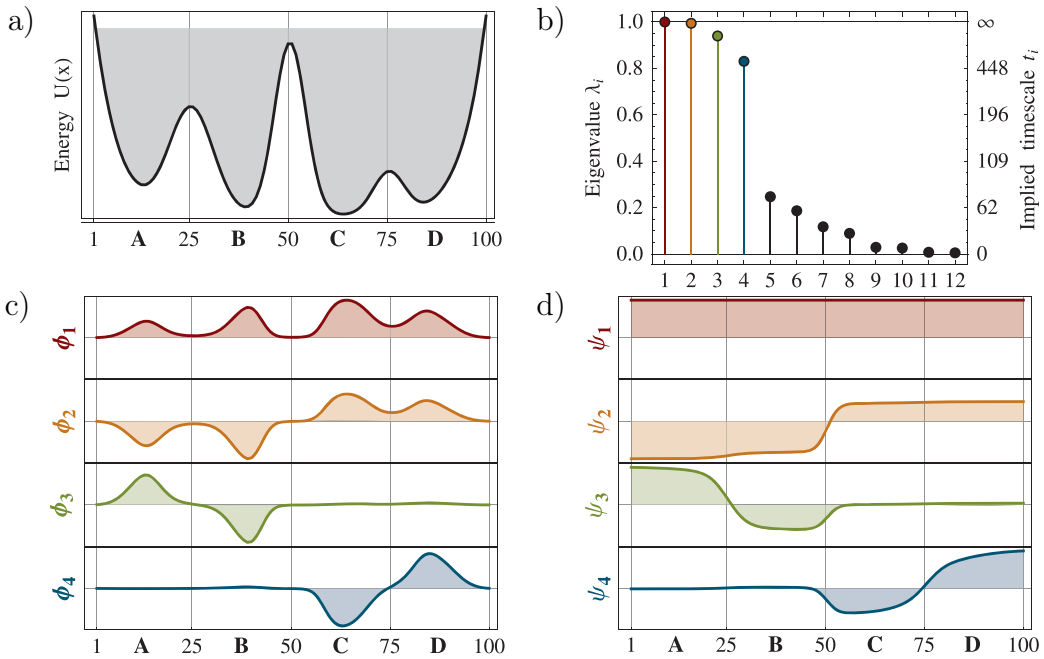


Figure 2.1.: a) one dimensional energy landscape with 4 local minima. b) sorted eigenvalues spectrum of T. c) The first 4 eigenfunctions of Q for this system. Φ_1 is the equilibrium distribution, Φ_2 is the slowest process that describes the transition over the highest barrier from left to right. Φ_3 and Φ_4 describe the transition over the other two barriers. d) Eigenfunctions of T. For the equilibrium process the eigenfunction Ψ_1 is a constant and Ψ_2 til Ψ_4 show how probability is flowing for each process. Source: [8]

Each eigenfunction with a eigenvalue different from 1 represents a process in the system that is decaying over time. The eigenvalues λ_i correspond to physical timescales and it can assigned an implied timescale it_i [8].

$$u_{t+k\tau_{MSM}} = \sum_{i=1}^{n} \exp\left(-\frac{k\tau_{MSM}}{it_i}\right) \langle u_t, \Psi_i \rangle_\mu \Psi_i(\boldsymbol{x})$$
$$\Rightarrow it_i = -\frac{\tau_{MSM}}{\ln \lambda_i} \tag{2.8}$$

This definition includes that the timescale for the equilibrium is infinity. If there is a separation of timescales of the diffusion in a state and the jumps between states then the sorted eigenvalue spectrum will have a gap, see fig. 2.1. The location of the gap will give the number of metastable states. For a simple energy landscape with 4 minima the eigenvalue spectrum shows a gap after the 4th eigenvalue, see fig 2.1 b). The eigenfunctions with an eigenvalue lower then λ_4 are describing the rapid mixing dynamics inside of the states.

This means that by calculating the eigenvalues and eigenfunctions of T we can extract the slow processes of a protein and their timescales.

estimating a MSM from simulation

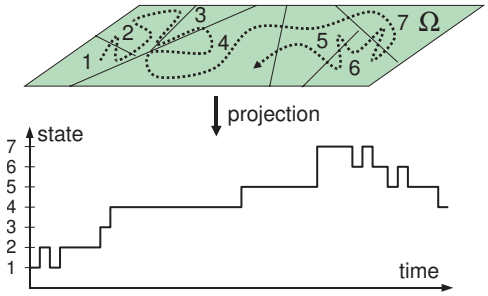


Figure 2.2.: A trajectory in the discretized phase space Ω is projected into the discretization. Source: [8]

To estimate the transfer operator from one or more simulations the conformation space it covers has to be discretized, see fig. 2.2. Typically a crisp discretization like a Voronoi tessellation is used. We will explain the discretization algorithms that we used in this thesis in chap. 3. For a crisp discretization the phase space is portioned into n sets $S = S_1, \ldots, S_n$ such that $\bigcup_{i=1}^{n} S_i = \Omega$ and $S_i \bigcap S_j = \emptyset \ \forall j \neq i$. The

probability for a transition from state i to j is then given by the probability to be in i at time t and j at time $t + \tau_{MSM}$ divided by the probability to be in i at time t [8].

$$\begin{aligned} T_{ij}(\tau_{MSM}) &= \mathbb{P}[\boldsymbol{x}(t+\tau_{MSM}) \in S_j | \boldsymbol{x}(t) \in S_i] \\ &= \frac{\mathbb{P}[\boldsymbol{x}(t+\tau_{MSM}) \in S_j \cap \boldsymbol{x}(t) \in S_i]}{\mathbb{P}[\boldsymbol{x}(t) \in S_i]} \end{aligned} \tag{2.9}$$

These probabilities can be estimated from simulations with the count matrix c_{ij} and the following membership function χ_i:

$$\begin{aligned} \chi_i(\boldsymbol{x}_k) &= \begin{cases} 1 & \boldsymbol{x}_k \in S_i \\ 0 & \text{otherwise} \end{cases} \\ c_{ij}(\tau_{MSM}) = c_{ij}(l\Delta t) &= \sum_{k=1}^{N-l} \chi_i(\boldsymbol{x}_k)\chi_j(\boldsymbol{x}_{k+l}) \end{aligned} \tag{2.10}$$

The count matrices from different simulations can be added up to obtain a combined matrix for the set of simulations. Then $T_{ij}(\tau_{MSM})$ becomes [8].

$$T_{ij}(\tau) = \frac{c_{ij}(\tau)}{\sum_{j=1}^{n} c_{ij}(\tau)} \tag{2.11}$$

choice of τ_{MSM}

We use the Chapman-Kolmogorov equation to construct a simple test to determine which lag-time τ_{MSM} should be chosen to construct a MSM. The equation says that the transfer operator for $n\tau_{MSM}$ is equal to applying the operator for τ_{MSM} n times [20].

$$T(n\tau_{MSM}) = T(\tau_{MSM})^n$$

If this equation is fulfilled the implied timescales for $T(n\tau_{MSM})$ are equal to that of $T(\tau_{MSM})^n$.

$$
\begin{aligned}
it &= \frac{n\tau_{MSM}}{-\ln \lambda_{i,T(n\tau_{MSM})}} = \frac{n\tau_{MSM}}{-\ln \lambda^{n}_{i,T(\tau_{MSM})}} \\
&= \frac{n\tau_{MSM}}{-n \ln \lambda_{i,T(\tau_{MSM})}} = \frac{\tau_{MSM}}{-\ln \lambda_{i,T(\tau_{MSM})}}
\end{aligned}
\tag{2.12}
$$

This is a necessary condition for a system to fulfill the Markov property but it is not sufficient. Fig 2.3 shows an example how the implied timescales behave as a function of the lag-time τ_{MSM}. For small τ_{MSM} the implied timescale will initially rise. When the lag-time is large enough so that the dynamics in a state are rapidly mixing and the probability to jump into any other states become independent from the previous state in the time τ_{MSM} then the implied timescales will be constant. The smallest τ_{MSM} at which this happens should then be chosen to build the MSM. The implied timescale can rise again if the lag-times used are getting larger then the implied timescales. Another reason for the implied timescales to rise again is that the number of statistically independent observed transitions will diminish with increasing lag-times [21].

