Master’s Thesis

Markov State Models for Comparing Protein Dynamics

Vergleich von Proteindynamik mithilfe von Markov-Modellen

prepared by

Daniel Quetschlich
from Bremen

at the Max-Planck-Institute for Biophysical Chemistry

Submission date: 22nd September 2016

First Referee: Prof. Dr. Helmut Grubmüller

Second Referee: Prof. Dr. Bert de Groot
Abstract

Proteins are the nanomachines of the cell [1, 2]. Similar to macroscopic machines their function is directly linked to the motion of the protein. To explore this link this thesis investigates the motion patterns of a test set of 592 proteins. The test set comprises a simulated trajectory of 1 µs for each of the proteins carried out by atomistic molecular dynamics simulations. The employed tool for the comparison are Markov State Models (MSM) [3] which describe protein dynamics by a set of discrete states and transition rates between these states. These transition rates also encode the timescales of slow processes of the protein dynamics. For the derivation of the MSMs we used a protocol which consisted of Time-structure based Independent Component Analysis (TICA) [4] coordinate transformation, a microstate discretisation and a Bayesian transition matrix estimator. In the first step we found optimal parameters for the methods in the protocol. After that a Markov State Model was determined for each protein and the ten longest timescales of each protein were used as metric for protein dynamics. A principle component analysis revealed that most of the variance is given by the five longest timescales. However, the space of the timescales is populated continuously and no clusters of distinct protein classes could be found.

Keywords: Physics, Markov State Models, Proteins, Protein Dynamics, Molecular Dynamics, Dynasome, Time-structure based Independent Component Analysis
Contents

1 Introduction 1

2 Theoretical Foundations 3
  2.1 Markov State Models for Protein Dynamics 3
    2.1.1 Properties of the Propagator 5
  2.2 Markov State Models from Trajectories 7
  2.3 Time-structure based Independent Component Analysis (TICA) 10
  2.4 Discretisation 11

3 Methods 13
  3.1 Time-structure based Independent Component Analysis (TICA) 14
  3.2 K-Means Clustering Algorithm 15
  3.3 Estimation of the Transition Matrix 16
  3.4 Implied Timescales to Test Markovity of Transition Matrices 17
  3.5 Test Set 18

4 Results & Discussion 21
  4.1 Parameters for MSM Estimation 21
    4.1.1 TICA Coordinates 21
    4.1.2 Discretisation into Microstates 25
    4.1.3 Estimation of the Transition Matrix 26
    4.1.4 Discussion of the Protocol 31
  4.2 Comparison of All Proteins by Implied Timescales 32

5 Conclusions & Outlook 37

6 Appendix 39
  6.1 Negative Eigenvalues - 2 State Model 39
1 Introduction

To understand common global properties of certain objects it is sometimes necessary to compare a large number of these objects. An example for this principle is the Hertzsprung-Russell diagram. The Hertzsprung-Russell diagram is a scatter plot of star spectrum and star brightness \[5, 6\] which allows to understand the life cycle of stars \[7\]. In this work, we transfer this strategy to compare the motion pattern of an important class of biological macromolecules: proteins.

Proteins are the nanomachines of the cell \[1, 2\]. Protein functions range from metabolism where enzymes catalyse chemical reactions, to the creation of forces in the cytoskeleton \[8\]. Despite their various functions, all proteins share the same building blocks: a sequence of amino acids which folds into a three-dimensional structure \[9, pp. 69-80\]. At room temperature these structures are not static but dynamic. Their motion is directly linked to function, similar to macroscopic machines like the engine of a car.

Similar proteins exhibit similar functions. Generally, protein resemblance can be analysed on a sequence, structural and dynamical level. While sequence comparison is already a powerful tool to investigate protein function \[10\], Cothia and Lesk \[11\] pointed out, that protein structure alters less than protein sequence during evolutionary processes. The fact that protein dynamics realises protein function as an intermediate stage sparks the hope that protein dynamics based analysis methods will correlate even better with protein function than structure. Protein dynamics can be determined from molecular dynamics (MD) simulations of protein structures in solvent. Here, we compare and analyse the trajectories from MD simulations of different proteins.

However, different proteins have different structures. Therefore, it is not clear a priori how to compare the dynamics of different proteins in a meaningful way.
1 Introduction

So far, there are only few attempts to compare different protein functions through their dynamics [12–14]. In a recent study, the Dynasome project [14], investigated into the link between dynamics and function through a set 34 variables which have been chosen to describe the dynamics. This description has been applied to a test set of 112 proteins with 100 ns trajectories each. Despite the short trajectories their results already yield a fingerprint of protein dynamics and allows a success rate of 46% to classify proteins function by their dynamics. However, choosing a “patchwork” of variables contains some arbitrariness and is therefore not completely satisfying.

This thesis aims at extending the Dynasome project by applying a more systematic description of protein dynamics on an expanded data set.

To this aim we employ Markov State Models [3]. MSMs consist of a finite set of discrete states and transitions between these states. So they provide a coarsened picture of the high dimensional dynamics. The states and the rates are directly derived from the kinetics of a protein. This makes MSMs an excellent tool to model protein dynamics.

In this project we derived MSMs for a test set of 592 proteins from molecular dynamics trajectories of 1 μs per protein. As a first step, we applied Time-Lagged Independent Component Analysis transformation and subsequently performed k-means clustering on the TICA-projected space. With this discretisation the transition matrix estimation was done by a Bayesian estimator.

We applied a two stage procedure. First, suitable discretisation and estimation parameters which are consistent for the whole test set were determined. This step forms the main part of this work. Afterwards the MSMs were analysed by comparing the timescales of the obtained Markov Models for all proteins.

The remainder of this thesis is structured as follows: chapter 2 introduces the theoretical framework of Markov State Models for proteins. In chapter 3 the applied protocol is explained. Results and discussion can be found in chapter 4. Chapter 5 completes the thesis by outlining conclusions and an outlook for future research.
2 Theoretical Foundations

This chapter introduces the applied theoretical methods starting with Markov state models. The presentation in this chapter is adapted from chapter 3 and 4 of a book on MSMs for protein dynamics by Bowman et al [3].

2.1 Markov State Models for Protein Dynamics

Markov State Models are history-independent stochastic models which consist of a network of states and transition rates between these states. For protein dynamics a useful picture is to think of these states as local minima in the free energy landscape in the 3N-dimensional space of the atomic coordinates $\Omega$ of the considered protein. Here $N$ is the number of atoms in the protein. These energy minima form metastable states, i.e. conformations, which are separated by high energy barriers. The aim of MSMs is to estimate the metastable states and their transition rates from a discretisation of the space $\Omega$ and a transition matrix which comprises the probabilities of jumps between the discrete states.

Before addressing the mathematical details of MSM in depth, a toy model is given in figure 2.1 to achieve some understanding about the crucial dynamics and the information we want to extract. The toy model consists of a particle performing a one dimensional random walk in a free energy landscape as shown in figure 2.1.

The main features of the landscape are four states separated by three barriers of different heights. By knowing the equilibrium distribution and the transition rates between the states, we can get a good understanding on the dynamics. This information is encoded in the propagator.
Theoretical Foundations

Figure 2.1: Free energy landscape (F(x), top), equilibrium distribution (µ(x)), eigenvalues (φ_j) and eigenvalues (λ_j) of a 1D toy model [3, Chap. 3].

The propagator P(τ) is an operator which describes the time evolution of the ensemble density ρ_{t+τ}(y) in atomic coordinates space Ω after lag time τ:

ρ_{t+τ}(y) = P(τ) ∘ ρ_t(y) = \int_{Ω} p(\vec{x} \rightarrow \vec{y}, τ)ρ_t(\vec{x})d\vec{x}.

Here the ensemble density as a function of the position \vec{x} is denoted as ρ_t(\vec{x}) and the transition probability (in time τ) from position \vec{x} to position \vec{y} as p(\vec{x} \rightarrow \vec{y}, τ).

The eigenvectors and eigenvalues of the propagator contain information on states and rates of dynamical processes in the protein, respectively. The first four eigenvectors of P(τ) are given in 2.1 (middle) and the spectrum of the operator in 2.1 (bottom). The eigenvectors in 2.1 (middle) show that the largest eigenvalue of P(τ) equals one and the corresponding eigenvector the stationary distribution. The three consequent eigenvectors have a negative support on one side and positive support on the other side of the respective barrier. This means that the propagator transports ensemble density across the barrier. Note that a larger eigenvalue corresponds to a higher barrier. Thus, slower processes have higher eigenvalues and smaller eigenvalues describe faster processes.

