MAX-PLANCK-INSTITUT FÜR
BIOPHYSIKALISCHE CHEMIE
KARL-FRIEDRICH-BONHOEFFER-INSTITUT
GÖTTINGEN

Master's Thesis

Comparing Protein Dynamics using Markov State Models

Vergleichende Analyse der Dynamik von Proteinen mit Markov-Modellen

Author	Nicolai Kozlowski
	nicolai.kozlowski@stud.uni-goettingen.de
Supervisor	Helmut Grubmüller

Contents

1 Introduction **3**

2 Theory **7**
2.1 Dynamics Fingerprints 7

3 Methods **9**
3.1 Molecular Dynamics Simulations 9
3.2 Principal Component Analysis 12
3.3 Time-lagged Independent Component Analysis 12
3.4 K-means Clustering 13
3.5 Markov State Models 14
3.6 Scaled Sliding Window Counting 16
3.7 Transition Probabilities Estimation 17
3.8 Robust Perron Cluster Analysis 17
3.9 MSM Estimation and Fingerprint Extraction 18
3.10 Preceding Approach 19
3.11 Variational Autoencoder 21
3.12 Fingerprint Analysis 23
3.12.1 Distance-based Analysis 23
3.12.2 Conventional Clustering Measures 24
3.12.3 Nearest Neighbor Based Analysis 25
3.13 Bootstrapping 25

4 Results and Discussion **25**
4.1 tICA Projections 26
4.2 MSM Fingerprints 30
4.3 Dynasome 1 Fingerprints 32
4.4 Comparison of Fingerprints 35
4.4.1 Distance-based Analysis 35
4.4.2 Clustering Measures 37

5 Outlook **39**
5.1 Sampling 40
5.2 MSM Construction 41
5.3 Function-Specific Information 42

References **43**

6 Appendix **51**

1 Introduction

Proteins are crucial building blocks of life. They conduct many essential tasks in living organisms, haemoglobin binds and transports oxygen, myosin and actin contract muscles and ion channels enable neural signal processing [1]. Despite their importance for many essential processes in life, we do not know the function of most proteins. The majority of function annotations available in large protein databases are predictions and experimentally detected annotations only make up $< 1\,\%$ in UniProt (protein sequence database) [2] and $40\,\%$ in the Protein Data Bank (PDB, contains 3D-Structures) [3]. Such predictions are error-prone, and wrong annotations possibly mislead researchers to waste time and resources. Therefore, increasing the reliability of protein function predictions is a crucial interest of current research [2, 4].

The first developed and most common function prediction algorithms are based on the protein primary sequence [5]. This sequence is unique to the protein and is encoded in the DNA. Comparing protein sequences, similarities and common motifs were discovered that expanded our understanding of proteins [6]. These similarities (next to other features) are exploited by homology modelling algorithms to predict a protein function given its sequence [2, 4, 7]. The main idea behind these prediction algorithms is that proteins with high sequence similarity are likely to also conduct the same function. They therefore only work well if there is an experimentally studied protein with high sequence similarity [4].

In solution, many proteins fold into globular structures. These 3D-structures can also be predicted by homology modelling or determined experimentally by X-ray crystallography, nuclear magnetic resonance spectroscopy or 3D-electron microscopy [8, 9, 10, 11], for example. Comparing 3D-structures, again similarities and common motifs were found (secondary structure elements like α-helices and β-sheets) expanding our knowledge on proteins. These can be utilized for function prediction on the premise that similar 3D-structure (e.g. binding pockets shaped alike) implies similar function. Including experimentally determined 3D-structures as inputs enhanced function prediction, particularly for low primary sequence similarity cases [12].

However, protein sequence and structure based algorithms still do not provide sufficient accuracy in function prediction, leaving room for improvement [3]. Protein function prediction is a multi-label classification task with ≈ 10000 labels in total (called Gene Ontology molecular function terms [13, 14]), and multiple labels per protein [15]. In the most recent benchmarking competition for primary sequence based protein function prediction, CAFA3, the top performing method GOLabeler [15] reached an F_1-score of around 0,6 [4]. The F_1-score is a measure for accuracy in classification tasks, it is calculated as the harmonic mean of precision ($\frac{\Sigma \text{ true positives}}{\Sigma \text{ predicted positives}}$) and recall ($\frac{\Sigma \text{ true positives}}{\Sigma \text{ ground truth positives}}$). The structure based function annotation method FINDSITE reached a score of $F_1 = 0{,}63$ on

a different protein benchmark set [16]. These and other current methods mainly exploit similarities to proteins with known functions and the detection of enzymatic binding pockets [15, 12, 17]. Therefore, their coverage is limited and predicting non-enzymatic functions of proteins without well-studied homologs remains challenging. To make more accurate function predictions for a broader variety of proteins without relying on primary sequence or 3D-structure similarity, we aim to include protein dynamics in the function prediction process.

Intramolecular protein motions are driven by thermal noise (or, in some cases ATP/GTP hydrolysis) and are determined by their specific free energy landscape. The most relevant motions are conformational changes on timescales ranging from μs to ms [18]. They can be calculated using molecular dynamics (MD) simulations if a 3D-structure to start the simulation from is available. In MD simulations, the Newtonian equations of motion are integrated for a system of molecules and atoms, e.g. a protein and water surrounding it, calculating interaction forces via empirical force fields. Thereby, information about the dynamics of the protein is captured in a trajectory, describing how the positions of all its atoms change with time.

We can produce MD trajectories at a rate of roughly 1 μs per GPU per week due to recent advances in hard- and software [19]. This rate makes generating sufficiently long trajectories ($\geq 1\,\mu$s) for a large number of proteins feasible using large computer clusters. The computational cost is still high though, which is why protein dynamics have not been studied in a comparative way to the same extend as sequences or 3D-structures so far. However, we expect the connection between protein dynamics and function to be strong, as observations from experimental and computational setups imply that some functions are conducted by protein motion and can not be conducted by rigid molecules [20, 21]. This connection, as schematically illustrated in Fig. 1, is the background of this work.

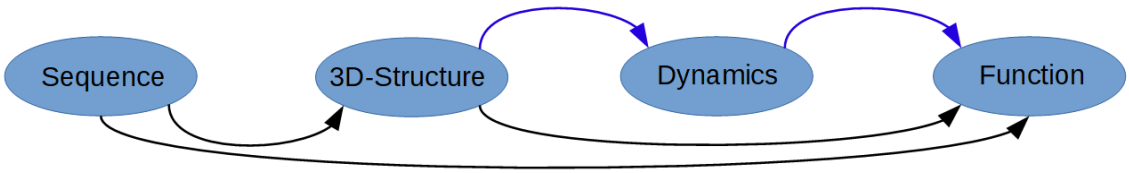


Figure 1: Sketch of connections between protein sequence, 3D-structure, dynamics and function. The connections linking structure with dynamics and dynamics with function (illustrated by blue arrows) have not been studied thoroughly yet within the framework of automated function prediction. We expect more accurate function prediction by exploiting these connections, because many protein functions are conducted by motion.

MD simulations output high-dimensional time series data that are hard to interpret and hard to compare between different proteins. In order to exploit the connection between dynamics and function, it is therefore necessary to map the trajectory onto a low-dimensional representation of the dynamics. This mapping should capture the characteristic kinetic behaviour of a protein and be automated and universally applicable to all proteins. We call it a *dynamics fingerprint* for that reason. The idea of comparing proteins by their dynamics fingerprints was first proposed by Hensen et al. in 2011 [22]. They named the space of all protein motions *dynasome* and also demonstrated how to use it for function prediction, assuming that proteins with similar dynamics fingerprints are likely to conduct the same function. In this thesis, we will follow up on their work, and get novel insights into the dynasome.

The dynamics fingerprint Hensen et al. proposed was captured from 100 ns long MD trajectories of 112 proteins. These are too short to cover all relevant motions, but the computational effort of simulating these trajectories was at the limit of feasibility in 2011. The fingerprint consists of 34 kinetic observables, which were extracted directly from the MD trajectories. These include, for example, the mean and standard deviation of root-mean square fluctuations and solvent-accessible surface areas (for the whole list of observables, see Ch. 3.10). The choice of these observables is arbitrary, with an infinite number of possible other inclusions, making it practically impossible to find the best set. Another drawback of this method is that some of the observables contain information about the secondary structure composition of the protein. This information is helpful for function prediction, but comes at the price of mixing information on structure and dynamics in the fingerprint. This mixing makes it difficult to attribute the fingerprint's informative value to either structure or dynamics, which is undesirable when we are trying to understand the diversity, specificity and predictive power of protein dynamics specifically. [22]

Another shortcoming of the previous work on the dynasome is that, for every protein, only one trajectory was considered and therefore only one fingerprint generated. Now for MD simulations, independently calculated trajectories of the same protein are not identical. Instead, they are different realizations of the same underlying random process, which is governed by the protein free energy landscape. Therefore, for any analysis of MD trajectories, it is crucial to examine to what extend the results are reproducible with other, independent trajectories. In the context of protein dynamics fingerprints this consideration is particularly meaningful: fingerprints captured from MD simulations contain protein-specific information, describing the random process, and trajectory-specific information, describing just one realization. Because a protein is labelled with the same function terms at any time (although it might not perform all its functions within every trajectory), we are interested in protein-specific information to make reliable function predictions.

To improve on these deficiencies, in this work, we investigate dynamics fingerprints regarding their protein specificity — i.e. reproducibility with respect to different trajectories. We developed a novel approach to capture dynamics fingerprint using Markov State Models (MSMs) and examine, whether it captures more protein-specific dynamics information than the previous approach.

MSMs are discrete, kinetic models which assume a system is in one of several states at any given time and can change to another state within a certain time frame. For proteins, these states correspond to different structural arrangements (e.g. open or closed binding pocket) called conformations. Most proteins only take a limited number of conformations, typically remaining in roughly the same conformation for at least several nanoseconds before they change to another one. MSMs describe this metastable behaviour by assigning a particular state to every commonly visited conformation and a transition probability to every possible transition between the states. The matrix formed by these transition probabilities fully characterizes the MSM. MSMs have proven to be a powerful tool to describe protein dynamics and are widely used for this purpose [18, 23, 24, 25, 26, 27].

In a next step, we extracted observables from the MSM to form a dynamics fingerprint. Observables suitable for that must be meaningful properties of the MSM and invariant to permuting states, drastically limiting the number of possible choices to properties of the eigendecomposition of the *transition probability matrix*. Hence the choice of observables is less arbitrary compared to the preceding work [22]. We calculated *implied timescales* and *eigenvector entropies* to form a fingerprint.

We generated $3 \times 1\,\mu$s MD trajectories each for a large set of 200 proteins. These are an order of magnitude longer than those used in previous work and hence capture protein dynamics on longer timescales. Also, there are multiple trajectories per protein, enabling us to investigate the protein specificity of fingerprints. From every trajectory we captured dynamics fingerprints using our MSM-based approach and the previous approach by Hensen et al. [22]. We also constructed a third set of fingerprints, which consists of all the observables chosen by Hensen et al. but the ones which carry structural information.