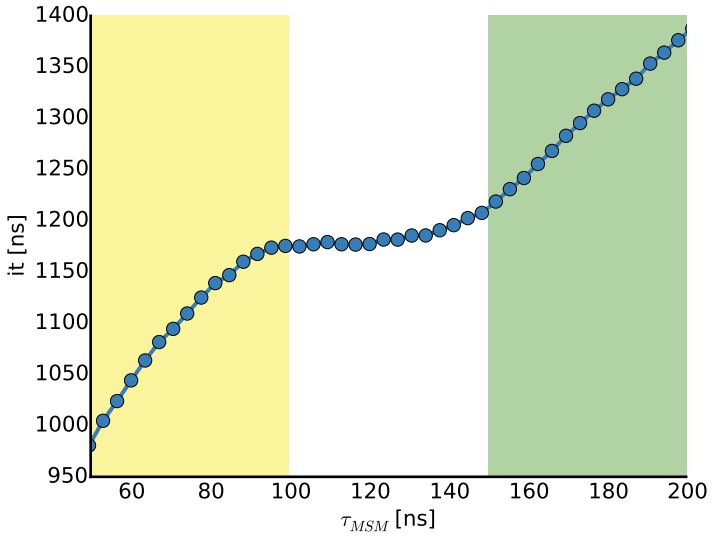


Figure 2.3.: Implied timescales as a function of the lag-time τ_{MSM}. In the yellow region the lag-time τ_{MSM} is chosen so short that the system is not memory free. In the green region the timescale is rising again because either τ_{MSM} is getting larger then the it or sampling is becoming an issue.

2.3. Choice of low dimensional subspace

To build an accurate MSM it is not only important to have a good micro state discretization and chose the correct lag-time τ_{MSM} but also to chose an appropriate low dimensional coordinate space for the discretization. Dihedral angles [22] and RMSD [15] based coordinate spaces have been used in the past. These methods require an insight into the specific protein as it is often not clear in the beginning which sub-space is best or if e.g. should the RSMD only be used from specific parts of the protein [15]. There is no known way to automatically always choose the best coordinate space. Two methods to choose a possible coordinate space are PCA and TICA.

2.3.1. Time-lagged Independent Component Analysis

TICA is a method to determine n slow motions in a protein that are a linear combination of the atomic positions. This linear combination can already be a good approximation of the slow processes describes with MSM that generally are not a simple linear combination of the atomic positions. TICA was used to study slow dynamics in lysine, arginine, ornithine-binding protein [23] and to build MSMs [2].

The TICA components have to fulfill two properties:

- They are uncorrelated at time zero
- Their autocovariance at a fixed lag-time τ_{TICA} are maximal.

The TICA eigenvectors can be obtained by solving the following eigenvalue problem [24]:

$$C(\tau_{TICA})\boldsymbol{v}_i = C(0)\boldsymbol{v}_i\lambda_i \tag{2.13}$$

Where $C(\tau_{TICA})$ is the time-lagged covariance matrix defined as:

$$c_{ij}(\tau_{TICA}) = \langle r_i(t)r_j(t+\tau_{TICA})\rangle \tag{2.14}$$

Where $r_i(t)$ is the mean free i-th atomic coordinate.

2.3.2. Principal Component Analysis

Principal Component Analysis is a common dimension reduction technique for MD-simulations that aims to select n modes containing as much as possible of the variations present in a simulation. Here we assume that the large variance motions identified by the PCA will also be the slowest motions. Fig. 2.4 shows the PCA modes for a two dimensional multivariate Gaussian.

The first mode is determined by a linear function $\boldsymbol{v}_1^T\boldsymbol{x}$ of $\boldsymbol{x}$ that maximizes the variance [13].

$$\boldsymbol{v}_i^T\boldsymbol{x} = \sum_{j=1}^{M} v_{ij}x_j \tag{2.15}$$

The second mode is determined by a linear function $\boldsymbol{v}_2^T\boldsymbol{x}$ that is uncorrelated to $\boldsymbol{v}_1$ and has maximal variance. This continues until the ith linear function $\boldsymbol{v}_i^T\boldsymbol{x}$ that has a maximal variance while being uncorrelated to $\boldsymbol{v}_1^T\boldsymbol{x}$,$\boldsymbol{v}_2^T\boldsymbol{x}$,...,$\boldsymbol{v}_{i-1}^T\boldsymbol{x}$. The functions $\boldsymbol{v}_i$ are the eigenfunctions of the covariance matrix $C(0)$, see eq. 2.14, corresponding to the i-th largest eigenvalue λ_i and the eigenvalues λ_i give the variance in the i-th mode.

$$C(0)\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i \tag{2.16}$$

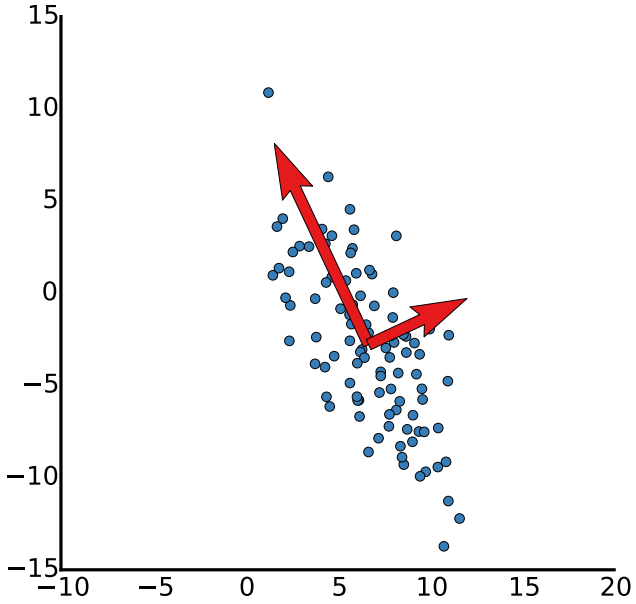


Figure 2.4.: 100 random points from a multivariate Gaussian distribution. The two modes with the largest variance identified by the PCA are shown as red arrows.

3. Methods

3.1. Markov State Model

3.1.1. calculating discretizations

To calculate the discretization of the chosen sub-space we will use the clustering algorithms k-center and k_{means}. Both algorithms will place a pre determined number of cluster centers based on different cost-functions. Each observed point is then designed to the nearest cluster centers. This is called a Voronoi tessellation.

k-center

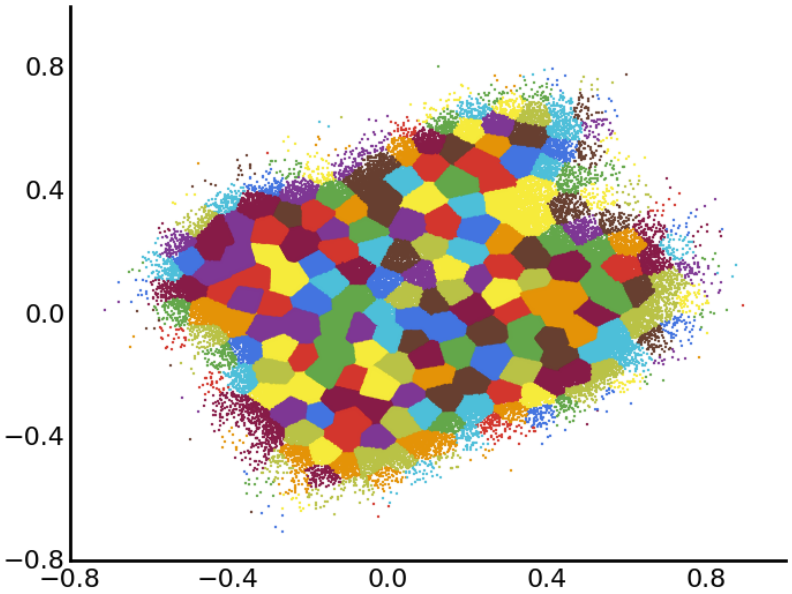


Figure 3.1.: k-center clustering with 300 clusters in the first 2 PCA modes of Ubiquitin. The clusters are evenly sized independent of the local density.