In this example the main features of the energy landscape are encoded in the largest eigenvalues. This fact is assumed to be a general feature in the framework of Markov State models. The rationale is that functionally relevant information of protein dynamics is contained in the large eigenvalues because they describe...
the transition between metastable states. Furthermore, this example shows that the propagator of this simple system has a spectrum dominated by the first four eigenvalues and followed by a large gap and a rather continuous spectrum of small eigenvalues. This is not necessarily true for proteins which exhibit motions on many time scales.

### 2.1.1 Properties of the Propagator

We assume that the propagator satisfies three main assumptions: ergodicity, history independence (i.e. Markovity) and microscopically reversible dynamics (detailed balance). These assumptions will be discussed below.

**Ergodicity**

The first assumption is that ergodicity is satisfied. This means that a time average of some quantity equals the ensemble average. Ergodicity also implies the existence of a stationary distribution $\mu(\vec{x})$, the Boltzmann distribution which mirrors the free-energy landscape $F(\vec{x})$

$$\mu(\vec{x}) = \frac{\exp(-\beta F(\vec{x}))}{Z(\vec{x})},$$

(2.1)

where $Z(\vec{x})$ is the partition function and $\beta = 1/k_B T$ the thermodynamic beta.

**Markovity**

The Markov property means that there is no history dependence for evolution of the density $\rho_{t+\tau}(\vec{x})$, it only depends on the current density $\rho_t(\vec{x})$. So it is possible to write the propagator as projection on the eigenvectors

$$\rho_{t+\tau}(\vec{x}) = P(\tau) \circ \rho_t(\vec{x}) \approx \sum_{j=1}^{\infty} \lambda_j \cdot \langle \phi_j, \rho_t(\vec{x}) \rangle_{\mu} \cdot \rho_t(\vec{x}).$$

In this equation $\phi_j$ and $\lambda_j$ denote eigenvectors and eigenvalues of the propagator, respectively. The applied scalar product $\langle \cdot, \cdot \rangle_{\mu}$ is defined by

$$\langle \rho_t, \rho'_t \rangle_{\mu} = \int_{\Omega} \frac{\rho_t(\vec{x}) \cdot \rho'_t(\vec{x})}{\mu(\vec{x})} d\vec{x}.$$

(2.2)
The advantage of this scalar product is, that it makes the propagator Hermitian, as it will be pointed out in section Detailed Balance.

Markovity implies that the density after multiples of the lag time $k \cdot \tau$ is given by $k$ times applying $P(\tau)$, i.e. the Chapman-Kolomogorov equation is fulfilled [3, Chap. 3]:

$$\rho_{t+k\tau}(\vec{x}) = P(\tau)^k \circ \rho_t(\vec{x}).$$

Therefore, the propagation of the ensemble density in time is given by

$$\rho_{t+k\tau}(\vec{x}) = P(\tau)^k \circ \rho_t(\vec{x}) \approx \sum_{j=1}^{\infty} \lambda_j^k \cdot \langle \phi_j, \rho_t(\vec{x}) \rangle \mu \cdot \rho_t(\vec{x}). \quad (2.3)$$

The eigenvalues of the propagator are limited by $|\lambda_j| \leq 1$ because the propagator describes a stochastic process [15], with an eigenvalue of one for the stationary distribution and small eigenvalues for fast processes. The eigenvalues are directly linked to implied timescales of the dynamics

$$t_j = -\frac{\tau}{\ln(\lambda_j)}.$$ 

These timescales can be compared to experimental results or can be used as a verification method for Markov State Models (see section 3.4) [16]. Apart from, using these timescales in equation (2.3) to yields the final form of the propagator

$$\rho_{t+k\tau}(\vec{x}) \approx \sum_{j=1}^{\infty} e^{-\frac{k}{t_j}} \cdot \langle \phi_j, \rho_t(\vec{x}) \rangle \mu \cdot \rho_t(\vec{x}).$$

So, the implied timescales give information how fast the different processes decay into equilibrium.

**Detailed Balance**

The third assumption is, that detailed balance is satisfied. This condition means that in equilibrium the probability flux from $\vec{x}$ to $\vec{y}$ equals the reverse flux the flux from $\vec{y}$ to $\vec{x}$. For the transition probabilities $P(\vec{x} \rightarrow \vec{y})$ it follows that [3, Chap. 3]:

$$\mu(\vec{x}) \cdot P(\vec{x} \rightarrow \vec{y}, \tau) = \mu(\vec{y}) \cdot P(\vec{y} \rightarrow \vec{x}, \tau). \quad (2.4)$$
Here, $\mu(\vec{x})$ and $\mu(\vec{y})$ denote the equilibrium probability to be in the position $\vec{x}$ and $\vec{y}$, respectively.

Detailed balance is important because the propagator is not Hermitian under the usual scalar product $\langle \rho_t(\vec{x}), \rho'_t(\vec{x}) \rangle = \int_\Omega \rho_t(\vec{x}) \cdot \rho'_t(\vec{x}) \, d\vec{x}$. Hermiticity is desirable because it means that the eigenvalues are real and there is a partition of unity. The existence of a partition of unity ensures that we are able to find a partition into eigenvectors.

In order to enforce Hermiticity we make use of detailed balance and consider $P(\vec{x} \to \vec{y}, \tau) / \mu(\vec{y})$ instead of $P(\vec{x} \to \vec{y}, \tau)$. Now the operator is Hermitian with the scalar product $\langle \cdot, \cdot \rangle$. This is equivalent to defining the scalar product $\langle \cdot, \cdot \rangle_\mu$ as stated in equation (2.2).

### 2.2 Markov State Models from Trajectories

To estimate the propagator from simulations such as MD, it is useful to define the transfer operator $T(\tau)$ because this operator can directly be estimated from MD trajectory data [3, Chap. 3]. This operator is closely linked to the propagator, but instead of propagating densities, it describes the time evolution of membership functions $u_{t+\tau}(\vec{x}) = T(\tau) \circ u_t(\vec{x})$. These membership functions are defined by

$$u_t(\vec{x}) = \frac{\rho_t(\vec{x})}{\mu(\vec{x})}.$$

The propagator and the transfer operator share the same eigenvalues $\lambda_j$ [3, Chap. 3]. So both operators contain the same information on the dynamics. Furthermore, the corresponding eigenvectors of the transfer operator are linked in the same manner as the $u_t$ and $\rho_t$ [3, Chap. 3]:

$$\phi_j(\vec{x}) = \frac{\psi_j(\vec{x})}{\mu(\vec{x})}.$$

The $u_t(\vec{x})$ is almost flat within a metastable state because the local density distribution is divided out. This is sketched in figure 2.2 where we can see the transformation of the eigenvectors of the example above. The transfer operator eigenvectors are flat in the energy minima, so the information on the distribution
Figure 2.2: The transformation from the propagator eigenvectors $\phi_j$ to the transfer operator eigenvectors $\psi_j$. Source: [3, Chap. 3, changed]

Figure 2.3: Sketch of the discretisation and the jump process. The trajectory is shown as a dashed line. Source: [3 Chap. 3]

within the metastable states is lost and only the information from where to where density is transported remains.

The flatness within a metastable state is crucial, since it allows us to approximate the space $\Omega$ by a set of crisp set functions to estimate Markov State Models from a continuous trajectory. To this end it is necessary to discretise the space and count the transitions between these states $S_i$. So, the transfer operator is approximated by a discrete jump process. This is sketched in figure 2.3. The space is discretised into 7 states (top) and we only see the projection in the discrete state as a function of time (bottom).