To examine the protein specificity of a set of fingerprints, we calculated euclidean distances between all pairs of fingerprints. We then calculated the average logarithmic distance between fingerprints captured from two trajectories of the same protein and the average logarithmic distance between fingerprints captured from two trajectories of different proteins. The quotient of these averages is a measure for the protein specificity of a set of fingerprints.

We found that the MSM fingerprints captured a similar amount of protein-specific information as the previous approach. If the latter is deprived of structural information,

however, it captures significantly less protein-specific information. We therefore conclude that indeed more dynamics-based protein-specific information is captured by our novel MSM fingerprints in contrast to the previous approach. There is still room to further improve the protein specificity of dynamics fingerprints though, for example through more sampling or optimizing parameters for MSM construction. Utilizing dynamics fingerprints for protein function prediction seems to be out of reach at the current stage.

2 Theory

2.1 Dynamics Fingerprints

The dynamics fingerprint of a protein is a d-dimensional vector $\mathbf{v} \in V \subset \mathbb{R}^d$ in the vector space V we call *fingerprint space*. This fingerprint space is unique to the method extracting the fingerprint, meaning two fingerprints extracted by different methods are not comparable with each other. Within V, in contrast, different fingerprints can be compared to identify similarities between the underlying dynamics, using for example euclidean distances as a dissimilarity measure [22].

We assume that several contributions to the dynamics fingerprint

$$\mathbf{v} \approx \boldsymbol{\mu}_{\text{func}} + \boldsymbol{\mu}_{\text{prot}} + \boldsymbol{\delta}_{\text{traj}} + \boldsymbol{\delta}_{\text{meth}} \tag{1}$$

play an important role. We divide them into informative terms, denoted by $\boldsymbol{\mu}$, and noise terms, denoted by $\boldsymbol{\delta}$. We assume these different terms to be independent and additive here to establish a simple formalization of fingerprint contributions and promote an intuitive understanding of their interplay. We do not claim this approach to be rigorous, and a more sophisticated formalization dropping these assumptions might be developed in future.

The function-specific term $\boldsymbol{\mu}_{\text{func}}$ is the same for all proteins sharing a certain function. We suggest this term to be a relevant contribution because it has been shown that many protein functions are coupled to a specific motion in experiments and simulations [20, 21]. In preceding work, a function-specific contribution to the fingerprint was suggested and confirmed, but it was not large enough to endow the fingerprint space with a clustered structure [22]. If $\boldsymbol{\mu}_{\text{func}}$ was the dominating term in Eq. (1) (i.e. $\|\boldsymbol{\mu}_{\text{func}}\| \gg \|\boldsymbol{\mu}_{\text{prot}}\| + \|\boldsymbol{\delta}_{\text{traj}}\| + \|\boldsymbol{\delta}_{\text{meth}}\|$), dynamics fingerprints of different proteins would form clusters in fingerprint space. However, it is not necessarily the case that such clusters even exist. Still, the term $\boldsymbol{\mu}_{\text{func}}$ is critical for function prediction based on dynamics fingerprints, but it remains elusive so far.

The term $\boldsymbol{\mu}_{\text{prot}}$ is protein specific. Because the dynamics in MD simulations are governed by the free-energy landscape of the simulated protein, we expect $\boldsymbol{\mu}_{\text{prot}}$ to be a major contribution.

The trajectory-specific contribution $\boldsymbol{\delta}_{\text{traj}}$ arises from the stochastic nature of MD simulations: Two independent, unbiased MD simulations of the same protein starting from the same conformation can sample entirely different regions of the free-energy landscape (apart from the initial conformation). For infinitely long and ergodic trajectories, every part of the free-energy landscape is visited infinitely often, and $\boldsymbol{\delta}_{\text{traj}}$ vanishes. Because $\boldsymbol{\delta}_{\text{traj}}$ is closely related to limited sampling, we expect it to decrease with increasing trajectory length and to increase with system size (i.e. primary sequence length).

The term $\boldsymbol{\delta}_{\text{meth}}$ describes method-specific noise. This noise is caused by the stochastic nature of the methods used to extract a fingerprint from an MD simulation, e.g. random initialization of cluster centers in k-means clustering (Ch. 3.4). If all extraction methods used are deterministic, $\boldsymbol{\delta}_{\text{meth}}$ vanishes. Its magnitude and behaviour with respect to involved parameters like trajectory length strongly depend on the methods used.

When considering multiple trajectories of the same protein, their fingerprints are scattered around a mean given by $\boldsymbol{\mu}_{\text{func}}$ and $\boldsymbol{\mu}_{\text{prot}}$ with a spread given by $\boldsymbol{\delta}_{\text{traj}}$ and $\boldsymbol{\delta}_{\text{meth}}$. For the most part of this work, we ignore whether noise is trajectory or method specific. Also, because neither our fingerprints, nor the fingerprints in preceding work form distinct function clusters, we assume $\|\boldsymbol{\mu}_{\text{func}}\| \ll \|\boldsymbol{\mu}_{\text{prot}}\|$. Taking these approximations into account, equation (1) simplifies to

$$\mathbf{v} \approx \boldsymbol{\mu}_{\text{prot}} + \boldsymbol{\delta}. \tag{2}$$

This equation suggests that fingerprints extracted from multiple trajectories of the same protein scatter around $\boldsymbol{\mu}_{\text{prot}}$ with a spread $\boldsymbol{\delta}$. The balance of these two contributions can be understood as a signal-to-noise ratio α, where the protein-specific information, manifested as $\boldsymbol{\mu}_{\text{prot}}$, is the signal we are interested in.

Because α and the magnitude of the fingerprint $\|\mathbf{v}\|$ might vastly differ between proteins, the overall cluster structure in fingerprint space can be complex, as illustrated in Fig. 2. Figure 2a shows a sketch of a fingerprints space where α is large. Fingerprints of the same protein form distinct non-overlapping clusters, enabling us to uniquely associate areas of fingerprint space with individual proteins. For small values of α, a more complex structure is expected, similar to the one sketched in Fig. 2b. In such a case, it is not obvious if any and how much protein-specific information was extracted with these fingerprints.

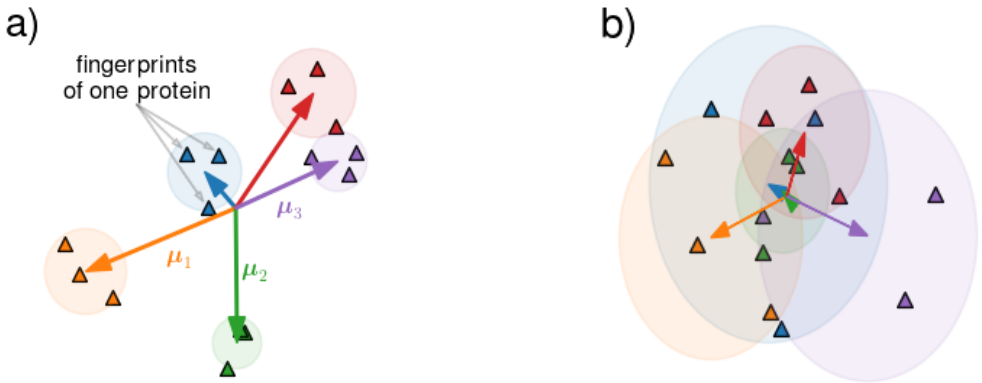


Figure 2: Sketches of dynamics fingerprint spaces. a) A well-clustered fingerprint space with high signal-to-noise ratio α. Different colors indicate different proteins, each occupying a unique part of fingerprint space with a mean value $\boldsymbol{\mu}$. **b)** Similar fingerprints with lower signal-to-noise ratio α and different fingerprint magnitude for each protein form a less structured fingerprint space with large overlap between areas occupied by different proteins.

To determine the amount of protein-specific information, pairwise distances between fingerprints are calculated and divided into two groups: Distances between fingerprints of the same protein and distances between fingerprints of different proteins. If distances between fingerprints of the same protein are smaller than distances between fingerprints of different proteins, on average, protein-specific information was captured with these fingerprints. The factor by which they are smaller is a measure for the amount of protein-specific information captured. This analysis is described in more detail in Ch. 3.12.

3 Methods

3.1 Molecular Dynamics Simulations

MD simulations were used in this work to generate trajectories capturing the internal motion of molecules, to later compare them.

The motions of all atoms of a molecule can be explicitly calculated by solving the Schroedinger-equation for all atom cores and electrons, but doing so for larger molecules like proteins is computationally not feasible. The small integration time step and the large number of floating point operations required make the calculation too slow to reach simulation times of microseconds that would cover large protein motions. To overcome this hurdle, some approximations are made to reduce the problem to solving

the Newtonian equation of motion for the atom trunks within fast-computable potentials.

Foremost, the Born-Oppenheimer approximation is applied. It exploits that electrons move much faster than atomic nuclei. From the electrons' perspective, the nuclei move so slow that their position can be assumed to be fixed, and for the nuclei, the electrons move so fast that the forces they exert can be averaged [28]. In the context of MD simulations, only the motion of the nuclei is of interest and hence electrons are treated implicitly.

There are three relevant forces on the molecular scale to be considered:

- The Coulomb force acts between any two charged particles, like the atomic cores considered in MD. The charge assigned to an atomic core is determined by the atom type and its chemical environment. The force exerted by a charge q_2 on a charge q_1 at a distance r is $F_{\mathrm{C}}(r) = \frac{1}{4\pi\epsilon_0}\frac{q_1 q_2}{r^2}$ [29] with ϵ_0 being the vacuum permittivity. The number of computer operations required for explicitly calculating coulomb forces between N atomic cores scales as N^2. To avoid this large number of computer operations, the Particle Mesh Ewald (PME) method [30] is used.

- Van-der-Waals forces are exerted between every pair of atoms. They arise from fluctuations in the electromagnetic fields of atoms [31]. These forces are calculated from a Lennard-Jones-(12,6) potential $V_{\mathrm{LJ}}(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6}$, where C_{12} and C_6 are constants specific to the involved particles [32]. The repulsive term also includes forces exerted due to Pauli repulsion, which are — strictly speaking — not Van-der-Waals forces, but also have to be considered in MD simulations. Again, the required computer operations scale as N^2, but because this interaction is short-ranged, it is ignored for atom pairs more than 1 nm apart.

- Chemical bonds are the source of a variety of bond forces. These forces are functions of the distance between a bonded pair of atoms, the bond angle between a pair of bonds that share an atom, or dihedral bond angles. For bond distances, bond angles or improper dihedral angles, these forces are approximated by harmonic potentials $V(x) = -k(x - x_0)^2$ with a spring constant k depending on the variable type (bond distance, bond angle or dihedral angle), the local topology of the bond network and the involved atom types. For proper dihedral angles, a cosine expansion potential $v(\phi) = \sum_{m=1}^{M} k_m(1 + \cos(m\phi - \phi_0))$ is used [33].