k-center creates clusters with approximately equal radii [25]. This is done by finding a clustering that minimizes the maximal distance of all points in a cluster

to the cluster center. Which can be expressed by the following cost function:

$$C_c(x, \sigma) = \arg\min_S \max_i ||x_i - \sigma(x_i)|| \tag{3.1}$$

S is the set of clusters and σ is a function to map a point x_i to the nearest cluster center. It is important to note that only observations can be cluster centers. An approximate solution to this problem can be implemented with a complexity of $\Omega(kN)$, where k is the number of clusters and N is the number of observations and works as follows:

1. pick a random point as the initial cluster center and assign all other points to that cluster.
2. calculate the distances to the nearest cluster center
3. Choose the point that has the greatest distance to all cluster centers
4. reassign every point to the closest cluster center
5. repeat step 2-3 until a termination criteria is met e.g. number of clusters

Because this clustering method optimizes for the unweighted inner cluster distance it is less likely to assign cluster centers close to each other in regions with a high density. This can help constructing stable MSM because according to Prinz et al. [8] the discretization error can be minimized by using more cluster centers in the transitions regions and only sparsely cluster the metastable regions.

k-means

k-means is minimizing the within-cluster sum of squared distances.

$$\arg\min_S \sum_{i=1}^{k} \sum_{x_j \in S_i} ||x_j - \mu_i||^2 \tag{3.2}$$

Where μ_i is the arithmetic mean of the cluster S_i. The largest distinction to k-center is that the cluster center can be assigned to any point in phase-space. The standard Llyods algorithm is already fast with an average complexity of $\Omega(nkt)$ [26],

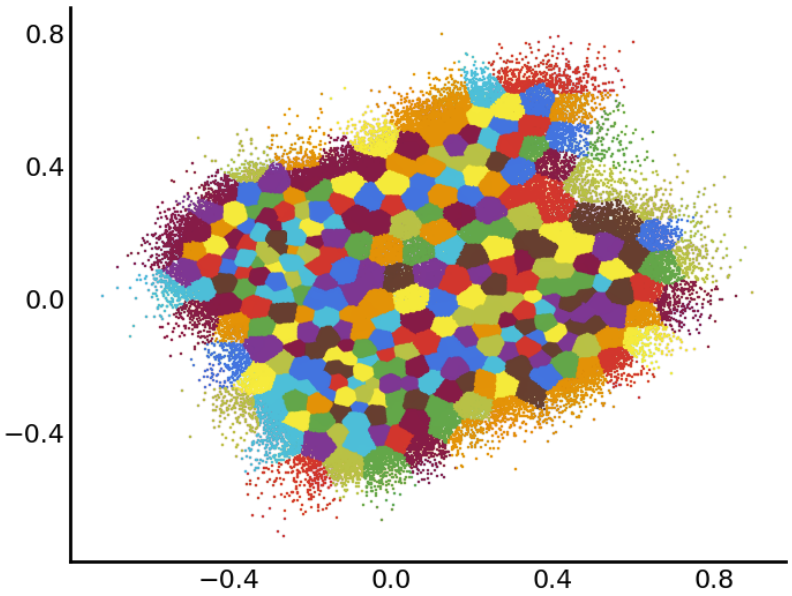


Figure 3.2.: k-means clustering with 300 clusters in the first 2 PCA modes of Ubiquitin. The clusters tend to be smaller in regions where the density is high and larger in places with lower density.

where n is the number of samples, k the number of cluster centers and t the number of iterations. But for large data sets it still takes a considerable amount of time. Because of this we use the Mini-batch-k-means algorithm [27]. This algorithm uses a random batch of points b in each iteration step and therefore has a run-time of $\Omega(bkt)$. The speedup gained by this depends on the ratio between the number of points in a batch and the total number of points. In our case b is much smaller then the total number of observed transitions. The algorithm works as follows.

1. choose n cluster randomly from trajectory
2. pick b random examples
3. calculate nearest center for examples
4. update centers based on examples
5. repeat 2-4 until termination

The k-center clustering can potentially put a lot of cluster centers at the outer edges of the phase space with only a small number of observations in that cluster and few clusters in high density regions with a large number of the total observations.

In comparison k-means will produce a clustering where the number of observations inside a cluster is more evenly distributed. Which in return will give better statistics in the count matrix.

3.1.2. estimating errors

To estimate the statistical error of the count matrix, eq. 2.10, and derived quantities we are using a bootstrap method [28]. We will generate new samples of transition from the original observed transitions and then estimate the error by averaging over the values calculated from the new samples. A new sample is generated by randomly choosing a transition from the original transitions, until the same amount of data as in the original data set is obtained.

3.2. Time-lagged Independent Component Analysis

The time lagged covariance matrix, eq. 2.14, can be directly calculated from a trajectory with:

$$c_{ij} = \frac{1}{N-\tau-1} \sum_{t=1}^{N-\tau} r_i(t) r_j(t+\tau)$$

Where r_i are the mean free observed atomic positions $r_i = x_i - \langle x_i \rangle_T$. To calculate this for an ensemble of trajectories the matrix can be averaged if all the simulations have the same length $C = \frac{1}{M} \sum_{i=1}^{M} \tilde{C}_i$. Assuming that the dynamics are reversible the matrix is symmetrical. For finite data sets, symmetry must be enforced, $C = \frac{1}{2}(\tilde{C} + \tilde{C}^T)$.

choice of τ_{TICA}

Since τ_{TICA} is a free parameter in TICA we need a way to determine a possible choice of τ_{TICA} from the simulation. Our approach is to calculate the ACF from the projections of the first 10 PCA modes and pick the longest estimated autocorrelation time τ of the projections. We will estimate the autocorrelation time from the normalized autocorrelation function.

$$ACF(t) = \langle x_{t'} x_{t'+t} \rangle_T = \frac{1}{N-t-1} \frac{1}{\langle x \rangle} \sum_{i=0}^{T-t} x_i x_{i+t}$$

To estimate the autocorrelation time from the ACF we use 3 different methods that are based on the assumption that the true normalized ACF is an exponential decay.

$$\mathrm{ACF}(t) = e^{-\frac{t}{\tau}} \tag{3.3}$$

The simplest estimate is the time where the ACF first falls below $\frac{1}{e}$.

$$\frac{1}{e} = e^{-\frac{\tau}{\tau}} \Rightarrow \tau = \arg\min_t ACF(t) < \frac{1}{e}$$

The second method is to use the fact that the integral over the exponential decay is equal to the autocorrelation time.

$$\tau = \int_0^\infty \mathrm{ACF}(t) dt$$

For finite data sets this integral cannot be evaluated until infinity. So for the estimates the infinity is replaced with an finite T that is set to the shortest time the ACF falls below 0 or half the time of the simulation.

The third method is to estimate the autocorrelation time only from the initial values. For this we calculate the integral over the ACF up to a time T and then take the Taylor series of this up to the third order. The equation can then be solved for τ.

$$
\begin{aligned}
F &= \int_0^T dt \mathrm{ACF}(t) \\
&= \int_0^T e^{-\frac{t}{\tau}} dt \\
&= \tau(1 - e^{-\frac{T}{\tau}}) \\
&\approx T - \frac{T^2}{2\tau} + \frac{T^3}{6\tau^2} \\
\rightarrow \tau &= \frac{-3T^2 - \sqrt{3(8F - 5T)T^3}}{12(F - T)}
\end{aligned} \tag{3.4}
$$

In an ideal case for T smaller then τ errors are at most 5%. This function only works so long as $(8F - 5T) > 0$, which if F is substituted means that $T \leq \tau$. To determine the optimal T from a simulation we use the largest T that is smaller then $\frac{1}{e}$ and bigger then $8F$.

The 3 algorithms are named as follows in the rest of this thesis.

1. "decay" $\tau_{TICA} = \arg\min_t \mathrm{ACF}(t) < \frac{1}{e}$
2. "integral" $\tau_{TICA} = \int_0^T \mathrm{ACF}(t)dt$
3. "taylor" $\tau_{TICA} = \frac{-3T^2 - \sqrt{3(8F-5T)T^3}}{12(F-T)}$

3.3. Protein setup

We analyzed 100 simulation of Ubiquitin (PDB code 1UBQ) in the NPT ensemble with a pressure of 1 Bar, a temperature of 300 K and a length of 381.4 ns each for a total time of 38.14 μs. The simulations were done using the Gromacs 4.6 molecular dynamics package with the AMBER99SB forcefield [29] and the SPCE water model with an integration step of 4 fs. Snapshots have been recorded every 20 ps.