Generally, the first step is to find a discretisation with a finite number of states $n$. Within the discrete states the density of states is approximated by membership functions. This is carried out by a set of crisp step functions $\chi_i(\vec{x})$ which mirror the
2.2 Markov State Models from Trajectories

projection into discrete states

$$\chi_i(\vec{x}) = \begin{cases} 1 & \text{for } \vec{x} \in S_i \\ 0 & \text{else} \end{cases}$$

The membership functions have the property that $$\sum_{i=1}^{n} \chi_i = 1$$ because the states should form a pairwise disjoint tessellation of the whole space $$\Omega$$. With such discretisation the transfer operator can be approximated by a transfer matrix of the jump process between the discrete states

$$T_{ij}(\tau) = \frac{\langle \chi_i, T(\tau) \circ \chi_j \rangle \mu}{\langle \chi_i, \chi_i \rangle \mu}.$$ 

By using ergodicity this transition matrix can be estimated from the trajectory. Here, we will use a Bayesian estimator. This estimator maximises the probability of obtaining a transition matrix $$T(\tau)$$ given a trajectory $$\vec{Y}$$:

$$\arg \max_{T(\tau)} P(T(\tau)|\vec{Y}) \propto P(T) \cdot P(\vec{Y}|T(\tau)). \quad (2.5)$$

The prior is denominated as $$P(T(\tau))$$ and the probability to find a trajectory for a given transition matrix as $$P(\vec{Y}|T(\tau))$$. For reasons of mathematical convenience, the prior is noted in the same form as $$P(\vec{Y}|T(\tau))$$

$$P(\vec{Y}|T(\tau)) = \prod_{i,j=1}^{n} T_{ij}(\tau)^{c_{ij}^{obs}} \quad \text{and} \quad P(T(\tau)) = \prod_{i,j=1}^{n} T_{ij}(\tau)^{c_{ij}^{prior}}. \quad (2.6)$$

Here, the $$c_{ij}^{obs}$$ and $$c_{ij}^{prior}$$ denote the counts from the trajectory and the prior, respectively. The elements of the transition matrix $$T_{ij}(\tau)$$ can now be determined by solving equation [2.5] with the constraint that $$\sum_{i=1}^{n} T_{ij}(\tau) = 1$$ (i.e. the normalisation of the probability). A solution by Lagrangian multipliers yields

$$T_{ij}(\tau) = \frac{c_{ij}^{obs} + c_{ij}^{prior}}{\sum_i (c_{ij}^{obs} + c_{ij}^{prior})}.$$ 

The choice of the discretisation is an ambiguous problem with multiple solutions. We employed an approach which uses TICA coordinates and clustering into microstates for the discretisation. So these methods are shortly introduced below.
2 Theoretical Foundations

2.3 Time-structure based Independent Component Analysis (TICA)

TICA is a transformation method to determine the slowest uncorrelated motions as a linear combination of atomic coordinates [4]. TICA components form the natural choice as coordinates for Markov State Model estimation because they form the best input of linear coordinates to approximate the eigenvalues of the propagator [16].

In TICA a slow component is defined by having a maximal autocovariance at a given lag time $\tau_{TICA}$. The time lagged covariance matrix $C(\tau_{TICA})$ is defined by

$$C_{ij}(\tau_{TICA}) = \langle (x_i(t + \tau_{TICA}) - \bar{x}_i)(x_j(t) - \bar{x}_j) \rangle_t,$$

where $x_j$ is the j-th atomic coordinate, $\bar{x}_j$ the time average of the same coordinate, and $\langle . \rangle_t$ denotes the time-average.

TICA components are obtained by solving the following generalised eigenvalue problem [16]

$$C(\tau_{TICA}) \cdot U = C(0) \cdot U \cdot \Lambda.$$  \hspace{1cm} (2.8)

Here, $U$ denotes the matrix of the independent components (ICs), $\tau_{TICA}$ the lag time and $\Lambda$ designates the diagonal matrix of the eigenvalues.

The independent components and eigenvalues of the TICA problem can be used to obtain a kinetic map [17]. In this map the so-called kinetic distance $D_{kin}(\vec{x}, \vec{y}, \tau)$ is the applied metric. The kinetic distance of two points $\vec{x}$ and $\vec{y}$ is defined by the difference in probability to find them in the same position after a time $\tau$:

$$D^2_{kin}(\vec{x}, \vec{y}, \tau) = \|p(\vec{x} \rightarrow \vec{z}, \tau) - p(\vec{y} \rightarrow \vec{z}, \tau)\|_\mu$$

$$= \int_\Omega \frac{|p(\vec{x} \rightarrow \vec{z}, \tau) - p(\vec{y} \rightarrow \vec{z}, \tau)|}{\mu(\vec{z})} d\vec{z}.$$  

This metric is equal to the Euclidean metric if we use the eigenvectors as a basis.
set and rescale the eigenvectors by their corresponding eigenvalues

\[ \tilde{\psi}_i \rightarrow \lambda_i \cdot \tilde{\psi}_i. \]

Due to the fact that TICA approximates the eigenvectors and eigenvalues of the propagator the TICA transformation can be used to approximate the kinetic maps. Kinetic maps usually yield a convergence of the implied timescales at lower MSM lag times during the MSM estimation (see sec. 3.4 for more details). Since converging ITS are a necessary condition for Markovity, kinetic maps are a useful tool to improve MSM estimation and we applied them in this work [17].

2.4 Discretisation

Any discretisation means a loss of information because the ensemble density within a state \( S_i \) is assumed to be constant. So the true eigenfunction of the propagator \( \psi_i(\vec{x}) \) will be approximated by a discrete eigenfunction \( Q\psi_i(\vec{x}) \). As a consequence discretisation introduces a discretisation error \( \delta_i \).

\[ \delta_i = ||\psi_i(\vec{x}) - Q\psi_i(\vec{x})||_{\mu^{-1}}. \]

The extent of the error depends on the chosen discretisation. An example of different discretisations of a one-dimensional double well is shown in figure 2.4. In this example the naïve approach to assign one state to one well does not yield the smallest error, although a separation at \( x = 50 \) results in smaller errors than a separation at \( x = 40 \). Here, a finer discretisation leads to less error. Especially a fine discretisation of the transition region with six states in total between the two wells lowers the error more than a discretisation into ten equally large states.

The discretisation error on the eigenfunctions is important because it translates into errors of the eigenvalues and implied timescales. For these quantities the error vanishes under two conditions: a sufficiently fine discretisation or a long lag time \( \tau \). Since a long lag time \( \tau \) lowers our sampling of limited data, a fine discretisation, the microstates, was chosen in this work.
Figure 2.4: Different discretisations and discretisation errors for a 1D toy model. The partition lines (black, vertical) mark the borders of different states $S_i$. The discretisation error is denoted as $\delta_2$. The four discretisations are a partition on top of the barrier ($x = 50$), left of the barrier ($x = 40$), equal partition into 10 states ($x = 10, \ldots, 90$) and into six states ($x = 40, 45, 50, 55, 60$). Source: [3, Chap. 3].
3 Methods

The goal of this work was to derive MSMs for each protein of a test set of 592 proteins (see in sec. 3.5). For MSM estimation we applied a protocol which consists of three major steps:

The first step was to map the atomic coordinates onto TICA coordinates [1] and use kinetic maps which allowed us to cluster by kinetic distance in the next step [17]. Only a limited number $cd$ of the slow independent components was kept in this step for two reasons: the main reason is that the aim of this work is to compare motion patterns of the proteins. So the MSM estimation should follow the same protocol in all cases. By choosing the same $cd$ for all proteins we ensured that the subsequent discretisations are similarly fine. The second reason is to save computational costs by reducing the number of dimensions. The cutoff $cd$ and the TICA lag time $\tau_{TICA}$ are the two free parameters in this step.

The actual discretisation was conducted in this projected space. To achieve a similarly fine discretisation for all proteins we employed the $k$-means algorithm [18] with a the fixed number of states $k$ for all proteins.

In the last step the transition matrices were estimated via counting the transitions between the discrete states. The estimation was conducted by a Bayesian estimator. The Bayesian estimator was chosen because it is asymptotically unbiased and incorporates the effect of a prior. Furthermore, we enforced that the probability flux of the transition matrix is reversible in this step because it should be fulfilled in equilibrium MD simulation and ensures that the eigenvalues of the matrix are real values [3, p. 49ff].

This procedure is applied on a test set to obtain an MSM for each individual protein. The protocol and the parameters of each step are sketched in figure 3 and
3 Methods

implemented in PyEmma 2.0.2. The remainder of this chapter will introduce the applied methods and the test set in more detail.

<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICA</td>
<td>TICA lag-time $\tau_{\text{TICA}}$, cutoff dimension $cd$</td>
</tr>
<tr>
<td>k-means</td>
<td>number of clusters $k$</td>
</tr>
<tr>
<td>MSM estimation</td>
<td>MSM lag-time $\tau_{\text{MSM}}$</td>
</tr>
</tbody>
</table>

Figure 3.1: Sketch of the protocol (left) and the associated parameters (right). For origin and details of the parameters see sec. 3.1 - 3.3.