The sum of all choices made on how to calculate these forces and which parameters to use is called a *force field* $\boldsymbol{F}$. Inserting the force field into the Newtonian equation of motion yields

$$\frac{\mathrm{d}^2\mathbf{x}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i(\mathbf{x})}{m_i}, \tag{3}$$

with $\mathbf{x}$ denoting a $N \times 3$-dimensional matrix containing the three cartesian coordinates of all N atoms, $\mathbf{x}_i$ denoting its i-th row vector — i.e. the three cartesian coordinates of atom i — and m_i denoting the atomic mass of atom i. This approximated equation of motion for macromolecules is numerically integrated to generate a trajectory.

For this study, three $1\,\mu$s MD trajectories for each of 200 solvated proteins were generated to capture their dynamics by Marco Dalla-Sega and Carsten Kutzner, (former) members of our group. The proteins were selected by Marco Dalla-Sega and Tim Meyer from two published protein lists containing suitable candidates [22, 34] based on two criteria: First, the proteins must be globular and should be small, as for large and/or disordered proteins, we do not expect to sufficiently cover their whole range of dynamics with $1\,\mu$s trajectories. Second, different function classes should be covered by several representatives each, so that function-specific contributions can be distinguished from protein-specific contributions and examined separately. The chosen proteins are listed in the appendix in A. (Helmut Grubmüller, personal communication, 2020)

For the simulations used in this study the GROMACS 4.5 software package [35] with Amber ff99SB-ILDN force field [33] and TIP4P-Ew water model [36] was used. Starting structures were taken from the PBD [37] entries listed in A (Appendix). Solvent and ions (Na^+ and Cl^-) were added, establishing a salt concentration of $0{,}15\,\mathrm{mol\,l^{-1}}$ and neutralizing the overall system charge for charged proteins. Energy minimization was performed using GROMACS steepest descent at a time step of 1 fs until convergence to machine precision (single precision, $\leq 5 \cdot 10^4$ steps). Then equilibration was performed for $0,5 + 1$ ns (NVT+NPT) at an integration time step of 2 fs followed by the actual production run with a length of $1\,\mu$s (NPT) at an integration time step of 4 fs, all using periodic boundary conditions. A velocity rescaling thermostat [38] was applied at $T = 300\,\mathrm{K}$ with temperature coupling constant $\tau_T = 0{,}1$ ps and isotropic pressure coupling was applied at $p = 1$ bar with pressure coupling constant $\tau_p = 1$ ps using Berendsen pressure coupling [39] during equilibration and Parrinello-Rahman pressure coupling [40] in the production run. All bond lengths were constraint, using the SETTLE algorithm [41] for the solvent and LINCS [42] for the solute, with a LINCS order of 4 during energy minimization and equilibration and LINCS order of 6 in the production run. Van-der-Waals forces were ignored for distances > 1 nm and Coulomb forces were calculated using PME [30] with a real-space cutoff of 1 nm, PME order of 4 and Fourier-grid spacing of 1,2 Å. In the production run coordinates were saved every 10 ps, resulting in a trajectory of 10^5 frames.

The starting structure of the production run might be very improbable in the proteins equilibrium distribution, even after energy minimization and the equilibration runs, hence the first 20 ns of every trajectory were discarded as an additional equilibration phase [22].

Next to internal (i.e. intramolecular) protein motions, the trajectories contain external motions, namely center of mass motion and rotation of the whole molecule. These motions are diffusion processes which are determined by the protein shape and size [43], which is information we prefer not to have in our dynamics fingerprints for the reasons mentioned in Ch. 1. To remove external motions from the trajectories, for every frame, the proteins were centered in the simulation box and fitted to the respective starting structures using the GROMACS tool trjconv [35].

3.2 Principal Component Analysis

Principal Component Analysis (PCA) [44] is a common analysis method for multidimensional data. It was used in this work as a dimension reduction method for sets of dynamics fingerprints. To obtain principal components (PCs) of a set of fingerprints $\mathbf{X}$, its covariance matrix [45]

$$\mathbf{C} = (\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T, \tag{4}$$

where $\overline{\mathbf{X}}$ denotes the mean of $\mathbf{X}$, was computed and diagonalized. The right eigenvectors of $\mathbf{C}$ are the PCs, and its eigenvalues $\lambda_{1..d}$ are the variances of the corresponding PCs. The fingerprints were subsequently projected onto the two highest variance PCs by computing the dot product of the fingerprints with those PCs. The combined space spanned by the two highest variance PCs is assumed to be a relevant 2D-projection of fingerprint space and therefore enables us to visualize it.

PCA was also used to compute several of the fingerprint observables in previous work (Ch. 3.10) by Hensen et al. from the MD trajectories [22].

3.3 Time-lagged Independent Component Analysis

Time-lagged Independent Component Analysis (tICA) is an analysis method for time series data that identifies slow collective motions. It finds uncorrelated collective coordinates and maximizes their respective autocorrelations at a lag time τ [46, 47]. TICA was used in this work to reduce the dimensionality of MD trajectories. The slow collective motions found by tICA are likely to capture metastable behaviour, which is a key assumption of MSMs, as mentioned in Ch. 1. Hence tICA is a particularly well-suited method to find collective coordinates in MD trajectories to later build MSMs with [48, 47].

To perform tICA on a trajectory $\mathbf{X}_{1..T}$, two overlapping subsets of that trajectory were considered: a time-lagged subset $\mathbf{X}^\tau = \mathbf{X}_{1+\tau..T}$, and a not-time-lagged subset

$\mathbf{X}^0 = \mathbf{X}_{1..T-\tau}$. Using these subsets, the covariance and time-lagged covariance matrices [48]

$$\mathbf{C}_{00} = (\mathbf{X}^0 - \overline{\mathbf{X}^0})(\mathbf{X}^0 - \overline{\mathbf{X}^0})^T, \tag{5}$$
$$\mathbf{C}_{0\tau} = (\mathbf{X}^0 - \overline{\mathbf{X}^0})(\mathbf{X}^\tau - \overline{\mathbf{X}^\tau})^T, \tag{6}$$
$$\mathbf{C}_{\tau 0} = (\mathbf{X}^\tau - \overline{\mathbf{X}^\tau})(\mathbf{X}^0 - \overline{\mathbf{X}^0})^T \text{ and} \tag{7}$$
$$\mathbf{C}_{\tau\tau} = (\mathbf{X}^\tau - \overline{\mathbf{X}^\tau})(\mathbf{X}^\tau - \overline{\mathbf{X}^\tau})^T \tag{8}$$

were computed. Here, $\overline{\mathbf{X}^0}$ denotes the time average of $\mathbf{X}^0$. Using these matrices, the generalized eigenvalue problem [48]

$$(\mathbf{C}_{00} + \mathbf{C}_{\tau\tau})\mathbf{r}_i = (\mathbf{C}_{0\tau} + \mathbf{C}_{\tau 0})\lambda_i \mathbf{r}_i \tag{9}$$

was formulated. The corresponding eigenvectors $\mathbf{r}_i$ — called time-lagged independent components (tICs) — and eigenvalues λ_i were obtained by solving it.

For subsequent analysis, the eigenvectors were scaled with the corresponding eigenvalues [49] and sorted by them. Starting with the highest eigenvalue tIC, a number of tICs sufficient to contain 95 % of the trajectory variance was kept and the rest was discarded.

The lag time τ is typically chosen specifically for the analysed trajectory [49, 50, 51]. However, in this work, our goal was not to find the perfect MSM construction pipeline for every trajectory, but to find a pipeline that yields comparable MSMs and that is reasonable for every trajectory. By reasonable we mean that the pipeline should be able to identify metastable protein conformations and assign MSM states to them, if such conformations are contained in the trajectory. To make the pipeline yield comparable results, we strived to fix all parameters within it. Hence we fixed the tICA lag time τ for all trajectories to the same value, and chose that value equal to the MSM lag time t_l. Our literature research showed that MSM lag times in the range of 1–10 ns are appropriate to analyse MD trajectories of small proteins [23, 52, 53, 54] and accordingly $t_l = \tau = 3$ ns was chosen.

We used the software package PyEMMA [55] to perform tICAs for this study.

3.4 K-means Clustering

K-means clustering was used to decompose MD trajectories into discrete states to build MSMs on, because it is shown to be among the best choices for that purpose [56] and commonly used in MSM construction pipelines [23, 25, 53, 52, 57].

Initially, the k-means algorithm randomly places k cluster centers. Their position is then iteratively optimized by minimizing the squared distance of every data point to its nearest cluster center. [58]

The number of cluster centers k is an input parameter usually chosen based on the complexity of the configration space of the investigated system and the amount of sampling at hand [23]. For our MSM construction pipeline we fixed it at $k = 200$, consistent with choices made in several recently published studies of protein kinetics [52, 53, 57]. Our choice is also consistent with a criterion for sufficient sampling described by Pande et al.: The number of observed transitions, estimated by T/t_l, with T denoting trajectory length and t_l denoting MSM lag time, should match the order of magnitude of the number of possible transitions in the MSM, which scales as $k \ln k$ [23]. In our case, $T/t_l = 1\,\mu\text{s}/3\,\text{ns} = 333,\bar{3}$ and $k \ln k \approx 1000$, so the number of transitions observed in a trajectory is smaller by a factor of 3 compared to the number of possible transitions, matching the order of magnitude. Note that T/t_l is actually a lower limit to the number of uncorrelated transitions in a trajectory and determining this number for trajectories is an unresolved current problem [59].

3.5 Markov State Models

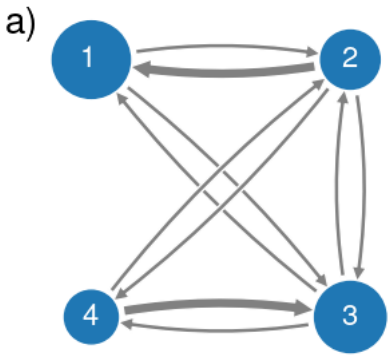
As mentioned in Ch. 1, Markov State Models (MSMs) were used to approximate protein kinetics in this study. An example of an MSM is shown in Fig. 3. To understand the observables we extracted from an MSM to form a dynamics fingerprint and why we chose them, we first need to elaborate a little on MSMs and transition probability matrices (Fig. 3b).