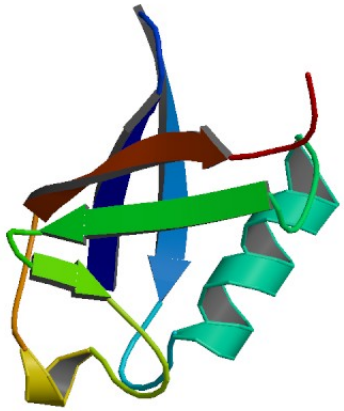


Figure 3.3.: Crystal structure of Ubiquitin from the protein data base (PDB code 1UBQ)

4. Results & Discussion

4.1. TICA

To find out if TICA will find the slow motions for simulations that are not converged we applied TICA and PCA to one simulation of a 100 dimensional random walk in a flat energy landscape and compared them, see fig. 4.1. The random walk was generated using 100 independent 1-dimensional walkers with a uniform stepsize distribution between -1 and 1.

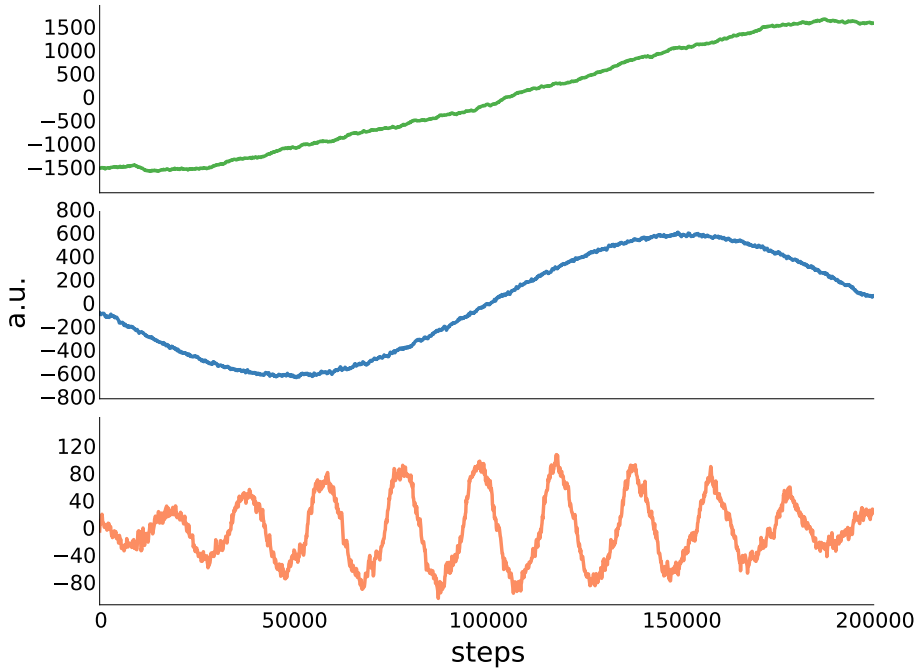


Figure 4.1.: Projection of the first PCA and TICA modes of a 100 dimensional random walk in a flat energy landscape with 200000 steps. green) The projection of the first PCA-mode. blue) The projection of the first TICA-mode built with a lag-time of 5000 steps. Here the mode resembles a sin curve with a full period. orange) Projection of the first TICA-mode built with a lag-time of 20000 steps. The mode shows a higher frequency oscillation

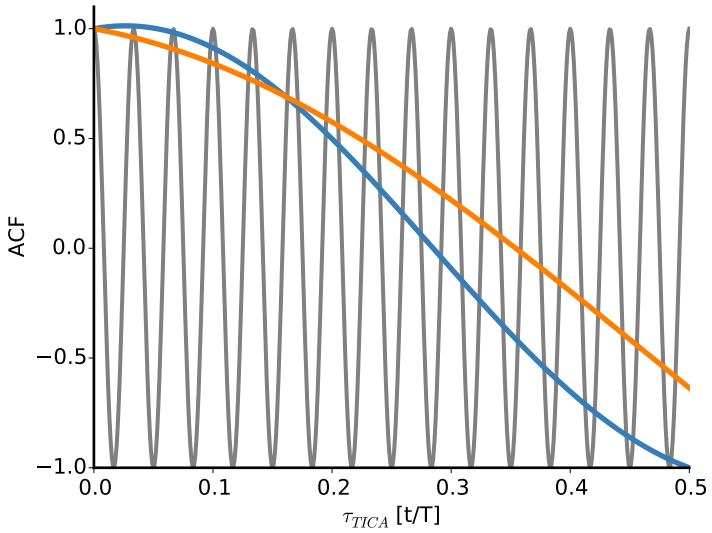


Figure 4.2.: Sketched here is the normalized ACF for periodical signals with a single frequency. The blue curve is the ACF for a sine curve with period 1. The orange curve is the ACF for cosine with period one-half. The ACF for a sine curve with period 30 is shown in gray

The projection of the first PCA resembles a cosine with period one-half. Indeed it is has been shown [30] that the PCA modes for a high dimensional random walk in a flat energy landscape are cosine functions with a period of one half the index of the PCA-mode. The first PCA-mode is the slowest motion in this system all other modes show higher frequency oscillations. If TICA would always find the slowest motion we would expect it to show similar modes as PCA. Fig. 4.1 shows that this is not the case. For small lag-times the first TICA mode is a sine function with period one and for larger lag-times τ_{TICA} the first mode becomes a superposition of periodic functions with a higher frequency.

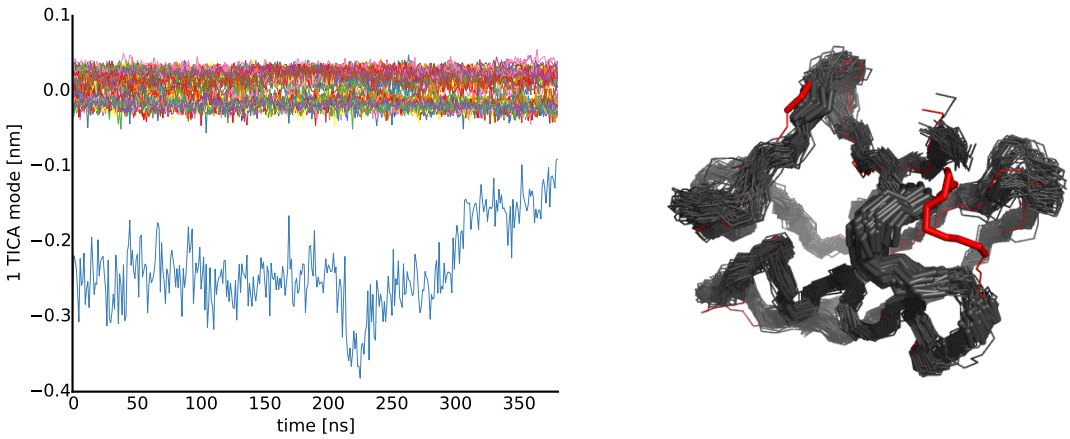
This happens because TICA is optimizing for modes that have the highest ACF value at a lag-time τ_{TICA}. Fig. 4.2 shows a sketch of the ACF for sines and cosines with different periods. For any given time $\tau_{TICA} \in [0, \frac{T}{2}]$ there is a function that has a higher ACF then a cosine with period one, so TICA cannot find this motion. Fig. 4.2 also shows that for small τ_{TICA} a sine curve with period on has the highest ACF value.

This means that TICA does not necessarily find the slowest motion for a simulation that is not converged. Instead TICA is more likely to find motions with a high frequency oscillation.

4.2. Ubiquitin

We want to know if a simulation time less then 100 μs is enough to find slow motions in a protein using MSMs. For this we analyze 100 simulations of Ubiquitin, that were provided by Servaas Michielssens. Each simulation is 380 ns for a total simulation time of 38 μs. We applied TICA and PCA on the C_α-atoms of residue 1-71 to construct different sub-spaces. We then proceeded to build MSM in these sub-spaces with discretizations calculated from the k-means and k-center clustering algorithms.