3.1 Time-structure based Independent Component Analysis (TICA)

To obtain TICA coordinates it is necessary to determine the time-lagged correlation matrix $C(\tau_{\text{TICA}})$. For the trajectory data this is done by a discretised version of equation (2.7)

$$c_{ij}(\tau_{\text{TICA}}) = \frac{1}{N - \tau_{\text{TICA}} - 1} \sum_{l=1}^{n-\tau_{\text{TICA}}} (x_i(l + \tau_{\text{TICA}}) - \bar{x}_i)(x_j(l) - \bar{x}_j).$$

Here $l$ is the index for the discrete time steps. For algebraic reasons $C(\tau_{\text{TICA}})$ needs to be symmetric. However, this condition is sometimes not fulfilled for the numerical obtained $C(\tau_{\text{TICA}})$. For this reason symmetry was enforced by averaging over $c_{ij}(\tau_{\text{TICA}})$ and $c_{ji}(\tau_{\text{TICA}})$

$$\bar{c}_{ij}(\tau_{\text{TICA}}) = \frac{1}{2} (c_{ij}(\tau_{\text{TICA}}) + c_{ji}(\tau_{\text{TICA}})).$$

This symmetric time-lagged correlation matrix $\bar{C}(\tau_{\text{TICA}})$ was used in the algorithm AMUSE algorithm to TICA eigenvalue problem (eq. (2.8)) [20].

From the obtained independent components only the $cd$ ICs with the largest eigenvalues were kept for the actual discretisation and the trajectory data was projected onto the kinetic maps of this subspace.
3.2 K-Means Clustering Algorithm

The discretisation into microstates was realised by the $k$-means clustering algorithm. This algorithm yields a Voronoi tessellation of the trajectory data into $k$ clusters. Thereby it optimises the average distance $\Xi$ of data to cluster centres.

$$\Xi = \arg \min_S \sum_{i=1}^k \sum_{\bar{x} \in S_i} \|\bar{x} - \bar{z}_i\|^2$$  \hspace{1cm} (3.1)

In this equation, $S$ is the set of $k$ states, $\bar{z}_i$ is the centre of the $i-th$ state and $\|\cdot\|^2$ the Euclidean distance.

We used Lloyd’s algorithm which consists of the following steps [21]:

1. Choose $k$ random points of the trajectory
2. Perform Voronoi tessellation of setspace
3. Calculate $\Xi = \sum_{i=1}^k \sum_{\bar{x} \in S_i} \|\bar{x} - \bar{z}_i\|^2$
4. Set $\bar{z}_i = \sum_{\bar{x} \in S_i} \bar{x} / N_i$
5. Repeat 2. - 4. until $\Xi$ is converged

The main parameter of this discretisation is the number of clusters $k$. A higher number of clusters lead to a finer discretisation.

One of the weaknesses of $k$-means is the fact that it tends to assign too many clusters to densely populated regions and underrepresents sparsely populated regions like transition regions between free energy minima. The reason for this behaviour is that densely populated regions contribute much to the average distance $\Xi$, whereas sparsely inhabited regions only contribute little. So a finer discretisation in the densely populated regions contributes more to the minimisation of $\Xi$ more than increase of the distance to the centre in the less populated regions contributes to increase $\Xi$. However, underrepresentation of outliers is less pronounced than in alternative methods like regular time clustering [19]. Therefore, $k$-means clustering is a good choice.
Recent studies suggest that MSM estimation is robust to the clustering method as long as there is sufficient trajectory data and the number of microstates is sufficiently large [22].

### 3.3 Estimation of the Transition Matrix

After obtaining a discretisation by the k-means algorithm, the transition matrix was estimated by a maximum probability estimator. To do so, it is necessary to choose a counting scheme, a prior and a MSM lag time. The two former ones will be set here, but the lag time $\tau_{\text{MSM}}$ will be discussed in the results section.

#### Count Matrices

With a given discretisation the count matrix of the transitions between states can be estimated in different schemes. The simplest approach would be starting at $t = 0$ and counting transitions by subsequently observe the discretisation state at multiples of $\tau_{\text{MSM}}$. However, this approach ignores intermediate data. So we apply a window-counting mode (see fig. 3.2 for both modes). This mode samples the trajectory at multiples of $\tau_{\text{MSM}}$, but does so for all multiples of the time resolution of the trajectory data $\Delta t$ and corrects for multiple counting

$$c_{ij}(\tau_{\text{MSM}}) = \left\lfloor \frac{(N - 1)/l}{N - l} \right\rfloor \sum_{k=0}^{N-l-1} \chi_i(\vec{x}(k \cdot \Delta t))\chi_j(\vec{x}((k + l) \cdot \Delta t)).$$  \hspace{1cm} (3.2)

Here $l$ is given by $l = \tau_{\text{MSM}}/\Delta t$ and $N$ is the number of time steps of length $\Delta t$ in the trajectory. This method has the advantage of extracting the maximum of the data. However, jumps may be counted multiple times, so they cannot be assumed to be statistically independent. Therefore, assuming they were would yield to low error estimations. Thus, no Bayesian errors should be extracted from this counting mode [3 chap. 4].

#### Detailed Balance and Prior

One problem for the estimation of transition matrices is that the estimators do necessarily fulfil the condition of detailed balance. Since there is no closed form for the maximum probability estimator which incorporates detailed balance, we applied an iterative algorithm of Prinz et al [22]. To ensure that the reversibility of the proba-
3.4 Implied Timescales to Test Markovity of Transition Matrices

To assess how well the transition matrix fulfils the Markov property we utilised implied timescales at different MSM lag times. The definition of the ITS of the

\[ c_{ij}^{\text{prior}} = \begin{cases} 
-1 & \text{for } i \geq j \\
0 & \text{else}
\end{cases} \]

It is optimised to obtain a correct solution while minimising the computational costs and was benchmarked against a test set of proteins. This makes it an advantageous choice.

Figure 3.2: Sampling at \( \tau \) and window-counting. For sampling at \( \tau \) only the data at the red marks will be used while the window-mode exploits all data. Source: [3, chap. 4].

A prior for this algorithm was developed by Trendelkamp et. al. [23]. The proposed prior there was used here. It is given by

\[ c_{ij}^{\text{prior}} = \begin{cases} 
-1 & \text{for } i \geq j \\
0 & \text{else}
\end{cases} \]

It is optimised to obtain a correct solution while minimising the computational costs and was benchmarked against a test set of proteins. This makes it an advantageous choice.

3.4 Implied Timescales to Test Markovity of Transition Matrices

To assess how well the transition matrix fulfils the Markov property we utilised implied timescales at different MSM lag times. The definition of the ITS of the
propagator implies that
\[ t_j = -\frac{\tau}{\ln(\lambda_j)} = -\frac{k\tau}{\ln(\lambda_k^j)} = t_j. \] (3.3)

In other words, implied timescales do not depend on the lag-time at which the MSM is estimated [24]. Since the transition matrix are supposed to approximate the transfer operator, equation (3.3) should also be fulfilled by the ITS of the estimated transition matrix.

This fact can be tested by estimating transition matrices at different MSM lag times \(\tau_{\text{MSM}}\) and comparing the implied timescales: as soon as the ITS as a function of \(\tau_{\text{MSM}}\) are converged the Markov property is fulfilled.

This fact was used in order to choose the MSM lag-times \(\tau_{\text{MSM}}\): the MSM lag time should be large enough to fulfil the Markov property, but apart from as short as possible.

The second criterion which can constructed from the ITS is that the first longest ITS is always underestimated [16]. This means, the larger the first timescale the better the MSM. Therefore, an MSM with converged and large timescales will fulfill the Markov property best.

### 3.5 Test Set

The test set for this project consists of molecular dynamics simulations for 592 proteins. They were chosen by hand from the protein data bank to model the full protein space as good as possible. A full list of the PDB codes can be found in the appendix.

The protein structures were checked by WHAT IF [25] and WHAG to correct geometric errors. The rotamer library in WHAT IF was also used to fill gaps and missing side chains into the structure. Symmetry relaxed crystal water molecules that contact the monomer in the asymmetric unit cell were used. If case alternate atoms existed, the most abundant atom was selected. If multiple atoms were equally abundant, the one with the label A was selected. For hydrogen bonding network the optimization protocol of Hooft et al was followed [26]. The same protocol was
applied to determine optimal protonation states and rotamer angles for Glb, Asn and His residues. If aromatic groups exhibited unphysical deviations from planarity they were corrected by moving them into a planar conformation.