The state of an MSM at any point in time t is defined by a probability distribution $\mathbf{p}(t)$, a vector that, for every state, contains a probability to be in that state (e.g. if a 4-state system is in state 1 at time t, $\mathbf{p}(t) = (1, 0, 0, 0)$). The transition probability matrix $\mathbf{P}$ is a propagator, forwarding $\mathbf{p}(t)$ in time by a lag time t_l, so

$$\mathbf{p}(t + t_l) = \mathbf{P} \cdot \mathbf{p}(t). \tag{10}$$

The parameter t_l was chosen to be $t_l = 3\,\text{ns}$ for the reasons mentioned in Ch. 3.3. Because the transition probability matrix entries k_{ab} are probabilities for the system's state to change from a to b within a time frame t_l, $\mathbf{P}$ has to be row stochastic and its diagonal elements are given by $k_{aa} = 1 - \sum_{b \neq a} k_{ab}$. Therefore, the model is fully determined by the transition probabilities k_{ab} with $a \neq b$.

Some of the most meaningful properties of an MSM are obtained by spectral decomposition of $\mathbf{P}$. The largest eigenvalue of a row stochastic matrix is always $\lambda_1 = 1$



b)

$$\mathbf{P} = \begin{pmatrix} 0.8 & 0.1 & 0.1 & 0 \\ 0.2 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.1 & 0.2 & 0.7 \end{pmatrix}$$

Figure 3: Illustrative Markov State Model. a) Circles and arrows represent states (protein conformations) and possible transitions between them, respectively. Their sizes encode the states´ equilibrium population and the transition probability. **b)** The corresponding transition probability matrix $\mathbf{P}$ describing the example MSM. The entries are probabilities for given transitions within a time frame t_l.

[18]. Therefore, the corresponding eigenvector $\mathbf{v}_1$ remains unchanged when applying the transition probability matrix $\mathbf{P}$, i.e.

$$\mathbf{P}\mathbf{v}_1 = \lambda_1 \mathbf{v}_1 = 1 \cdot \mathbf{v}_1 = \mathbf{v}_1 = \boldsymbol{\pi}. \tag{11}$$

$\mathbf{v}_1$ is called the MSM stationary vector or equilibrium distribution and is denoted by $\boldsymbol{\pi}$. If $\mathbf{P}$ describes a system in equilibrium, there should be no probability fluxes in the equilibrium state, i.e. [60]

$$\pi_a k_{ab} = \pi_b k_{ba} \quad \forall a \neq b, \tag{12}$$

a property called *detailed balance*, should be fulfilled. Because we assume proteins and solvents to be in equilibrium in our simulations, we enforce *detailed balance* when estimating MSMs.

All other eigenvalues λ_i are smaller than 1 and their corresponding eigenvectors represent equilibration processes. Using these eigenvalues we computed *implied timescales*

$$t_i = \frac{-t_l}{\ln \lambda_i}, \tag{13}$$

which describe how fast the respective processes reach equilibrium. These timescales give insight into the rate at which a protein changes its conformation along several pathways in configuration space and therefore capture relevant information on the protein kinetics.

They are also invariant to permutation of states, hence we chose them as observables for our MSM dynamics fingerprint.

We also included observables describing how different states contribute to the equilibration processes and to the equilibrium distribution in the fingerprint, because we considered that to be relevant information as well. To this end, *eigenvector entropies*

$$S_i = \sum_j \mathrm{v}_{i,j} \ln \mathrm{v}_{i,j} \tag{14}$$

were computed. Note that the sum is over the entries of $\mathbf{v}_i$, so one entropy contains information on one eigenvector. These are also invariant to permutation of states.

These two types of observables, *implied timescales* and *eigenvector entropies* made up the entire MSM fingerprint. Note that there are other observables that can be calculated from an MSM and meet our two criteria of being kinetically relevant and invariant to permutation of states we did not include, e.g. the entries of the equilibrium distribution $\boldsymbol{\pi}$ sorted by size.

3.6 Scaled Sliding Window Counting

Scaled sliding window counting was used to extract transition counts from discrete trajectories. Such counts are needed to estimate transition probabilities for the observed transitions.

To obtain transition counts c_{ab} from a discrete trajectory $X_{1..T}$, overlapping windows of size t_l were considered, namely $X_{1..t_l}, X_{2..t_l+1}, X_{3..t_l+2}$ and so on. For every one of these windows, one transition $a \to b$ was counted, where a and b are the states visited in the first and last frame of the window, respectively. However, these counts are not uncorrelated [59] and to account for that the obtained counts are divided by t_l, so [61]

$$c_{ab} = \frac{1}{t_l} \sum_{t \in [1, T-t_l]} \delta_{a,X_t} \delta_{b,X_{t+t_l}}, \tag{15}$$

using the Kronecker-delta $\delta_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$ The transition counts are arranged in a count matrix $\mathbf{C}$.

Due to the reweighting factor $1/t_l$, the total number of transition counts extracted from a trajectory is $\sum_{a,b} c_{ab} = T/t_l$. As mentioned in Ch. 3.4, this estimate is just a lower

limit for the actual number of uncorrelated transitions [59]. However, the transition probabilities estimated from the counts are unaffected by the reweighting factor for the estimation scheme used in this work [62].

3.7 Transition Probabilities Estimation

Transition probabilities k_{ab} between different states $a \neq b$ of a discrete trajectory were estimated from its transition counts $\mathbf{C}$. The probability density $p(k_{ab}|\mathbf{C})$, that a transition probability $k \in [k_{ab}, k_{ab} + \Delta k]$ gave rise to the observed counts $\mathbf{C}$, was calculated using a bayesian approach [18]

$$p(k_{ab}|\mathbf{C}) \propto p(\mathbf{C}|k_{ab})p(k_{ab}) \tag{16}$$

from the likelihood [18]

$$p(\mathbf{C}|k_{ab}) \propto k_{ab}^{c_{ab}} \exp\left(-k_{ab} c_{aa} t_l\right) \tag{17}$$

and the prior $p(k_{ab})$.

The prior was assumed to be uniform or $p(k_{ab}) \propto 1/k_{ab}$ in most, recent studies [27, 57, 62, 63, 64, 65, 66]. Both appeared to be reasonable and there is, to our knowledge, no systematic study regarding the choice of prior for transition probabilities. We assume that the choice of the prior does not affect the results of this study much though, because either prior affects all MSMs in a similar way and our later analyses are based on differences between MSMs and not on the actual numerical values of MSM properties. We used a uniform prior, resulting in transition probabilities being estimated by maximizing the likelihood (Eq. (17)), because this estimate is easier to calculate compared to the posterior distribution $p(k_{ab}|\mathbf{C})$ when using $p(k_{ab}) \propto 1/k_{ab}$.

During the maximum likelihood estimation *detailed balance* was enforced, as mentioned in Ch. 3.5. We used our own implementation of the iterative estimation algorithm described by Trendelkamp-Schroer et al. in 2015 [62].

3.8 Robust Perron Cluster Analysis

Robust Perron Cluster Analysis (PCCA+) is a fuzzy spectral clustering algorithm we used for coarse-graining MSMs, going from $k = 200$ states to $n = 10$ states. The algorithm is specifically tailored to coarse-grain MSM states using their transition probability matrix $\mathbf{P}_{k \times k}$ to maximize metastability of the state clusters [67].

In order to find the most metastable clusters, the eigenvectors $\mathbf{v}_i$ of $\mathbf{P}$ corresponding to the n largest eigenvalues λ_i were computed from the eigenvalue equation

$$\mathbf{P}\mathbf{v}_i = \lambda_i \mathbf{v}_i. \tag{18}$$

These eigenvectors were arranged in an eigenvector matrix $\mathbf{V}_{k\times n} = [\mathbf{v}_1, ..., \mathbf{v}_n]$ and normalized so that [67]

$$\mathbf{V}^T \cdot \mathrm{diag}(\boldsymbol{\pi})^2 \cdot \mathbf{V} = \mathbf{1}, \tag{19}$$

with $\mathrm{diag}(\boldsymbol{\pi})$ denoting the diagonal matrix with the MSM equilibrium distribution $\boldsymbol{\pi}$ on its diagonal and $\mathbf{1}$ denoting the identity matrix. A positive and row stochastic membership matrix $\mathbf{X}_{k\times n}$ was computed from $\mathbf{V}$ using a linear, invertible transformation $\mathbf{A}_{n\times n}$ by [67]

$$\mathbf{X} = \mathbf{V}\mathbf{A}. \tag{20}$$

There are many possible choices for $\mathbf{A}$ respecting the mentioned constraints, so an initial guess for $\mathbf{A}$ was drawn randomly and subsequently iteratively adjusted to maximize the objective function [67]

$$f_{obj} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{a_{ij}^2}{a_{1i}} \tag{21}$$

with a_{ij} denoting entries of $\mathbf{A}$. It is shown that using this objective function, metastability of the coarse-grained MSM is maximized [67].

After convergence, the entries x_{ij} of the membership matrix $\mathbf{X}_{k\times n}$ were interpreted as probabilities that state i belongs to cluster j [67]. Every state was assigned to the cluster it most likely belongs to. In some cases, all states were assigned to less than $n = 10$ clusters and some clusters were empty. In all those cases at least 7 clusters were populated though.

We used the PCCA+ implementation provided by the python package msmtools.

3.9 MSM Estimation and Fingerprint Extraction

A set of 200 proteins (listed in A) was selected for this work, and for every protein, three unbiased MD simulations were performed, as described in Ch. 3.1.

The trajectories were further processed applying tICA (Ch. 3.3) to reduce their dimensionality. Then, k-means clustering was performed as described in Ch. 3.4. The trajectory was discretized by assigning every frame to the nearest cluster center. Transitions were counted in the discretized trajectory using a scaled sliding window approach (Ch. 3.6). Using these counts, transition probabilities were estimated according to Ch. 3.7, fitting a 200-state MSM (Ch. 3.5) to the trajectory.

PCCA+ (Ch. 3.8) was applied to the MSM, merging the 200 states into 10 clusters. In some cases, PCCA+ assigned states to less than 10, but at least 7 clusters. The assignment were used to coarse-grain the discrete trajectory. Again, scaled sliding window counting (Ch. 3.6) was used to obtain transition counts for the coarse-grained trajectory, and using these, a coarse-grained MSM with $n = 7$–10 states was estimated according to Ch. 3.7.

To serve as fingerprint variables, we extracted the 6 largest timescales and the corresponding eigenvector entropies as well as the eigenvector entropy of the equilibrium distribution from the coarse-grained MSMs, as described in Ch. 3.5.

Because the obtained timescales were distributed among multiple orders of magnitude, we used their logarithm (base e) as fingerprint variables. The 13 fingerprint variables were then standardized to have mean $\mu = 0$ and variance $\sigma^2 = 1$ within our data set applying a linear transformation.