Fig. 4.3a shows the projection of all simulations onto the first TICA mode. One simulation does not overlap with any of the others in the projection of the first mode. This happens independent of lag-times we used to construct TICA. The lag-times were calculated from the estimated autocorrelation tims of the projections of the first 10 PCA modes with the methods described in sec. 3.2. To check if this wasn't just an artifact of TICA we looked at the backbone of this simulation and compared it to the others, see fig: 4.3b the outlier is shown red. This confirms that this simulation is different from the others. Because MSM are build from observed transitions between different regions of the phase space and this one never transitions to any of the other simulations we exclude it from further analysis.



(a) first TICA mode (b) backbone

Figure 4.3.: a) Projection of all 100 simulations onto the first TICA mode b) Snapshot of all trajectories at 10 ns. Residues with a large contribution to the first TICA mode are drawn thicker. The outlier is shown in red.

4.2.1. Markov State Model

PCA

We constructed 3 sub-spaces using the PCA from the C_α-atoms of residue 1-71. One from the projection onto the first mode, one with the projections onto the first 2 modes and the last one with the projections of the first 3 modes. Each sub-space was discretized with 100 clusters centers using k-means and k-center . 100 lag-times chosen uniformly between 0.1 ns and 150 ns were used to build the MSMs. The mean and standard deviation of the implied timescales were calculated using 100 bootstrap samples for each sub-space, see sec. 3.1.2. The implied timescales for the first two eigenfunctions of the different sub-spaces are shown in fig. 4.4.

The first implied timescales, see blue circles in fig. 4.4, calculated in the sub-space of the first mode are marginally larger then the lag-time τ_{MSM} used to build the MSM with both clustering algorithms. The second implied timescales cannot be resolved because they are always below τ_{MSM}. The standard deviation is under 1 ns for all calculated implied timescales.

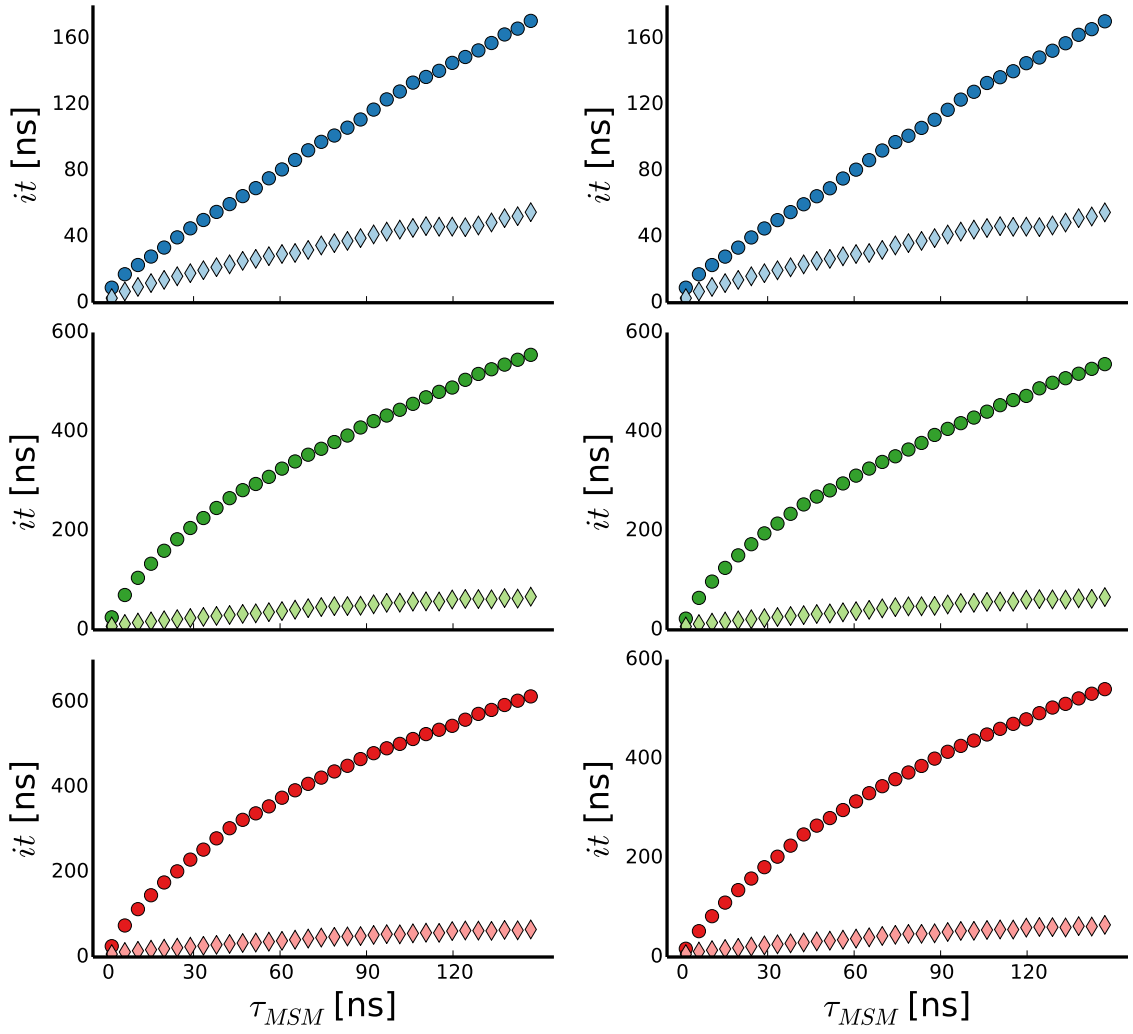
When we cluster the 2 dimensional subspace of the first two PCA modes the first implied timescale, green circles in fig. 4.4, is significantly larger then the lag-times τ_{MSM} and rises to almost 600 ns for the largest lag-time. The second implied timescales cannot be resolved in this sub-space either. This does not change if the first 3 modes are used. The standard deviation is under 3 ns for all calculated implied timescales.

In all used sub-spaces the implied timescales never level-off. This could be because of large internal barriers in one or more micro-states or because the PCA modes are not describing the slow motions in Ubiquitin.

TICA

We constructed TICA modes from the C_α-atoms of residue 1-71 for different lag-times. The lag-times τ_{TICA}, see tab. 4.1, chosen to construct the TICA modes were estimated from the largest autocorrelation time from the projections of the first 10 PCA modes using the methods described in sec. 3.2.

The first 3 modes are parallel independent of the lag-time is chosen to construct TICA, see tab. A.1. That TICA has found the same modes for different lag-times indicates that the motion in these modes has an exponentially decaying autocorrelation function.