Each protein has been simulated for the time of 1 μs with a time step of 4 fs and virtual sites by the software Gromacs 4.5 [27]. The force field in these simulations was AmberP99SB + ILDN with the TIP4P water model [28, 29]. The proteins were placed into a solvent shell of 1.1 nm. The water shell included sodium and chloride ions in a concentration of 150 mM. Energy minimisation for each protein was executed by steepest descent for 100 steps. Afterwards the solvent was equilibrated for 500 ps while heavy atoms were positionally restrained with a force constant of 1000 kJmol⁻¹nm⁻². The MD simulations were carried out in isobaric-isothermic ensemble (NPT) where they were temperature coupled by velocity rescaling (τ_T = 5 ps) and pressure coupled at 1 bar (Berendsen, τ_P = 1 ps) [30, 31]. Within a cutoff of 1 nm electrostatic and Lennard-Jones interactions were calculated directly and the neighbour list was updated every 10 steps. Long-range electrostatic interactions were computed by particle mesh Ewald with a spacing of 0.125 nm [32]. The coordinates were saved every 10 picoseconds.

To keep computations feasible only the Cα-atom positions of the proteins have been used to estimate a Markov Model.
4 Results & Discussion

The results are divided into two sections. The first section discusses how the TICA cutoff \( cd \), the TICA lag time \( \tau_{\text{TICA}} \), the number of microstates \( k \) and the MSM lag time \( \tau_{\text{MSM}} \) were chosen and shows observations of the TICA transformation, the microstate discretisation by k-means clustering and the Bayesian MSM estimation. The latter part deals with a comparison of the MSM for all 592 proteins by the 10 longest finite implied timescales.

4.1 Parameters for MSM Estimation

4.1.1 TICA Coordinates

The first step of the MSM procedure is TICA. Here we had to choose the cutoff dimension \( cd \) and the TICA lag time \( \tau_{\text{TICA}} \). To investigate the dynamics in TICA space, the projection of trajectories on the first two independent components for three proteins and two different lag times (\( \tau_{\text{TICA}} = 100 \text{ ps}, 1 \text{ ns} \)) are given in figure 4.1. The chosen proteins are Escherichia coli CheY (Protein Data Bank (PDB) [33]: 3CHY [34]), CspA, a major cold shock protein of Escherichia coli (PDB: 1MJC [35]) and repeats 16 and 17 of chicken brain alpha spectrin (PDB: 1CUN [36]). These three proteins have been chosen as representative examples for different cases of dynamics.

In figure 4.1 we observed the following for \( \tau_{\text{TICA}} = 100 \text{ ps} \): on the left (PDB: 3CHY) there are some deep islands of high density linked to a large region of rather low density. In the middle (PDB: 1MJC) there is a U-like structure. On the right (PDB: 1CUN) the density exhibits number of islands with few regions of low density. The U-like patterns of the middle structure were observed for many proteins. This structures resemble the behaviour of a random walk in many dimensions [37, 38]. Thus, these patterns indicate that there is a large random walk content in the
4 Results & Discussion

dynamics of many proteins. The shape in this projection often looks similar for different lag times but there are also differences. For example the U-like structure of 1MJC seems to be mirrored at the y-axis. The shape of the projection of 1CUN seems to be similar for both lag times, but 3CHY has a slightly different structure at $\tau_{\text{TICA}} = 1 \text{ ns}$.

The occurrence of the variety of different patterns in the projection may have two reasons: sampling and the fact that there are different proteins. Whether sampling is the main source was tested by deriving TICA coordinates for only the latter half of the trajectory. For the examples above this is shown in figure 4.1 (row 4). The comparison of row 3 and 4 in figure 4.1 reveals that the projection on the first two independent components can indeed show different patterns for only parts of the trajectories. The patterns are not complete, but only show parts of the patterns, often rotated or mirrored in space. For the CHeY protein the LH density only shows the left and the upper part of the whole trajectory for the density. Similar behaviour is visible at spectrin, where the left and the bottom part of the trajectory get reproduced. For CspA the patterns of the whole trajectory do not get reproduced. These examples illustrate that sampling of the trajectory plays a role for TICA coordinates. But still parts of the features could be seen in TICA coordinates for the whole or half of the trajectory. Thus, sampling and differences of the proteins both determine the shape of the density and the independent components together. These results match the expectations: if only a certain region of the free energy landscape is sampled and a rare transition did not take place, the slowest processes did not occur, so TICA cannot find them and the density looks different from sampling two different regions. Apart from, different proteins have different free energy landscapes. Therefore differences between proteins were expected.

Visual inspection of these projections does not provide a criterion for $\text{cd}$ nor a TICA lag-time $\tau_{\text{TICA}}$. Hence, a more systematic comparison of the coordinates is necessary to choose parameters for the TICA transformation.

To this end, we make use of the kinetic distance (see sec. 2.3) and analyse the cumulative variance fraction $\Gamma_{\lambda}(d)$. This quantity measures the fraction of the sum
Figure 4.1: Three example proteins. PDB code, structure (row 1), and heatmap of the density projected on the first two TICA components for $\tau_{\text{TICA}} = 100\text{ps}$ (row 2), $\tau_{\text{TICA}} = 1\text{ns}$ (row 3) and for $\tau_{\text{TICA}} = 1\text{ns}$ estimated from the latter half (LH) of the trajectory (row 4).
4 Results & Discussion

Figure 4.2: Cumulative variance fraction $\Gamma_\lambda(d)$ as a function of the number of eigenvectors $d$ for three different TICA lag times $\tau_{\text{TICA}}$ for all proteins.

The cumulative variance fraction for different TICA lag times for the whole data set is given in figure 4.2. For all lag times $\tau_{\text{TICA}}$ it is observable that most times incorporating a $\Gamma_\lambda(d)$ close to one requires many dimensions. So we can conclude that there are many proteins with many slow processes. However, to keep computations feasible choosing a low $c_d$ is necessary. Thus, the next step was to analyse the distribution of $\Gamma_\lambda(d)$ for small values of $d$.

A histogram of $\Gamma_\lambda(d)$ for $d = 10, 30, 50$ is shown in figure 4.3 and the corresponding means and standard deviations of $\Gamma_\lambda(d)$ are given in table 4.1. In these histograms one can recognise two trends: the first trend is that for higher $d$ we have a higher average $\Gamma_\lambda(d)$. This matches the expectations, since $\Gamma_\lambda(d)$ naturally is a monotonically increasing function. For $d = 10$ the mean of $\Gamma_\lambda(d)$ is always below 0.54. So choosing...
4.1 Parameters for MSM Estimation

Figure 4.3: Histograms of the cumulative variance fraction \( \Gamma_\lambda(d) \) for different \( d \).

<table>
<thead>
<tr>
<th>( d )</th>
<th>( \tau_{\text{TICA}} ) 100 ps</th>
<th>( \tau_{\text{TICA}} ) 1 ns</th>
<th>( \tau_{\text{TICA}} ) 10 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.24 ± 0.11</td>
<td>0.39 ± 0.15</td>
<td>0.54 ± 0.16</td>
</tr>
<tr>
<td>30</td>
<td>0.53 ± 0.16</td>
<td>0.72 ± 0.15</td>
<td>0.77 ± 0.13</td>
</tr>
<tr>
<td>50</td>
<td>0.69 ± 0.16</td>
<td>0.85 ± 0.11</td>
<td>0.85 ± 0.11</td>
</tr>
</tbody>
</table>

Table 4.1: Mean and standard deviation of \( \Gamma_\lambda(d) \) for different \( d \) and \( \tau_{\text{TICA}} \).

\( cd = 10 \) seems to be too low, whereas \( cd \leq 30 \) covers a way larger cumulative variance fraction. So both values would be reasonable choices. For the lag-time \( \tau_{\text{TICA}} \) the values 100 ps, 1 ns and 10 ns have been analysed. These have been chosen such that they cover the TICA lag time of usual examples like pentapeptide within one order of magnitude larger and smaller [19]. A lag-time of 1/10 ns cover more than 70% or 85% for a \( cd \) of 30 and 50, respectively. So both lag-times seem to be good parameters. Together this yields a restriction of the parameters to four different combinations \( (d = 30, 50 \text{ and } \tau_{\text{TICA}} = 1, 10 \text{ ns}) \). The final decision for a combination of parameters is obtained through the implied timescales of the transition matrix in sec. 4.1.3.

4.1.2 Discretisation into Microstates

In the second step of the protocol we employed the \( k \)-means algorithm with \( k = 100 \) to 1000 cluster centres. Similarly to the previous section, visual inspection of the density maps including the cluster centres provides insight in the distribution of the cluster centres. The position of the cluster centres for \( k = 100 \) and \( k = 250 \) for the density maps of the previous example are given in figure 4.4. As expected the \( k \)-means algorithm indeed tends to set many clusters to highly populated regions and thereby neglects or underrepresents regions of lower density (especially in
4 Results & Discussion

Figure 4.4: k-means clustering for $k = 100$ and $k = 250$ cluster centres for the examples above. White dots mark the cluster centres.