3.10 Preceding Approach

Preceding work on the dynasome (Dynasome 1) considered dynamics fingerprints $\mathbf{Y}$ composed of 34 handpicked observables (see Tab. 1), calculated from 100 ns MD trajectories [22]. These observables are strongly correlated with primary sequence length L. In order to limit their analysis to information on dynamics, the authors removed this sequence length information from their data. To this aim, the observables were decorrelated by subtracting an appropriate fit function. For all observables, except № 34, $y(L) = a + b \cdot L^c$ with fixed exponents c was used. Observable № 34 (S_{RMSF}) was decorrelated using $y(L) = a + log(L)$. The distribution of decorrelated observables was then standardized to have mean $\mu = 0$ and variance $\sigma^2 = 1$. [22]

For this study, a set of dynamics fingerprints (Dynasome 1 fingerprints) was calculated in the same way from our 1 μs MD trajectories. A slight adaptation was made in the decorrelation step: to better capture correlations within our data set, c was considered a free parameter for our fits.

Calculation of observables related with ruggedness of the energy landscape, № 21–23, relied on obtaining a ruggedness profile with a characteristic minimum [22]. However, the ruggedness profiles we calculated for our trajectories did not all have one characteristic minimum, but some had multiple. For such cases, it was unclear how the ruggedness profiles should be processed, and hence we excluded them from the Dynasome 1 fingerprints, which therefore consisted of 31 observables in this study.

Table 1: Observables forming the dynamics fingerprint in preceding work (Dynasome 1) [22]

Index	Symbol	Description
1..5	$\lambda_1..\lambda_5$	PCA eigenvalues 1-5
6	m^{λ}	Slope of the middle third of the PCA eigenvalue spectrum
7	χ^2_{λ}	R^2 value of the fit to the PCA eigenvalue spectrum
8..12	$\cos_1 .. \cos_5$	Cosine contents of the PCs 1-5
13..15	$\chi^2_{\mathcal{N},1}$, $\chi^2_{\mathcal{N},2}$, $\chi^2_{\mathcal{N},3}$	Goodness of fit of a Gaussian to PCs 1-3
16..20	$f_1^{\mathrm{acf}}..f_5^{\mathrm{acf}}$	Friction constant derived from a fit to the autocorrelation function of PCs 1-5
21	μ^{γ}	Average ruggedness of the energy landscape
22	skew^{γ}	Skewness of the distribution of ruggedness values of each degree of freedom
23	kurt^{γ}	Kurtosis of the distribution of these ruggedness values
24	μ^{RMSD}	Average root-mean square deviation from the X-ray structure
25	c_v^{RMSD}	Standard deviation (% of mean) of the root-mean square deviation from the X-ray structure
26	μ^{RMSF}	Average residual fluctuations with respect to the average ensemble structure
27	$c_v^{r_g}$	Standard deviation (% of mean) of the radius of gyration
28..31	c_v^{struct}, c_v^{α} , c_v^{β} , c_v^{turn}	Standard deviation (% of mean) of secondary structure content: total, α-helix, β-sheet, turn
32	μ^{SAS}	Average solvent accessible surface area
33	c_v^{SAS}	Standard deviation (% of mean) of the solvent accessible surface area
34	S_{RMSF}	RMSF entropy

3.11 Variational Autoencoder

Variational Autoencoders (VAEs) are artificial neural networks used to find meaningful, low-dimensional embeddings for a data sets or generating new instances resembling the original data [68, 69]. VAEs have an encoder-decoder type architecture with the additional twist of adding noise to the encodings during training time, which enforces encoding to be a continuous function. They were used in this work to perform dimension reduction of fingerprint spaces to visualize them in 2D. The two sub-networks were composed as follows:

- The variational encoder consisted of an input layer, several hidden layers, an encoding layer and a sampling layer. When training the encoder or using it for prediction, a fingerprint $\mathbf{v} \in \mathbb{R}^d$ from the data set was copied into the input layer. Therefore, the input layer had d nodes, where d is the number of fingerprint variables. This activation was then passed through the hidden layers on to the encoding layer with four nodes, which was twice the number of dimensions in the embedding. The sampling layer interpreted two of the embedding layer nodes as mean values and the other two as variances. These means and variances described a 2D normal distribution from which an encoding vector was randomly drawn. This encoding vector was used as output for the variational encoder during training.
  After training, for fingerprint space visualization, only the mean values from the encoding layer were used.

- The decoder received the 2D encoding vector from the variational encoder as input activation and passed it through several hidden layers on to a reconstruction layer with d nodes. The structure of hidden layers in the decoder was a mirrored version of that used in the variational encoder. The reconstruction layer activation was then compared with the input fingerprint $\mathbf{v}$ to determine the reconstruction accuracy and calculate a loss from that.

In VAEs in this study, dense layers were used exclusively (except the sampling layer described above), meaning that the activation $\mathbf{a}_i$ in layer i was computed from the activation $\mathbf{a}_{i-1}$ in the preceding layer as

$$\mathbf{a}_i = f(\mathbf{W}_i \mathbf{a}_{i-1} + \mathbf{b}_i), \tag{22}$$

using the learnable weight matrix $\mathbf{W}_i$ and bias vector $\mathbf{b}_i$ and the Scaled Exponential Linear Unit (SELU) activation function f [70]. During training, learnable parameters were adjusted using Adaptive Moment Estimation (Adam) [71] to minimize the loss function [68]

$$\mathcal{L} = \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{reco}}, \tag{23}$$

which was composed of the reconstruction loss $\mathcal{L}_{\text{reco}}$ and the Kullback-Leibler (KL) loss $\mathcal{L}_{\text{KL}}$. $\mathcal{L}_{\text{reco}}$ was computed as the mean squared error between the reconstruction layer activation and the input fingerprint. $\mathcal{L}_{\text{KL}}$ was calculated as the KL divergence between the normal distribution described by the encoding layer and a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$. These two terms regulated each other and the encoder had to find an optimal balance between them: The sharper it located the encoding vector in the k-dimensional embedding space, the larger the KL loss $\mathcal{L}_{\text{KL}}$ got, but the easier it got for the decoder to reconstruct the input fingerprint from the encoding, reducing the reconstruction loss $\mathcal{L}_{\text{reco}}$. The KL loss $\mathcal{L}_{\text{KL}}$ also enforced the encoding vectors of the data set to be normally distributed (with mean $\mu = 0$ and variance $\sigma^2 = 1$) overall. [69]

As mentioned in Ch. 2.1, different fingerprint spaces had to be treated separately and therefore required their own VAE each. The respective VAE layouts are listed in Tab. 2. Choices for layout and parameters that were shared between all used VAEs are listed in Tab. 3.

Table 2: VAE layouts for reducing dimensionality of different fingerprint spaces are displayed as sequences of numbers, each representing one layer with the given number of neurons. SSI denotes secondary structure information.

Fingerprint name	Fingerprint variables	Encoder layout	Decoder layout
MSM	13	13-8-4-2	2-4-8-13
Dynasome 1	31	31-16-8-4-2	2-4-8-16-31
Dynasome 1 without SSI	27	27-16-8-4-2	2-4-8-16-27

Table 3: Network layout and training parameters of the VAEs used in this work.

Parameter	Value
Type of layer	Dense
Activation function	Scaled Exponential Linear Unit (SELU) [70]
Embedding dimensions	2
Loss function	KL loss $\mathcal{L}_{\text{KL}}$ + reconstruction loss $\mathcal{L}_{\text{reco}}$ [68]
Optimizer	Adaptive Moment Estimation (Adam) [71]
Initial learning rate	10^{-5}
Number of epochs trained	Determined by minimizing validation loss 407 (MSM), 463 (Dyn 1), 403 (Dyn 1 w/o SSI)
Batch size	16
Training / Validation data split	80 % / 20 %
Input preprocessing	Standardization ($\mu = 0$, $\sigma^2 = 1$)

VAEs were trained on 80 % of the respective fingerprints (training data), whereas the

other 20 % were held back for validation. Weights were adjusted after a batch of 16 fingerprints was processed, and gradients were averaged over the batches. Training was furthermore organizied in epochs: Within every epoch the VAEs processed every fingerprint from the training data once. After every epoch their performance was evaluated by calculating the loss function $\mathcal{L}$ for the validation data. When this validation loss did not reach a new minimum for 50 consecutive epochs, training was stopped and the weights from the epoch with minimal validation loss were recovered.

3.12 Fingerprint Analysis

In this study, a set of fingerprints consists of 600 fingerprints, three for each of 200 proteins. These data were labelled in a sense that for every fingerprint it is known which protein it belongs to, and a label (i.e. number in [1..200]) can be assigned to it.

Three different analyses were conducted for every set of fingerprints in this study. The first one is based on comparison of distances between fingerprints of the same protein and distances between fingerprints of different proteins. It was used to proof protein-specific information in fingerprints and quantify it. The second and third are based on measuring how well the fingerprints match the cluctering implied by the labels, using conventional clustering measures and a nearest neighbor clustering measure, respectively. We used these to validate results from the distance-based analysis.

3.12.1 Distance-based Analysis

All pairwise euclidean distances between any two fingerprints were calculated and divided into two subsets: One contained all distances between trajectories of identical proteins (intra-protein), and the other one contained all distances between trajectories of different proteins (inter-protein). Because the distances covered multiple orders of magnitude, the distributions of logarithmic (base 10) intra- and inter-protein distances were considered. Their respective means μ_{intra} and μ_{inter} were calculated. The difference of these means

$$\beta = \mu_{\text{inter}} - \mu_{\text{intra}} \tag{24}$$

was used as an estimator for the amount of protein-specific information captured by the fingerprint: The larger β, the more protein specific the fingerprints. Because β is a difference of logarithmic distances, it corresponds to a factor of actual distances. This factor, $10^{-\beta}$, by which intra-protein distances are smaller compared to inter-protein distances, on average, was computed as well.

3.12.2 Conventional Clustering Measures

To quantify how well a particular clustering describes a set of data points, several well established measures are available. Usually these are calculated to determine the quality of cluster assignments, taking positions of data points as ground truth. In this work, it was attempted to use them the other way around — measuring the quality of fingerprints (data point positions) given their labels (cluster assignments) as ground truth. For all sets of fingerprints, the following three clustering measures were calculated:

1. The Davies-Bouldin index [72]

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(\mathbf{c}_i, \mathbf{c}_j)} \right) \tag{25}$$

   was computed from the number of clusters n, the cluster centers $\mathbf{c}$, the average distance σ from a cluster center $\mathbf{c}$ to its elements using the euclidean distance measure $d(\cdot, \cdot)$. A low Davies-Bouldin index corresponds to low intra-cluster spread (σ) and high inter-cluster distances ($d(\mathbf{c}_i, \mathbf{c}_j)$) and therefore indicates good clustering.

2. The Dunn index [73]

$$D = \frac{\min_{1 \leq i < j \leq n} d(\mathbf{c}_i, \mathbf{c}_j)}{\max_{1 \leq l \leq n} d'(l)} \tag{26}$$

   was calculated, where $d'(l)$ represents the maximal distance between any two elements in cluster l. It compares the minimal inter-cluster distance to the maximal intra-cluster distance. A high Dunn index indicates good clustering.