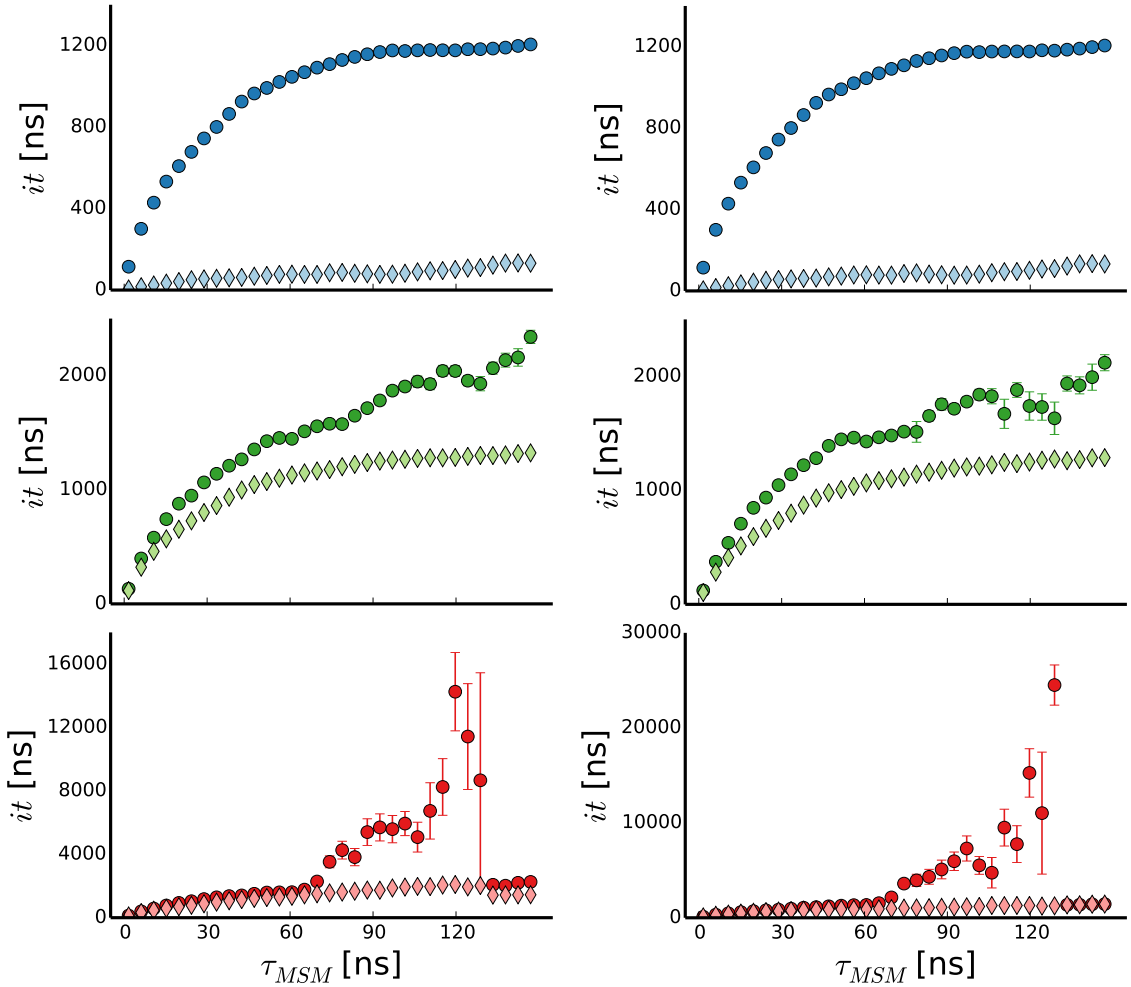


(a) k-means (b) k-center

Figure 4.4.: Implied timescales for MSM's build in different PCA sub-spaces. Blue are the it for MSM's build with the first mode, green for the 2 modes, red for 3 modes. The circles show the it of the first eigenfunction and diamonds the second. The clustering was calculated with 100 clusters-centers for each clustering algorithm and in all sub-spaces.

decay	integral	taylor
9.16 ns	11.06 ns	4.12 ns

Table 4.1.: Longest autocorrelation times calculated from the projections of the first 10 PCA-modes



(a) k-means (b) k-center

Figure 4.5.: Implied timescales for MSM's build in TICA sub-spaces, We used different sub-spaces blue are the it for MSM's build with the first mode, green for the 2 modes, red for 3 modes. The circles show the it of the first eigenfunction and diamonds the second. The clustering was calculated with 100 clusters-centers in all sub-spaces.

We use the first 3 TICA modes to construct the same sub-spaces as described in sec. 4.2.1. We will use the modes calculated with $\tau_{TICA} = 11.06\,\text{ns}$. Because the first 3 modes are independent from the different possible lag-times τ_{TICA} we calculated, we could have used either of the other two lag-times as well. Each sub-space was discretized with 100 clusters centers using k-means and k-center . 100 lag-times chosen uniformly between 0.1 ns and 150 ns were used to reconstruct MSMs. The mean and standard deviation of the implied timescales was calculated using 100 bootstrap samples for each sub-space, see sec. 3.1.2. The implied timescales for the first two eigenfunctions for the different sub-spaces are shown in fig. 4.5.

Using only the first mode the first implied timescale is leveling off at about 1200 ns for $\tau_{MSM} > 91\,\text{ns}$, see the blue circles in fig. 4.5. This behavior is independent of the clustering algorithm. The mean implied timescale for $\tau_{MSM} = 91\,\text{ns}$ is $1165 \pm 4\,\text{ns}$ for both clusterings. The second timescale is not resolved as it always stays below the input lag-time τ_{MSM}.

Using the two dimensional sup-space with the first 2 TICA modes we can resolve the first and second implied timescale, see green circles and diamonds in fig. 4.5. The first implied timescale is not leveling off using either clustering algorithm and reaches values above 2000 ns. The standard deviation are always under 200 ns using the k-center discretization and 80 ns using k-means . The second implied timescale is leveling-off at about 1300 ns for both discretizations.

The first 2 implied timescales of the sub-space constructed with the first 3 modes are shown in red in fig. 4.5. The maximal first implied timescale is reaching values of up to $24\,\mu$s using a k-center clustering and $14\,\mu$s with k-means . In this sub-space the standard deviation is also getting larger, up to 8878 ns using k-means . This is indicating that we have sampling problems and that a few transitions are dominating the first implied timescale.

Fig. 4.6 shows the projection of the first 3 TICA modes for all 99 simulations. There are 8 simulations transitioning to new phase space regions. 4 in mode 2 and 4 different simulations in mode 3. A timescale for the process in a single mode can be estimated from the number of events by dividing the total simulation time by the number of observed events. This gives an order of magnitude estimated for the timescale of $10\,\mu$s with 4 events in about $40\,\mu$s of simulation time. The implied timescales estimated from the MSMs differ from that value by a factor of 3 or less. This means that the timescales estimated with the MSMs are in a reasonable range but more simulations are needed to get a better estimate.

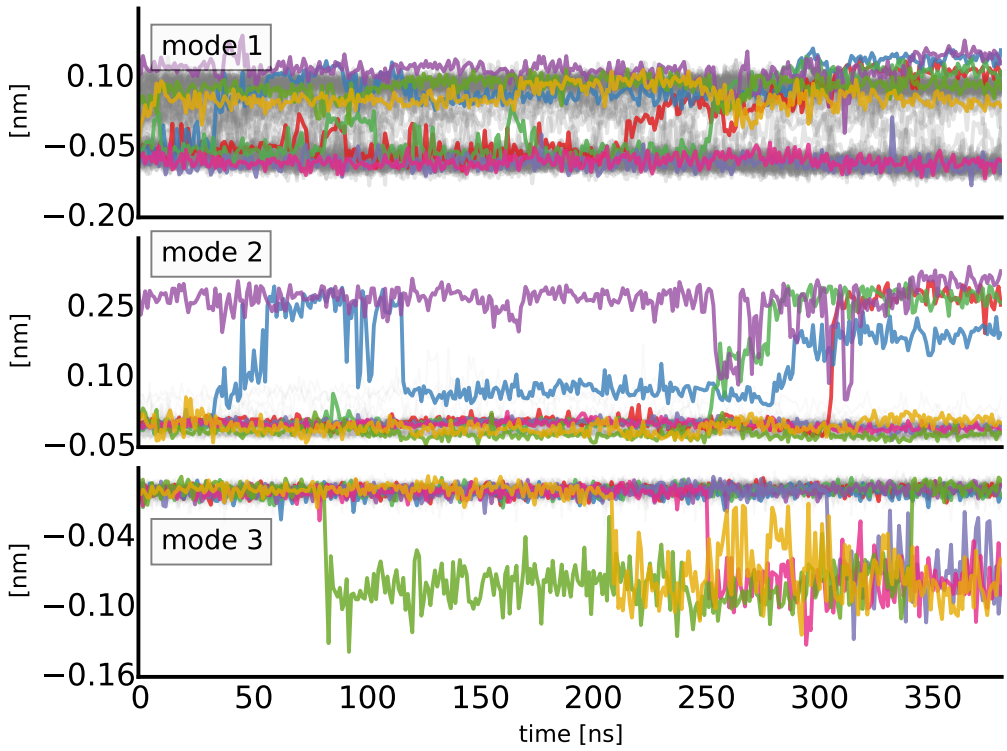


Figure 4.6.: Projection of 99 simulations onto the first 3 TICA modes. Simulations that experience a jump in mode 2 or 3 are drawn with a different color each. Simulations without a jump are down in light gray.