1MJC, $k = 100$). This effect is less pronounced for $k = 250$ than for $k = 100$. So more clusters appear to yield better results because a finer discretisation is expected to have less errors of the eigenvalues as pointed out in sec. 2.4. For 3CHY and 1CUN the concentration of clusters is not as strong as for 1MJC. So overall k-means discretises space relatively regularly.

However, the clustering is only an intermediate step and the goal is to have MSMs which are truly Markovian (see sec. 3.4). Therefore, we have to consider the transition matrices in order to assess the quality of our models. This was done in the next section.

4.1.3 Estimation of the Transition Matrix

The exploration of parameters for the whole test set of was mainly conducted on 10 randomly chosen proteins (see tab. 4.2) to save computation time because optimising all parameters at once required many runs.

The transition matrix was estimated on the largest set of states which is connected
4.1 Parameters for MSM Estimation

<table>
<thead>
<tr>
<th>PDB codes of the proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>3CHY 1ENH 1SHF 1UBQ 1MJC 1A6N 1ARB 1CUN 1WAS 1EP0</td>
</tr>
</tbody>
</table>

Table 4.2: Proteins used for more detailed scan of the parameter space.

Figure 4.5: Active set of cluster centres for the whole trajectory (WH, top) and only the latter half (LF, bottom). White dots mark the cluster centres.

reversibly, i.e. backwards and forwards transitions are observed (see sec. 3.3). To assess how much of the trajectory is covered in this active set, the density maps with cluster centres of the active set for the three example proteins are shown in figure 4.5.

In this figure the whole trajectory density maps are more completely covered with centres of the active set than those of the latter half of the trajectory. Often (3CHY and 1MJC) only one island of high density is covered. This matches the expectations because transitions between different transformations are rare events and therefore a trajectory length of 1 µs has a larger probability of forwards and backwards transition than only the latter half of the trajectory. Since a transition in both is necessary to be included in the reversible transition matrix estimation, transitions between the different conformations will not be seen in the transition matrix for half of the trajectory. Due to the fact, that the ten longest implied timescales were chosen to compare protein motion patterns, longer trajectories are expected to yield longer...
implied timescales and thus produce better MSMs. The reason why the eigenvalues (and subsequently the implied timescales) were chosen to describe protein dynamics is explained next.

Many clusters assigned to the minima. Transitions between close states in the same free energy minimum usually only correspond to minimal movement and do not incorporate transitions between different conformations. Furthermore the discretisation by the k-means algorithm is not deterministic. So the transition rates between those individual microstates does not yield the biologically relevant information, namely the transition between different conformations. The conformational changes and rates are encoded in the eigenvectors and values of the transition matrix. An example of eigenvalues for different parameters is given in figure 4.6.

These eigenvalues do not possess a clear gap as in the introductory example in the theory section. This matches the expectations for proteins because unlike the simple 1D example proteins possess a complex energy landscape with movements on all timescales.
An unexpected observation are negative eigenvalues. They mainly occur for high numbers of cluster like $k = 1000$ or long $\tau_{\text{MSM}}$. The occurrence of negative eigenvalues may indicate multiple jumps during a MSM lag time $\tau_{\text{MSM}}$ (see appendix for derivations). Therefore, negative eigenvalue should were minimised by fixing $k = 100$, i.e. choosing a rather coarse discretisation. This option was favoured over reducing the MSM lag time because $\tau_{\text{MSM}}$ needs to be long enough that the transition matrix should still satisfy the Markov property. This was tested next.

### Timescales to test Markovity

The implied timescales of transition matrices at different MSM lag times $\tau_{\text{MSM}}$ were used to test whether Markovity and the largest ITS can assess how well the eigenvector of the propagator of the protein is approximated (see sec. 3.4). These two criteria have been applied to determine values for the remaining free parameters ($\tau_{\text{TICA}}, cd, \tau_{\text{MSM}}$).

To illustrate different possibilities of the behaviour of the ITS for different cases the ten longest ITS are plotted as a function of the MSM lag time $\tau_{\text{MSM}}$ for our previous example proteins are shown in figure 4.7.

There are different kinds of behaviour. 3CHY shows a convergence of the ITS between 3 and 5 ns but decreasing timescales afterwards. A slightly different behaviour was observed for 1MJC, where timescales converge quickly but e.g. the second largest timescale (green) surges again after $\tau_{\text{MSM}} = 3$ ns. The third protein, spectrin, exhibits different behaviour. Here, the ITS seem to slow in growth (but do
not converge) up to $\tau_{MSM} = 3$ ns and afterwards they surge again.

All of the behaviour after $\tau_{MSM} = 3/5$ ns is probably a result of sampling errors which become worse for longer MSM lag times \cite{39}. Since the total simulation time per protein is $1 \mu s$, an MSM lag time of several nanoseconds limits sampling to several transitions between different free energy minima. Therefore, we chose an MSM lag time $\tau_{MSM} = 2$ ns as a compromise between convergence of the ITS and the sampling issues of longer MSM lag times.

In order to determine parameters for the TICA transformation the slowest implied timescales were compared for different values of $\tau_{TICA}$ and $cd$. Here both criteria for the longest ITS were included: convergence at low MSM lag times and large implied timescales. This is shown in figures 4.8 and 4.9 for the $cd$ and $\tau_{TICA}$, respectively.

For the cutoff dimension $cd$ it is visible, that increasing the number of independent components usually means an increase in the implied timescales (3CHY & 1MJC). Also it seems to converge better (3CHY & 1MJC). Nevertheless For 1CUN a larger $cd$ does not yield higher timescales or better convergence. Because this case does not show convergence for either cutoff the choice of $cd$ is grounded on the first two cases and $cd = 50$ was chosen for the test set of all proteins.

To determine a value $\tau_{TICA}$ we proceeded similarly: the longest implied timescales as a function of the MSM lag time for different $\tau_{TICA}$ for the three example proteins is given in figure 4.9. Here, a lag time of $\tau_{TICA} = 1$ ns exhibits the largest timescales.
4.1 Parameters for MSM Estimation

![Figure 4.9: Longest implied timescales for the three example proteins for different TICA lag times τ_{TICA}.](image)

Figure 4.9: Longest implied timescales for the three example proteins for different TICA lag times $\tau_{\text{TICA}}$.

and convergence (3CHY & 1MJC). Nevertheless, 1CUN does not show convergent behaviour for any timescale. Even though $\tau_{\text{TICA}} = 10 \text{ ns}$ yields a slightly higher longest timescale than $\tau_{\text{TICA}} = 1 \text{ ns}$. Since $\tau_{\text{TICA}} = 1 \text{ ns}$ shows faster convergence and higher longest ITS for the other two cases, the TICA lag time was fixed to $\tau_{\text{TICA}} = 1 \text{ ns}$ for the computations of MSMs for the whole test set of proteins.

Determining $\tau_{\text{TICA}}$ completes this section. Values for all parameters have been derived. Before comparing the transition matrices for the whole test set the reliability of the protocol will be discussed in the next step.

4.1.4 Discussion of the Protocol

The parameters of table 4.3 are a compromise between optimal parameters for individual proteins. A finer parameter scan might reveal a more fine-tuned set of parameters but a slightly different parameters would probably not change the overall results of the whole test set much because implied timescales are supposed to be independent of the MSM lag time (when it converged) and the slowest TICA independent components would stay similar because the trajectory stays the same. So fine-tuning of $c_d$, $\tau_{\text{TICA}}$, $k$ and $\tau_{\text{MSM}}$ would probably only lead to minor improvements. Since this work aims to find a fingerprint of protein motion patterns rather than an exact description of the dynamics, the chosen parameters can be expected to yield MSMs good enough for the comparison of the different proteins.

For the applied methods in the protocol TICA transformation (incl. kinetic maps) and a Bayesian estimator are reasonable choices, as pointed out in the methods
4 Results & Discussion

<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICA</td>
<td>TICA lag-time $\tau_{\text{TICA}}$</td>
<td>1 ns</td>
</tr>
<tr>
<td></td>
<td>cutoff dimension $cd$</td>
<td>50</td>
</tr>
<tr>
<td>k-means</td>
<td>number of clusters $k$</td>
<td>100</td>
</tr>
<tr>
<td>MSM estimation</td>
<td>MSM lag-time $\tau_{\text{MSM}}$</td>
<td>2 ns</td>
</tr>
</tbody>
</table>

Table 4.3: Derived values for the parameters of the protocol for MSM estimation on the whole test set.