3. The silhouette coefficient $\overline{s}$ was computed as the mean silhouette value [74]

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{27}$$

   over all data points i. Here, $a(i)$ denotes the mean distance from point i to all other points in the same cluster and $b(i)$ denotes the mean distance from point i to all points of the nearest other cluster. The nearest other cluster to point i is the one minimizing $b(i)$. Silhouette values range from -1 to 1. Negative silhouette values $s(i)$ indicate that the corresponding point i would better fit in another cluster [74]. The larger the silhouette coefficient $\overline{s}$, the better the clustering fits the data.

3.12.3 Nearest Neighbor Based Analysis

As mentioned earlier, the three conventional clustering measures listed in the previous chapter were originally designed to tackle a different problem than the one at hand. When applied to our problem, they might be biased by the different dimensionalities of fingerprints and/or by outliers. Hence they are complemented by a nearest-neighbor graph based clustering measure, which is assumed to work better for the comparison made here.

To calculate it, for every data point, the k nearest neighbors (NN) were scanned for points with the same cluster label. The number of such points found was then divided by the total number of other points sharing the label to obtain the fraction of these points within the k-NN realm. The mean of this fraction over all points of the data set was calculated and used as the k-NN score. The closer points sharing a label are to each other compared to others, the higher this score and the better the clustering. The parameter k was varied in the range 1–600.

3.13 Bootstrapping

Bootstrapping is a resampling method used in this work to estimate the error of mean for sets of logarithmic pairwise distances between fingerprints.

From each set of logarithmic distances $\mathbf{d} = (d_1, d_2, ..., d_n)$ of size n, 10^5 bootstrap resamples $(\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, ..., \tilde{\mathbf{d}}_{10^5})$ were drawn by picking n random elements from $\mathbf{d}$ with replacement 10^5 times. For each of these resamples, the mean was computed to obtain a sample distribution of 10^5 mean values. Confidence intervals were then computed from this sample distribution by fitting a normal distribution to it. [75]

4 Results and Discussion

To capture and compare the dynamics of different proteins, $3 \times 1\,\mu$s MD trajectories were generated for 200 proteins each. For every trajectory, an MSM was constructed and an MSM fingerprint was extracted (Ch. 3.9). A Dynasome 1 fingerprint was captured for every trajectory as well, using the approach from previous work (Ch. 3.10). Results of our analyses of the MSM fingerprints and of the Dynasome 1 fingerprints are presented and discussed in Chs. 4.2 and 4.3, respectively. Results of a comparison of the two sets of fingerprints regarding their protein specificity are presented and discussed in Ch. 4.4.

During MSM construction tICA (Ch. 3.3) was used to reduce the dimensionality of the MD trajectories by identifying slow collective motions. We first present and discuss our investigation of those tICAs, so that the reader can better comprehend the main results of this study afterwards.

4.1 tICA Projections

TICA was applied to each of the 200×3 MD trajectories generated for this study (Ch. 3.3). Many of the projections show similarities to cosines in the slowest tICs. As shown for an exemplary protein (E. coli asparagine synthetase, PDB code: 11AS [76]) in Fig. 4a, the projection onto the slowest component resembles half a period of a cosine and the projection onto the second slowest component resembles a full period. In the 2D space spanned by the two slowest components, the trajectories form a U-shape for that reason (Fig. 4b). The protein changes its conformation from one end of the 'U' to the other steadily, as illustrated by the color code which indicates time, ranging from purple at the beginning of the trajectory to yellow at its end. For other proteins, in contrast, the tICA projections reveal metastable behaviour, as shown exemplary in Figs. 4c and 4d for the human Pin1 WW domain (PDB code: 2F21 [77]).

The U-shaped projections indicate that the proteins undergo a very slow conformational change during the simulation, but we did not identify such a motion investigating animated trajectories using the molecular visualization software PyMOL [78].

The extent to which the tICA projection resembles cosines can be determined using cosine contents [79]. Cosine contents are calculated for each tIC as the inner product of the trajectory projected onto that tIC and a cosine with fitting period. To determine how strongly a projection resembles a U-shape, we calculated the mean of the cosine contents of the two leading tICs for every trajectory. They are plotted against the respective protein primary sequence length in Fig. 5. A positive correlation between cosine content and protein size is observed with a Pearson Correlation Coefficient (CC) of 0,43. The two example proteins above, asparagine synthetase and Pin1 WW domain, have primary sequence lengths of 330 and 35, respectively.

Larger proteins have higher-dimensional trajectories and conformational spaces. As a consequence, in general, longer simulations are needed to sample all their native conformations. The correlation of cosine content with protein size therefore indicates that insufficient sampling can be a reason for these U-shaped projection. We follow up on that point in the next paragraphs.

The cosine-like motion is known to appear in PCA projections (Ch. 3.2) of high-dimensional random walks [79, 80]. Their emergence in PCs of random walks was also proven ana-

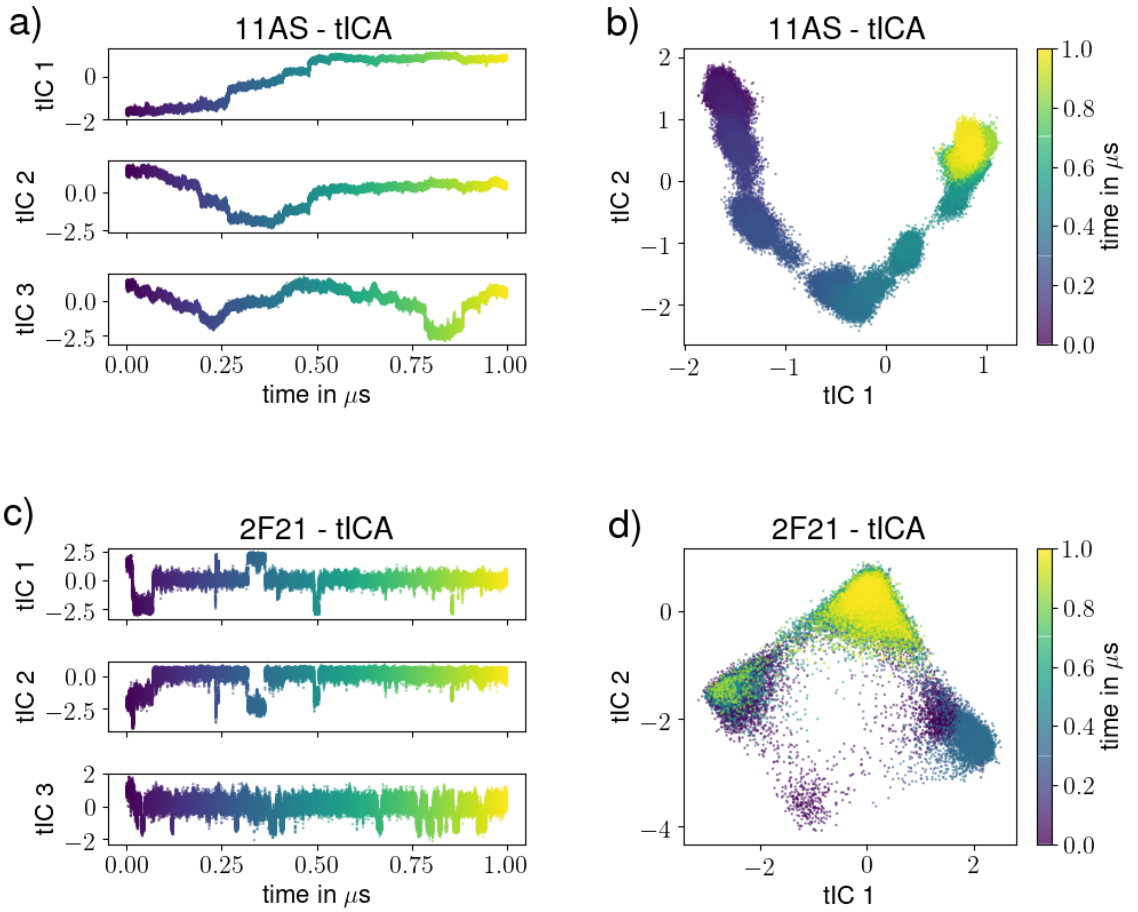


Figure 4: tICA projections of exemplary proteins asparagine synthetase (11AS) and Pin1 WW domain (2F21). **a)** Time series of 1D projections onto the first three tICs of a trajectory of asparagine synthetase. The motion along tICs one and two resemble half a period and a full period of a cosine, respectively. **b)** The projection onto the two slowest tICs of the asparagine synthetase trajectory show a U-shape. Color encodes time, revealing that the protein slowly changes its conformation from the purple side of the 'U' to the yellow side during the simulation. **c)**, **d)** Projections onto the leading tICs of a trajectory of Pin1 WW domain reveal metastable behaviour in 1D time series (**c**) and 2D projection (**d**).

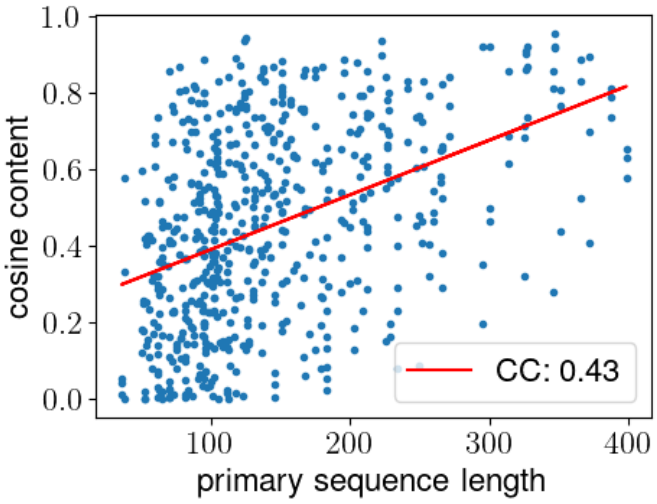


Figure 5: Dependence of cosine content of tICA projections on primary sequence length. The blue dots show the mean cosine content of the two leading tICs for the 600 MD trajectories used in this work. The Pearson Correlation Coefficient between cosine content and primary sequence length is 0,43. A linear least-squares regression is shown in red.

lytically [79]. To investigate the emergence of such motions in tICA projections, we conducted tICAs and PCAs of high-dimensional random walks. Figure 6 shows a comparison of the two different projections of a 300D random walk. It illustrates that cosine-like motions in the tICs are even more clearly visible and less noisy than in the PCs for random walks. We found that this statement also holds true for protein MD trajectories by comparing their tICA and PCA projections.

We interpret the high similarity between tICA projection of proteins and random walks as another indicator for insufficient sampling, meaning our simulation time of 1 μs per trajectory is likely too short to cover all relevant protein motions. Our interpretation is consistent with previous findings that some protein motions occur on a milliseconds timescale [18].