4.2.2. Φ_{52} Ψ_{53} dihedral angle flip

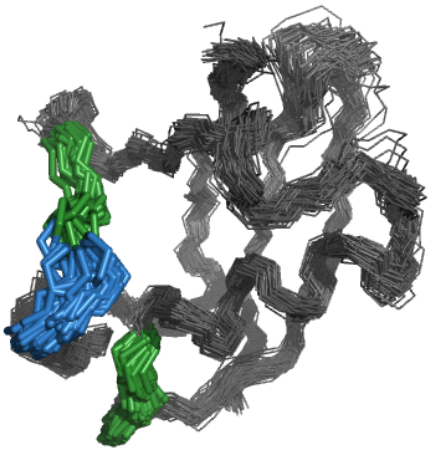


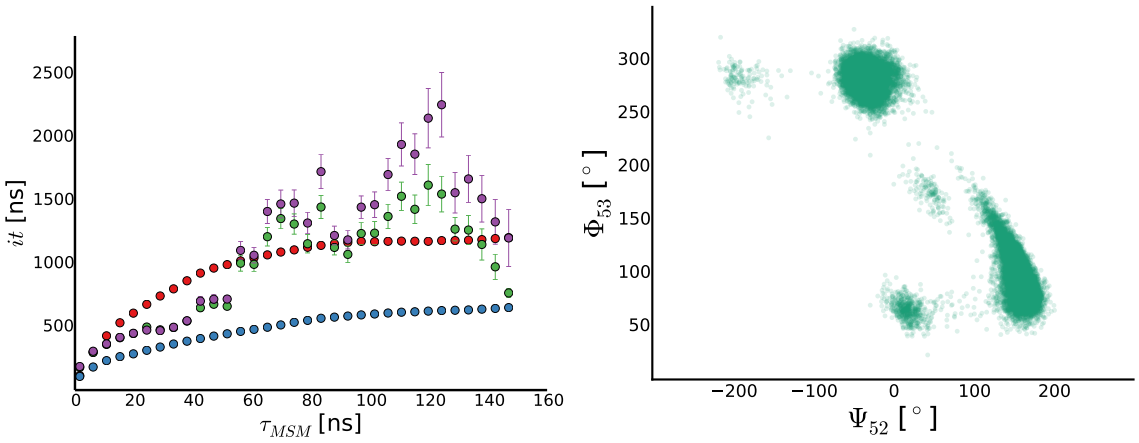
Figure 4.7.: a) The two states of the 53 and 52 residue. b) Backbone snapshot of 99 simulations at 10 ns. Residue 52 and 53 are shown in blue, the residues with the 10 largest contribution to the first TICA mode are green.

The MSMs build with the first TICA mode showed the best convergence in the implied timescales. To understand what motion the first TICA mode is describing we looked at the backbone of the residues from the coordinates with the 10 largest contributions to the first TICA-mode. These residues are 20, 21, 22, 49, 50, 51 and 52. Fig. 4.7 shows a snapshot of all simulations at 10 ns. All of the contributing residues except for 52 are shown in green.

The most notable motion we found is a flip in the dihedral angles Ψ_{52} and Φ_{53}. Residues 52 and 53 are shown in blue in fig. 4.7.

To find out if the timescale of the flip is similar to the timescale calculated for the projection of the first TICA mode we characterized the flip with two states using the difference $\Psi_{52} - \Phi_{53}$. To find a barrier separating the two states we clustered all 99 simulations using k-means with 2 cluster centers, see fig. A.1. The simulations where the distance between the cluster centers is above 100° are have the flip. The barrier is then defined as the mean of the cluster centers in all simulations with a flip.

We calculated MSM and implied timescales with the 2 state model and in the Ψ_{52}, Φ_{53} space like before, sec. 4.2.1. We used 100 lag-times uniformly distributed



(a) (b)

Figure 4.8.: a) First implied timescales in different sub-sapces. (red) the first TICA-mode, see Sec. 4.2.1. (blue) 2 state model of the difference $\Psi_{52} - \Phi_{53}$. (green) k-means discretization with 100 cluster centers in the in Ψ_{52},Φ_{53} space. (purple) k-center discretization with 100 cluster centers. b) Projection of all simulations into the Ψ_{52},Φ_{53} space.

between 0.1 ns and 150 ns and estimated the mean and standard deviation with 100 bootstrap samples, see fig. 4.8a.

The 2 state model has an implied timescale of 655 ± 2 ns for the MSM build with $\tau_{MSM} = 150$ ns. This is about half the value calculated for the first TICA mode using the same τ_{MSM}.

When we use the Ψ_{52},Φ_{53} space and cluster it with 100 cluster centers using k-means and k-center the implied timescales don't converge but fluctuate around value for the first TICA mode and have a large standard deviation.

Fig. 4.8b shows the projection of all simulations in the Ψ_{52}, Φ_{53} space. Building a good clustering in this space is hard because there are 3 irregular shaped regions with a very high density, two small with a lower density and few points in between. k-means will place most cluster centers here into the region with a higher density and very few in the transition regions. k-center on the other hand will place most cluster centers in the sparsely populated transition regions that leads to a handful of clusters containing almost all observations.

Considering this it is surprising that the calculated implied timescales differ only so little from implied timescales calculated with the first TICA mode. This is a good indication that the flip in these dihedral angles is responsible for the motion detected by the first TICA mode.

5. Conclusion

TICA

By applying TICA to a high dimensional random walk with a flat energy surface we could show that the TICA modes do not necessarily correspond to the slowest motions. We showed evidence that this is because TICA is optimizing for the value of the autocorrelation function at a given time τ_{TICA}. This means that TICA will only find the slowest motions if these motions have the largest ACF value at the time τ_{TICA}.

One check to see if TICA indeed found the slowest motions is then to look for oscillations that resemble a sine wave in the projection of first TICA mode. Another is to build TICA with different lag-times and compare the scalar product of the eigenvectors. If the eigenvectors are parallel then TICA has likely found a slow motion.

Ubiquitin

Using PCA on the C_α-atoms we could not identify a slow motion in Ubiquitin. This is not because of bad sampling, since the standard deviation is small, but rather because PCA is not an optimal choice to build MSM for Ubiquitin.

With TICA applied to the C_α-atoms we found a slow motion with a timescale of 1165 ± 4 ns in the first TICA mode. We could also show that this motion is linked to a flip in the Φ_{52} and Ψ_{53} dihedral angles.

TICA also identified one simulation that does not show a transition into a phase space region close to any of the other simulations. When this simulation was excluded from the analysis TICA still found 8 simulations that go into new regions of the phase space. The low number of transitions into these new regions prevented us from getting reliable estimates for the timescale of that motion. This suggest that there are slow motions in Ubiquitin that have a timescale larger then $10\,\mu$s.

This means MSM can be used without accumulating simulation time in the millisecond range. A total simulation time of 38 μs is already enough to study slow motions in the case of Ubiquitin.

6. Outlook

TICA

Formally the general eigenvalue problem for TICA, eq. 2.13 can be interpreted as the search for a coordinate set that maximize the value of the normalized ACF for a specific value of τ_{TICA}.

$$\frac{C(\tau_{TICA})}{C(0)}\mathbf{u}_i = \mathbf{u}_i\lambda_i(\tau_{TICA}) \tag{6.1}$$

Instead it would be possible to optimize for the autocorrelation time using a cross correlation time matrix.

$$C^* = \int_0^\infty C(\tau)d\tau \tag{6.2}$$

$$C^*\mathbf{u}_i = \mathbf{u}_i\lambda_i \tag{6.3}$$

This set of coordinates would be parameter free. This might find a cosine motion for a high dimensional random walk in a flat landscape because the autocorrelation time for the cosine is higher then that of a sine. We would need to check if this can outperform the current TICA algorithm to find good sub-spaces for MSM building with proteins..

Ubiquitin

To resolve the slow transitions and get better estimates of the timescales more sampling is needed for this it would be possible to either spawn new simulations from trajectories were we know that they are at the boundary of the space we sampled so far or we use an adaptive sampling scheme like Copernicus [31].