The largest source of error is the discretisation into microstates. Here plenty of other algorithms are available [3, Chap. 2]. Although $k$-means is known to steer a middle course between over- and underrepresentation of outliers, it is a rather simple approach and more sophisticated methods are being developed [40]. So reviewing the effect of further clustering algorithms can possibly yield convergence of the implied timescales at smaller MSM lag times. Thus, the discretisation would be the starting point to refine the protocol.

4.2 Comparison of All Proteins by Implied Timescales

The aim of this work is to compare motion patterns of different proteins. Since the eigenvalues encode the dynamics between different conformations, we used the space of the ten largest implied timescales of each protein as a space for protein dynamics. So each protein is a point in a ten dimensional vector space:

$$\vec{t}'(i) = (t'_1(i), t'_2(i), \ldots, t'_9(i), t'_{10}(i))^T.$$ (4.2)

Here, $t'_j(i)$ is the j-th longest implied timescale of the i-th protein. These timescales have to be normalised because we compare the largest timescales and subsequently smaller ones of each protein to each other. Otherwise, the longest implied timescales naturally has a larger variance than the second largest timescale. To this end, the timescales were normalised in the following way:

$$\vec{t}_j(i) \rightarrow \bar{t}_j(i), \quad \text{with} \quad t_j(i) = \frac{t'_j(i) - \bar{t}_j}{t'_j}.$$ (4.3)

Where $\bar{t}_j$ is the average of the j-th timescale for all proteins. In addition to normalisation, the proteins had to be filtered because sometimes the active set only
4.2 Comparison of All Proteins by Implied Timescales

Figure 4.10: Histogram of the number of states in the active set (left) and longest timescale as a function of the number of states in the active set.

comprised a very low number of states, so only a small fraction of the dynamics is covered in the transition matrix. This is shown in figure 4.10 (left) where a histogram of the size of the active set for all proteins is given. Most proteins have a size of 100, however also many proteins have a smaller active set. The size of the active set is important because it influences the implied timescales observed. This is shown in figure 4.10 (right) where the longest implied timescale is given as a function of the size of the active set. A clear correlation between these two quantities is visible. This result can be expected because a small active set means that transitions between different conformations only occurred in one direction. So the active set only incorporates dynamics within one conformation and ignores the slow process of conformational changes. To correct for this effect, a minimal active set of 50 was used for further analysis. This reduces the number of proteins for the actual comparison to 338.

For the actual comparison of the proteins we searched for the vectors with the largest correlation, so a principal component analysis (PCA) was carried out. The results for the eigenvalues of the PCA are shown in figure 4.11 (left).

It is observable, that the eigenvalues decay quickly and that the first eigenvalue is dominant. The second and the third eigenvalue are much smaller but still no-
noticeably larger than the rest of the eigenvalues. So the first three eigenvectors are most crucial for the motion patterns. To see which timescales are relevant in these vectors, the weight factors of the ten timescales for the first three eigenvectors are shown in figure 4.11 (right).

For all of the eigenvectors the longest timescale plays a role since it is weighted between 0.3 and 0.5. The second longest timescale has the largest weight of the first PCA eigenvector (eigenvalue $\lambda_1 = 4.45$). This is surprising because an intuitive guess would be that the longest timescale would be considered most important. This is because it corresponds to a transition between states which are separated by the largest energy barrier of the system. But in the dominant first eigenvector the absolute value of the weight factor of the longest implied timescale is only slightly larger than the absolute value of the third timescale (weight $-0.4$). The second eigenvector (eigenvalue: $\lambda_2 = 0.3$) has large weight on the third and fourth timescale. The third eigenvector (eigenvalue: $\lambda_3 = 0.06$) has comparably large weight on the third to fifth timescale and most weight on the fifth timescale. However, due to the smaller eigenvalues, the third eigenvector is noticeably less crucial for the dynamics than the first two eigenvectors. An interesting observation is that none of the first three eigenvectors has large weight on the faster timescales $t_6$ to $t_{10}$. This matches the
4.2 Comparison of All Proteins by Implied Timescales

expectation that the most important parts of the dynamics are encoded in the long timescales. Since the first two PCA eigenvectors span the two dimensional projection with the most variance, the next step is to analyse them more precisely.

In order to find whether different classes of proteins are distinguishable by the implied timescales, a scatter plot of the ITS of the proteins projected on the first two PCA eigenvectors is given in figure 4.12.

In this figure most of the proteins are close to the point of origin and there is a positive correlation between the eigenvectors. The space from point of origin to (5,5) is filled continuously with decreasing density towards (5,5). This means, that no distinct clusters could be found. So using the first ten implied timescales as a space for protein motion patterns did not yield a metric to discriminate different classes of proteins.

However, the comparison of the ten longest implied timescales by a principal component analysis and visual inspection was only a first approach to compare the
4 Results & Discussion

motion patterns. So it forms a starting point for further comparison of the implied timescales.
5 Conclusions & Outlook

Scanning the parameter space and testing implied timescales allowed us to find optimal values for the parameters of TICA transformation, discretisation and subsequent transition matrix estimation.

Comparing the ten longest implied timescales of the test set of proteins revealed that most of the variance in space is located in the five longest implied timescales. However, no clusters of different classes of proteins could be found by a principal component analysis.

Sampling showed to be an important issue in Markov State Model estimation because only regions which are connected through forwards and backwards transitions could be included in the analysis. This biased the results because proteins with high energy barriers between different conformations could appear to exhibit only shorter timescales than a protein with lower barriers between conformations where transitions forwards and backwards between different conformations occur. This problem is inherent in the trajectories, so

The next logical step would be to analyse the space of the first ten implied timescales closer (similar to the dynasome project) and apply e.g. a Graph of Mutual Adjacencies. Another idea would be to find a different a different quantity to compare the Markov State Models like apply clustering the microstates into macrostates by e.g. Perron-Cluster Cluster Analysis and then comparing matrix elements the transition matrix between macrostates.
6 Appendix

6.1 Negative Eigenvalues - 2 State Model

To achieve some understanding of the negative eigenvalues, we considered a simple toy model. This toy model is a symmetric 2-state model with states $A$ and $B$ with transition probabilities $p_{A \rightarrow B}(\tau) = p(\tau)$ and $p_{B \rightarrow A}(\tau) = p(\tau)$. So the transition matrix $T_{A \leftrightarrow B}(\tau)$ is given in equation (6.1).

$$
\begin{bmatrix}
A \\
B
\end{bmatrix}
\rightarrow
\begin{bmatrix}
B \\
A
\end{bmatrix}
with
T_{A \leftrightarrow B}(\tau) = \begin{bmatrix}
1 - p(\tau) & p(\tau) \\
p(\tau) & 1 - p(\tau)
\end{bmatrix}
\tag{6.1}
$$

Diagonising this matrix $T_{A \leftrightarrow B}(\tau)$ reveals the eigenvalues

$$
\lambda_1(\tau) = 1 \quad \text{and} \quad \lambda_2(\tau) = 1 - 2p(\tau).
\tag{6.2}
$$