Because the tICA projections were used to construct MSMs for our MSM fingerprint, we will briefly discuss some implications of U-shaped tICA projections on the MSM in that context. First, proteins, whose tICA projections strongly resemble U-shapes, do not repeatedly visit metastable conformations — a key assumption to derive MSMs from the trajectories that accurately describe the protein kinetics.

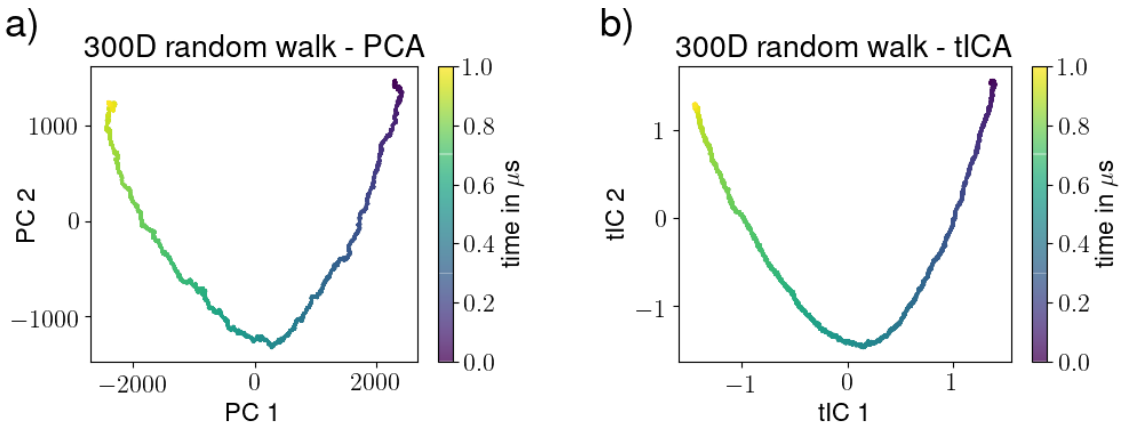


Figure 6: 2D projections of a 300D random walk. Color represents simulation time. **a)** The PCA projection shows a U-shape, as predicted by previous theoretical studies [79]. **b)** The tICA projection shows a U-shape too, and deviations from it are even smaller than for the PCA. The lag time used is $\tau = \frac{T}{333}$ with the simulation time T, analogous to the MD simulations.

Second, the transition pattern for all U-shaped trajectories is very similar, regardless of where states are placed. Nearly all transitions are in forward direction, going from state 1 to state 2, from state 2 to state 3 and so forth. Backward transitions or transitions between non-neighbouring states rarely occur. This strong similarity in transition patterns is a piece of information shared among all those trajectories, and therefore also among many proteins, making them more difficult to distinguish.

The third implication is on the coarse-graining of MSMs: A 200-state MSM was constructed from the tICA projection and subsequently coarse-grained to a 10-state MSM using PCCA+ (Ch. 3.8). This method relies on identifying groups of states that share more transitions with each other than with states outside the group. The transition pattern described above, however, connects every state to its neighbouring states in the same way. It therefore likely causes high uncertainty in the coarse-graining, and this uncertainty propagates into the MSM timescales and thereby into the fingerprints. Hence we expect that an MSM fingerprint is unable to precisely pinpoint the dynamics of a trajectory if its tICA projections show high similarity to cosines.

To sum up, we observed that for many proteins their tICA projections resemble cosines, most likely because our trajectories were too short to sample all their conformations and dynamics. This resemblance likely makes MSM fingerprints extracted from these trajectories less precise and less protein specific. Despite that, we did not exclude any proteins from further analyses to allow a fair comparison between different fingerprints.

4.2 MSM Fingerprints

MSM fingerprints were extracted from every trajectory by constructing an MSM (Ch. 3.9), using the tICA projections discussed in the previous chapter. An MSM fingerprint consists of six implied timescales and seven eigenvector entropies, captured from a single MSM. Hence every fingerprint is one point in the 13D MSM fingerprint space. Three trajectories were generated for each of the 200 proteins, resulting in three fingerprints per protein. Capturing multiple fingerprints per protein enables us to investigate the fingerprints' protein specificity.

To visualize the distribution of the MSM fingerprints, they were projected onto two dimensions using PCA (Ch. 3.2) and a VAE (Ch. 3.11). The projections are shown in Figs. 7a and 7b, respectively. The figures only include proteins with low fingerprint spread. To select these, the variance of the three fingerprints was calculated for every protein as a measure of spread. Then, the mean of those variances over all proteins was calculated and chosen as threshold, to exclude proteins with above average fingerprint spread.
In both projections, there are some outlier proteins covering their small, unique part of fingerprint space, but for the most part, the regions occupied by different proteins overlap. Large overlaps indicate low protein specificity, because these regions in fingerprint space can not be uniquely associated with one protein. Note however, that these are just 2D projections of a 13D fingerprint space and therefore most likely look more disordered than the original space, because proteins overlapping in the projections might be separated in another dimension.

To investigate the amount of protein-specific information extracted by the MSM fingerprints, all pairwise distances between fingerprints were computed and split into a set of intra-protein distances (between fingerprints of the same protein) and a set of inter-protein distances (between fingerprints of different proteins). For our data set of 200 proteins and three fingerprints each, this split yields 600 intra-protein distances and 179100 inter-protein distances. The distributions of distances within these two sets is shown in Fig. 8. Because they span multiple orders of magnitude, logarithmic distances (base 10) are shown. The logarithmic intra-protein distance distribution is shifted towards shorter distances compared to the inter-protein one. This shift implies that our MSM fingerprints contain protein-specific information.

To quantify the amount of protein-specific information captured by the fingerprints, we calculated the mean values of the two sets of logarithmic pairwise distances. The associated errors were obtained by bootstrapping (Ch. 3.13). The distributions of mean values of the bootstrap samples are shown in Fig. 9. The two distributions of means do not overlap. The difference of mean logarithmic intra- and inter-protein distances (Eq. (24)) is $\beta_{\mathrm{MSM}} = 0{,}049 \pm 0{,}006$. This difference means intra-protein distances are

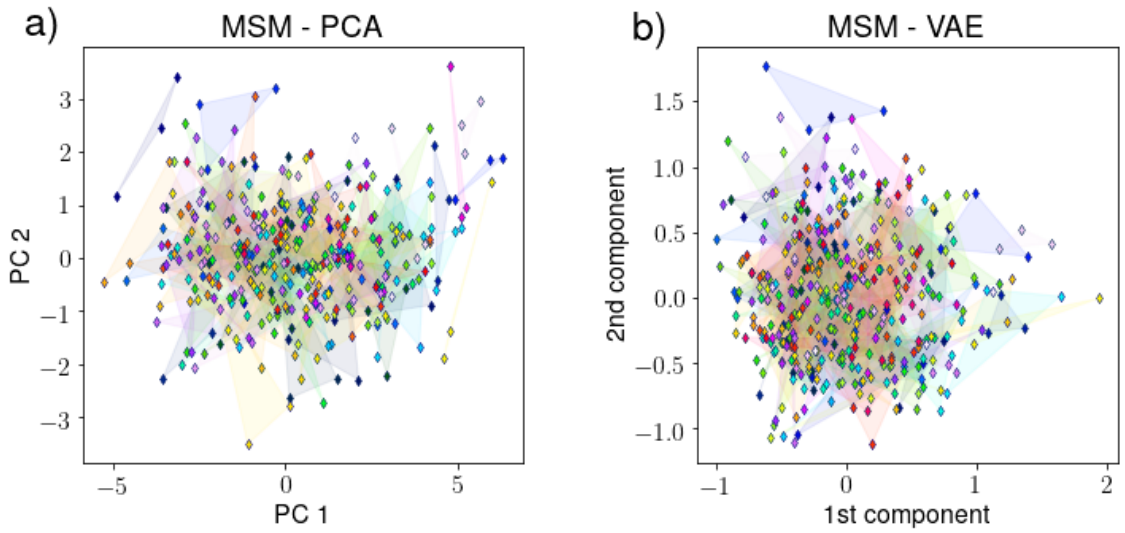


Figure 7: 2D projections of MSM fingerprint space by a) PCA and b) VAE. In both projections, fingerprints of a protein share the same color and are connected forming a triangular shade. In some cases, on the edge of the distribution, proteins cover their unique part of projection space. In the central regime, in contrast, many triangles overlap. Proteins with above average fingerprint spread are not shown.

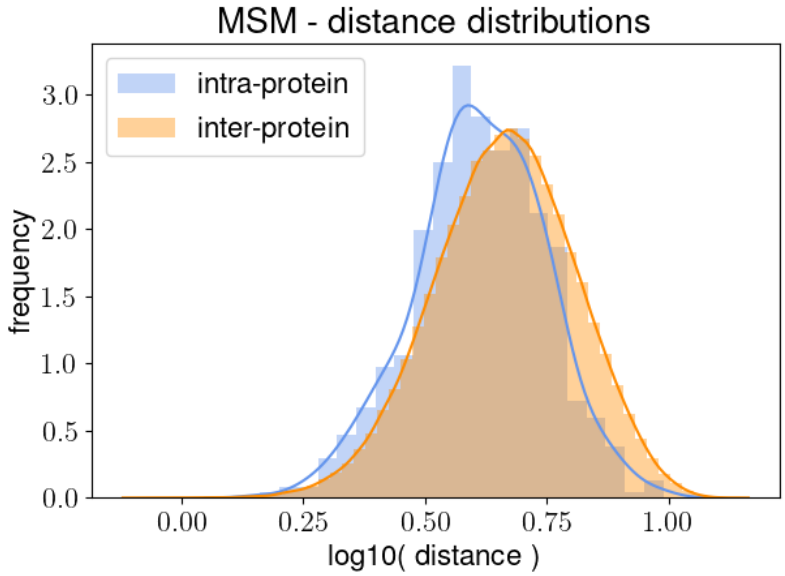


Figure 8: Distributions of distances between MSM fingerprints. The blue curve and histogram represent the distribution of intra-protein distances, the orange show the distribution of inter-protein distances.

shorter by a factor of $10^{-\beta_{\mathrm{MSM}}} = 0{,}893 \pm 0{,}012$. The errors were computed from the standard deviations of the bootstrap samples by Gaussian error propagation [81] and represent 68 % confidence intervals.