A. Appendix

	decay vs. integral	decay vs. taylor	taylor vs. integral
1	0.998	0.993	0.986
2	0.999	0.996	0.994
3	1.000	0.999	0.998
4	0.957	0.811	0.641
5	0.976	0.089	0.070

Table A.1.: Absolute value of the scalar product for the first 5 TICA modes. The modes were calculated using different lag-times τ_{TICA}, see Tab. 4.1

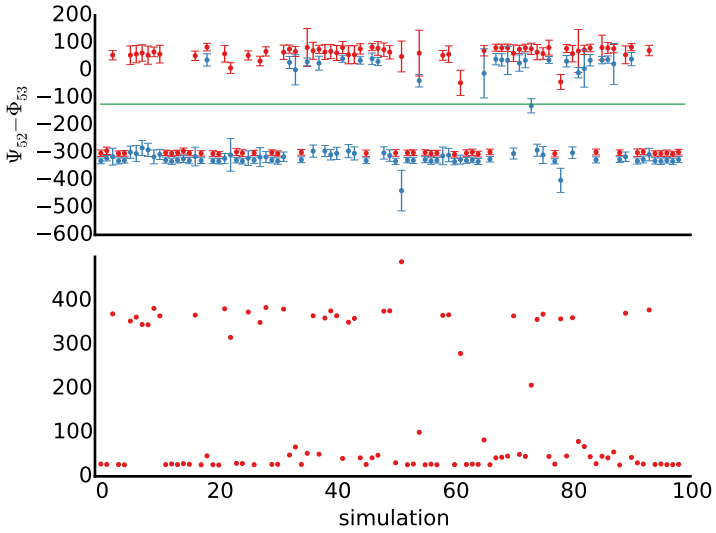


Figure A.1.: Top shows the cluster centers found by k-means for the distance $\Psi_{52} - \Phi_{53}$ for 99 simulations. The cluster centers with the larger value is shown in red and the other in blue. The green line is the mean of the cluster centers for all simulations where the distance between clusters centers is above 100 °. The bottom shows the distance between the cluster centers for each simulation.

Bibliography

[1] Lutz Molgedey and Heinz Georg Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical review letters*, 72(23): 3634, 1994.

[2] Guillermo Perez-Hernandez, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1):015102, 2013.

[3] Oliver F Lange, Nils-Alexander Lakomek, Christophe Farès, Gunnar F Schröder, Korvin FA Walter, Stefan Becker, Jens Meiler, Helmut Grubmüller, Christian Griesinger, and Bert L De Groot. Recognition dynamics up to microseconds revealed from an rdc-derived ubiquitin ensemble in solution. *science*, 320(5882):1471–1475, 2008.

[4] Jan H Peters and Bert L De Groot. Ubiquitin dynamics in complexes reveal molecular recognition mechanisms beyond induced fit and conformational selection. *PLoS computational biology*, 8(10):e1002704, 2012.

[5] Hans Frauenfelder, Fritz Parak, and Robert D Young. Conformational substates in proteins. *Annual review of biophysics and biophysical chemistry*, 17(1):451–479, 1988.

[6] Peter Deuflhard, Michael Dellnitz, Oliver Junge, and Christof Schütte. Computation of essential molecular dynamics by subdivision techniques. In *Computational molecular dynamics: challenges, methods, ideas*, pages 98–115. Springer, 1999.

[7] Ch Schütte, Alexander Fischer, Wilhelm Huisinga, and Peter Deuflhard. A direct approach to conformational dynamics based on hybrid monte carlo. *Journal of Computational Physics*, 151(1):146–168, 1999.

[8] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of chemical physics*, 134(17):174105, 2011.

[9] Susanna Kube and Marcus Weber. A coarse graining method for the identification of transition rates between molecular conformations. *The Journal of chemical physics*, 126(2):024103, 2007.

[10] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *Database Theory—ICDT'99*, pages 217–235. Springer, 1999.

[11] Sidney P Elmer, Sanghyun Park, and Vijay S Pande. Foldamer dynamics expressed via markov state models. i. explicit solvent molecular-dynamics simulations in acetonitrile, chloroform, methanol, and water. *The Journal of chemical physics*, 123(11):114902, 2005.

[12] Gregory R Bowman, Xuhui Huang, and Vijay S Pande. Using generalized ensemble simulations and markov state models to identify conformational states. *Methods*, 49(2):197–201, 2009.

[13] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[14] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons, 2001.

[15] Kai J Kohlhoff, Diwakar Shukla, Morgan Lawrenz, Gregory R Bowman, David E Konerding, Dan Belov, Russ B Altman, and Vijay S Pande. Cloud-based simulations on google exacycle reveal ligand modulation of gpcr activation pathways. *Nature chemistry*, 6(1):15–21, 2014.

[16] David De Sancho, Jeetain Mittal, and Robert B Best. Folding kinetics and unfolded state dynamics of the gb1 hairpin from molecular simulation. *Journal of Chemical Theory and Computation*, 9(3):1743–1753, 2013.

[17] Robert Zwanzig. *Nonequilibrium statistical mechanics*. Oxford University Press, 2001.

[18] Albert Einstein. Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der physik*, 322(8):549–560, 1905.

[19] Manfred Holz, Stefan R Heil, and Antonio Sacco. Temperature-dependent self-diffusion coefficients of water and six selected molecular liquids for calibration in accurate 1h nmr pfg measurements. *Physical Chemistry Chemical Physics*, 2(20):4740–4742, 2000.

[20] GregoryR. Bowman. An overview and practical guide to building markov state models. In Gregory R. Bowman, Vijay S. Pande, and Frank Noé, editors, *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, volume 797 of *Advances in Experimental Medicine and Biology*, pages 7–22. Springer Netherlands, 2014. ISBN 978-94-007-7605-0. doi: 10.1007/978-94-007-7606-7_2. URL `http://dx.doi.org/10.1007/978-94-007-7606-7_2`.

[21] John D. Chodera, Nina Singhal, Vijay S. Pande, Ken A. Dill, and William C. Swope. Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics. *The Journal of Chemical Physics*, 126(15):155101, 2007. doi: http://dx.doi.org/10.1063/1.2714538. URL `http://scitation.aip.org/content/aip/journal/jcp/126/15/10.1063/1.2714538`.

[22] Jan-Hendrik Prinz, John D Chodera, Vijay S Pande, William C Swope, Jeremy C Smith, and Frank Noé. Optimal use of data in parallel tempering simulations for the construction of discrete-state markov models of biomolecular dynamics. *The Journal of chemical physics*, 134(24):244108, 2011.

[23] Yusuke Naritomi and Sotaro Fuchigami. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *The Journal of chemical physics*, 134:065101, 2011.

[24] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:3634–3637, Jun 1994. doi: 10.1103/PhysRevLett.72.3634. URL `http://link.aps.org/doi/10.1103/PhysRevLett.72.3634`.

[25] Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[26] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[27] D Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[28] Bradley Efron. Computers and the theory of statistics: thinking the unthinkable. *Siam Review*, 21(4):460–480, 1979.

[29] Viktor Hornak, Robert Abel, Asim Okur, Bentley Strockbine, Adrian Roitberg, and Carlos Simmerling. Comparison of multiple amber force fields and development of improved protein backbone parameters. *Proteins: Structure, Function, and Bioinformatics*, 65(3):712–725, 2006.

[30] Berk Hess. Similarities between principal components of protein dynamics and random diffusion. *Physical Review E*, 62(6):8438, 2000.

[31] Sander Pronk, Per Larsson, Iman Pouya, Gregory R Bowman, Imran S Haque, Kyle Beauchamp, Berk Hess, Vijay S Pande, Peter M Kasson, and Erik Lindahl. Copernicus: A new paradigm for parallel adaptive molecular dynamics. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 60. ACM, 2011.

[32] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[33] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007. ISSN 1521-9615. doi: 10.1109/MCSE.2007.53. URL `http://ipython.org`.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] J.S. Seabold and J. Perktold. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, 2010.

[36] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2010.118.

[37] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

Acknowledgment

I would like to thank Helmut Grubmüller for giving me the chance to work on this project. Servaas Michielssens for the Ubiquitin simulations that he provided. Jan Henning Peters and Colin Smith for discussions about Ubiquitin. I would also like to thank Béla Voß, Andreas Volkhardt and Christian Blau for general discussions.

Last but not least I would like to thank everyone who helped me during my master's thesis, especially the whole Department for theoretical and computational biophysics at the MPI for biophysical chemistry for it's great working atmosphere.

Erklärung nach §18(8) der Prüfungsordnung für den Bachelor-Studiengang Physik und den Master-Studiengang Physik an der Universität Göttingen:

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe.
Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, den August 17, 2014

(Max Linke)