Here we can see that there is a negative eigenvalue if $p(\tau) > 1/2$, i.e. crossing is more probable than remaining in the state. Physically this indicates that during the lag time $\tau$ jumping might often occur multiple times.
<table>
<thead>
<tr>
<th>PDB codes of all proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>1WIT 3CHY 1ENH 1SHF 1UBQ 1MJC 1AN6 1ARB 1CUN 1WAS 1EP0 2PTH</td>
</tr>
<tr>
<td>1QAU 1EBD 4WBC 1DYN 1JAM 2GOS 2GIW 1HFC 1BP5 1HH8 1BS2 1IXA</td>
</tr>
<tr>
<td>1HER 1JD1 1BFD 1EQK 11AS 1WE8 1Z9D 1YSB 1YEL 1XX3 1XHK</td>
</tr>
<tr>
<td>1XQO 1XXR 1WHN 1WIN 1YB3 1WK1 1WFY 1XSZ 1WJW 1WHB 1WJG 1WJJ</td>
</tr>
<tr>
<td>1WFT 1WJ5 1XQ1 1WB7 1AD 1FK8 1FZW 2HNP 1E4V 1ESJ 1LYY 3GRS</td>
</tr>
<tr>
<td>1HEE 1G5B 1FZT 1E99 2FPE 1HYU 2Glt 1E6Y 1HPG 1GL6 1FHPQ 1HII</td>
</tr>
<tr>
<td>1HMF 1J22 1G6N 1HPL 16A 1QHN 1G5M 1IEC 11JU 1JP1 1CP 1GG7</td>
</tr>
<tr>
<td>1FVL 1G03 1JH3 1E01 1IS 1G0Q 1HT 11G 1FO 1EGLD 1G5D</td>
</tr>
<tr>
<td>1FSZ 1E09 1SHH 1JR2 61H4 1HJZ 3GAR 1JAW 1LE 1YEY 1WE4 2HGF</td>
</tr>
<tr>
<td>1FVA 1H5P 1IS9 1EPU 1E97 1HAI 1JAA 1IVF 1F3X 1JRH 1H2T</td>
</tr>
<tr>
<td>1LYH 1EUV 1JHF 1HFG 1HUS 3GCC 1H6A 1H4 2GKY 1JW3 1HMU 1FTT</td>
</tr>
<tr>
<td>1EQ1 2G03 1LJJ 1ETH 1G41 1G7M 1HUH 1HM8 1EQ6 1J33 2EZN 1GZT</td>
</tr>
<tr>
<td>1GE1 1G0S 1D01 1H0 1IS1 1G0N 1HPL 1J26 1G61 1HSP 1EMW 1U3 1GXL</td>
</tr>
<tr>
<td>1IQO 1J86 1H02 1H0F 1G6 1F7W 11G0 1HT 1E8M 1HP2 1J0B</td>
</tr>
<tr>
<td>1E15 1ZSG 1GWW 1D03 1H0 1IC 1D0F 1H0 1IV3 1F0A 1FO 1J4W</td>
</tr>
<tr>
<td>1LSD 1L5L 2RTC 1U9A 2TG1 1FTB 1T6G 1SBS 1L0H 1TUL 1V4E 1H8V</td>
</tr>
<tr>
<td>4MAT 1SS0 1TYA 1M6B 1K99 1TPG 1S04 1M0P 1KN8 1MZG 1LID 1KSV</td>
</tr>
<tr>
<td>1KEO 1L51 1LTU 1UNN 1K07 1LBY 1KVW 1USG 1K0F 1UN8 1K3C 2L8</td>
</tr>
<tr>
<td>2MOB 1H76 1HRA 1L6 16K 1M1 1M1 1M0 1KXW 1IS2 1VDH 1ML</td>
</tr>
<tr>
<td>1L3V 1L0 1MWP 1T3B 1SQR 1IK0 1K3W 1LN8 1GCU 1MVG 1MP1 1LRE</td>
</tr>
<tr>
<td>1K91 1MLL 1U02 1V9G 1Q84 1QLP 10TP 1VR9 1HU 1PY 1OBL 1QK9</td>
</tr>
<tr>
<td>2NUS 2NEF 1OP4 1QH 1GQ 1P1 1RLH 1Q7 1IK1 1Q60 1GFP 1P1</td>
</tr>
<tr>
<td>1N6Z 1PV5 1OQM 1N1J 1PO2 1BG 1ODG 1QZM 1QZ 1O99 1P35 1N5</td>
</tr>
<tr>
<td>1H5K 2J6 1HP4 1HD0 1XQ 1ZQ 1N9 1NY9 1ROK 1NK 1Q1V</td>
</tr>
<tr>
<td>1QC7 1PUJ 1IN6 1N7 1IK6 1A82 1QAZ 1B6 1INT 1E8G 1R4V 1BSG 1D6T</td>
</tr>
<tr>
<td>1CEQ 1CEQ 1C08 1UCX 1P0 1LS 1D0G 1D0F 1A6H 1D0X</td>
</tr>
<tr>
<td>1DB 1HA2 3PMG 1PNO 1OED 1H10 1FAD 1H3B 1EVL 1UNK 1DX 1C0</td>
</tr>
<tr>
<td>2PG1 3PRO 1WWR 1QCV 1BW3 1DVG 1F17 1A9 1CD5 1DC0 1B2 1DTW</td>
</tr>
<tr>
<td>1LDL 1CDZ 1BK 1BG2 3RUB 1HYR 1HEA 1QPM 1CRN 10AG 1DFT 1C2</td>
</tr>
<tr>
<td>1AKO 1CK 1AKZ 1EDO 1HOF 1B0L 1CZM 1HM0 1K2Y 1BY1 1F33 1B78</td>
</tr>
<tr>
<td>1BFW 1A6S 1WE 1WU2 1O8P 1NGN 1HAN 1FA 1F68 1BPX 1NJ 2FUS</td>
</tr>
<tr>
<td>1DHN 1HWC 1CIP 1YCO 1VS 1AG 1NOG 1CZ 1K 1S0 1D1D 1BM0</td>
</tr>
<tr>
<td>1EAM 1AA3 1RL6 1C3G 1TSF 1RZ 1UK 1C1Y 1B1U 1HA 1H1N 1HIC</td>
</tr>
<tr>
<td>1B8X 1B6Z 1A3A 1RHX 1D05 1H2T 1DZ0 1E00 1CMI 1WGR 1C4 2BES</td>
</tr>
<tr>
<td>1AIX 1D1L 1YX 1RO 1B04 1HQL 1XJH 1AP 1ASM 1X3 1XH 1INY</td>
</tr>
<tr>
<td>1YZ2 1CE 1QT 1NYO 1W4H 1U8 1QQV 1PDO 1QQH 1RQS 1MG 1D</td>
</tr>
<tr>
<td>1MS 1X8 2AMC 1BLG 1RLL 1K2H 1W00 1GMO 1JU 1XWM 1K8 1V5G</td>
</tr>
<tr>
<td>1BSH 1C99 1U1T 1BW3 1FF 1BGW 1YGE 13T 1PV 1IQ 1GYY 1ZDF</td>
</tr>
<tr>
<td>1CSZ 1HM 1B4Y 1B7S 1WNB 1U1V 1HE 1H1G 1WE 1J2 1E5 1ADN</td>
</tr>
<tr>
<td>1C04 1D0Q 1D7A 1DFD 1I0V 1APF 1D2 1DT 1YX 1KPT 1WHR</td>
</tr>
<tr>
<td>1WHX 1XHJ 1WHZ 1WHQ 1UCO 1B2V 1NY9 1YDL 1WPS 2FPM 1Q2W 1Q2F</td>
</tr>
<tr>
<td>1X8R 1X8S 1MK0 1WOT 1YLQ 1JBI 1KAH 1NN 1J3E 1N1U 1VCC</td>
</tr>
<tr>
<td>1WPR 1QHK 2HBB 1C57 1WIK 1AS 1GO 1RZW 1RO 1AUA 1AQ 1WFP</td>
</tr>
<tr>
<td>1WFM 1X7F 1C3 1HEE 1YE 1P 1WHM 1WFP 1XK 1CQ3 1BY 2B97</td>
</tr>
<tr>
<td>1NP 1HGF 1ABV 1ADT 1D3 1WHC 1WIX 1RQ6 1OYW 1ST 1DTP 1YOZ</td>
</tr>
<tr>
<td>1GW 1U84 1PUZ 1UG9 1PV0 1GO 1HP 1ZSB 1NHR 1WIJ 1CIK 1X9B</td>
</tr>
<tr>
<td>1SO 1BAA 1DOT 1X6 1XVI 1CBF 1D0 1DK 1YS 1W6X 1MT 1AA7</td>
</tr>
<tr>
<td>1ST 1G71 1WPB 1AMM 1ALC 2AY 2V0 1NKR 5CFA 1FTU 4AE 1AST</td>
</tr>
<tr>
<td>1BP1 1CNV 1B1D 1GD 1AH 1AV 1BGQ 1WE 1SH 1CE 2BA 1KE</td>
</tr>
<tr>
<td>1AAJ 1GSO 1DDE</td>
</tr>
</tbody>
</table>
Bibliography


Bibliography


[38] Andreas Volkhardt. Tried by max linke (mpi biophysics, frankfurt, germany), 2016.

Acknowledgements

I thank Helmut Grubmüller for giving me the chance to work on this project. Andreas Volkhardt for many discussions and support in technical as well as conceptual problem issues. Furthermore I want to thank everybody in the Department for Theoretical and Computational Biophysics at the MPI for Biophysical Chemistry for its great working atmosphere and for help with various problems and challenges.
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.

Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.


(Daniel Quetschlich)