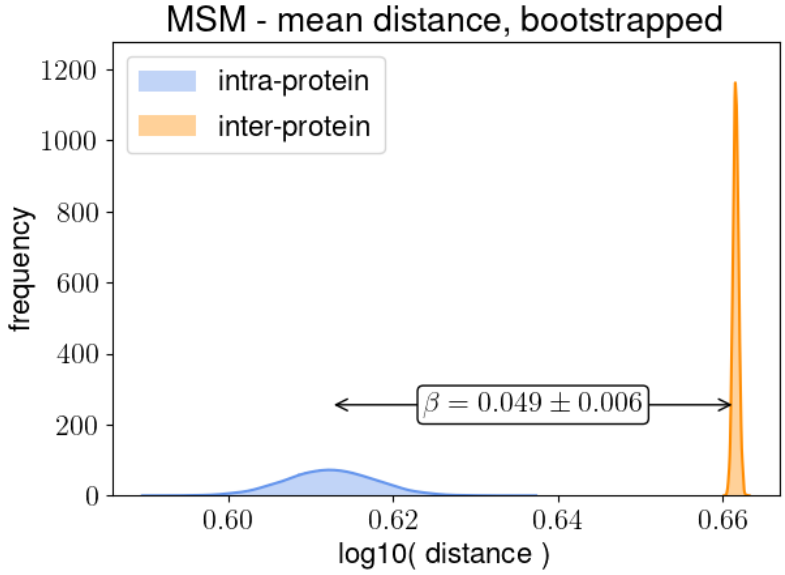


Figure 9: Means of bootstrap samples of logarithmic distances for MSM fingerprints. The blue and orange curve show the sample distributions of means of logarithmic intra- and inter-protein distances, respectively. The latter is narrower because more distances contribute to each mean. The means are separated by a gap of $\beta_{\mathrm{MSM}} = 0{,}049 \pm 0{,}006$.

From these observations we conclude that a significant amount of protein-specific information is captured in our MSM-based dynamics fingerprints. This result proves, that a conventional MSM construction pipeline with fixed parameters, as used here, can extract protein-specific dynamics information from 1 μs MD trajectories. However, the 2D-projections (Fig. 7) and the factor $10^{-\beta_{\mathrm{MSM}}} = 0{,}893 \pm 0{,}012$ imply that, except for some outliers, MSM fingerprints can not be uniquely associated with the protein they were captured from. In Ch. 4.4, the protein specificity of the MSM fingerprints is compared with the preceding fingerprint extraction approach as a benchmark.

4.3 Dynasome 1 Fingerprints

A Dynasome 1 fingerprint was calculated for every trajectory, extracting 31 observables from it, as described in Ch. 3.10. Here, they are examined regarding their protein spe-

cificity and how secondary structure based observables contribute to it. The results are used in Ch. 4.4 as a benchmark for the MSM fingerprints.

For the analysis of the Dynasome 1 fingerprints, the same procedure was used as for the MSM fingerprints in Ch. 4.2. Again, the mean logarithmic intra-protein distance is significantly shorter than the inter-protein one. Their difference is $\beta_{\mathrm{Dyn1}} = 0{,}056 \pm 0{,}005$, meaning intra-protein distances are shorter by a factor of $10^{-\beta_{\mathrm{Dyn1}}} = 0{,}88 \pm 0{,}01$.

The Dynasome 1 fingerprint space is visualized using 2D projections by PCA and a VAE in Figs. 10a and 10b, respectively. Again, we find regions uniquely covered by a single protein in these projections just for some outliers, but for the main part regions occupied by different proteins overlap.

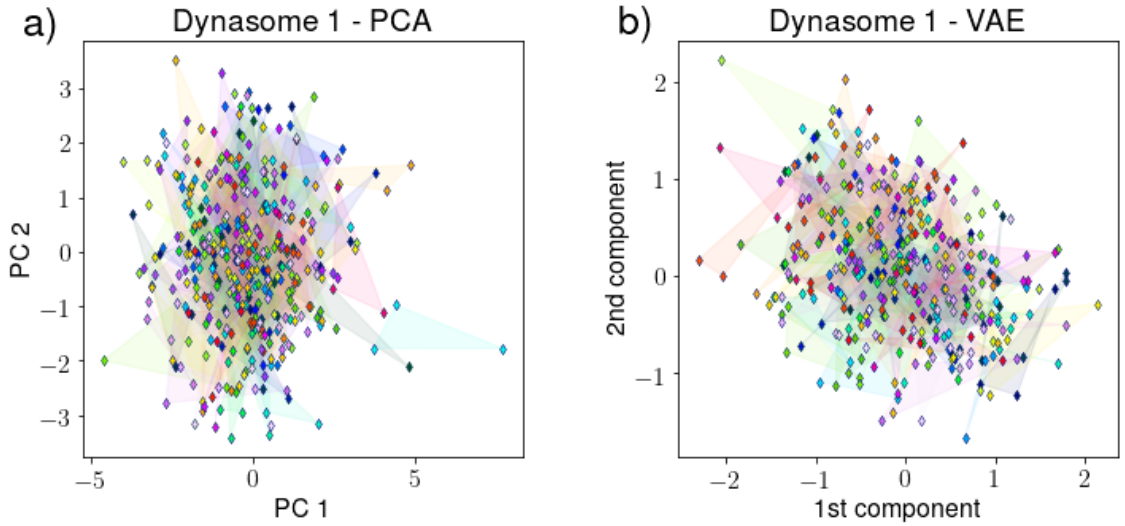


Figure 10: 2D projections of Dynasome 1 fingerprint space by a) PCA and b) VAE. In both projections, fingerprints of a protein share the same color and are connected forming a triangular shade. In some cases, on the edge of the distribution, proteins cover their unique part of projection space. In the central regime, in contrast, many triangles overlap. Proteins with above average fingerprint spread are not shown.

Further investigation of the VAE projection reveals that the Dynasome 1 fingerprints contain information on the proteins' secondary structure (SSI). To visualize that, the projection is shown in Figs. 11a and 11b with color code indicating α-helix and β-sheet content, respectively, of the protein a fingerprint was captured from. Both are correlated with the components of the projection. This correlation reveals that the VAE identifies secondary structure content as an important feature of the Dynasome 1 fingerprints.

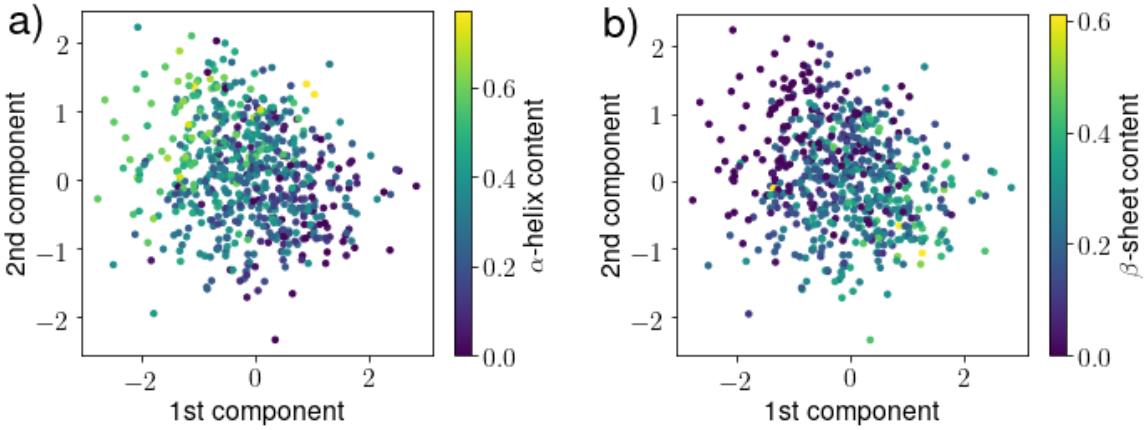


Figure 11: 2D projections of Dynasome 1 fingerprint space by a VAE. Color code indicates α-helix content and β-sheet content in **a)** and **b)**, respectively.

This SSI is very protein specific and therefore contributes to the overall protein specificity of the fingerprints. It is information on protein structure though and not on protein dynamics. To allow for a fair comparison with the MSM fingerprints, which do not contain structure information, we attempt to isolate information on dynamics within the Dynasome 1 fingerprints. To this aim, we discarded all observables that were calculated based on secondary structure knowledge. This criterion fitted observables № 28–31 (see Tab. 1). After discarding these, a set of 27 observables was left, № 1–20, 25–27 and 32–34, which we call 'Dynasome 1 without SSI' fingerprint.

Conducting the same analysis as before for these Dynasome 1 without SSI fingerprints reveals that, again, the mean logarithmic intra-protein distance is significantly shorter than the inter-protein one. Their difference here is $\beta_{\mathrm{Dyn1-SSI}} = 0{,}034 \pm 0{,}005$, meaning intra-protein distances are shorter by a factor of $10^{-\beta_{\mathrm{Dyn1-SSI}}} = 0{,}92 \pm 0{,}01$.

A VAE was trained on this fingerprint, too, projecting it onto two dimensions. Figures 12a and 12b show this projection with color code indicating α-helix and β-sheet content, respectively. There is no correlation between secondary structure content and the VAE projection here.

Because there is no SSI obvious from the VAE projection anymore, we conclude our attempt to remove SSI from the Dynasome 1 fingerprints succeeded. We will not perform any more sophisticated analysis of secondary structure correlations here. In principle, we consider it possible to capture a fingerprint of a protein purely based on its dynamics, that still contains information on the protein structure (including SSI). We should therefore not conclude that a fingerprint utilizes knowledge of secondary structure just based on the fact that it shows some correlation to secondary structure contents. However, for

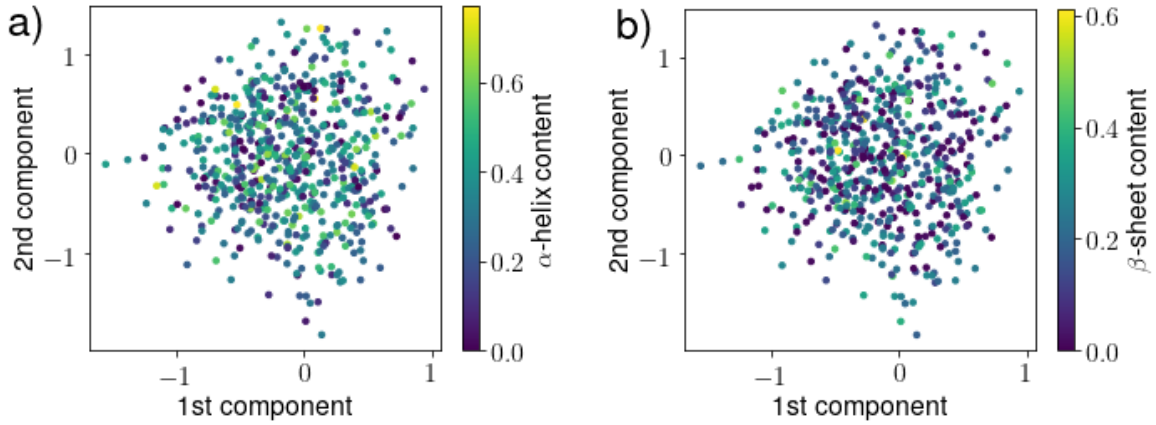


Figure 12: 2D projections of Dynasome 1 without SSI fingerprint space by a VAE. Color code indicates α-helix content and β-sheet content in **a)** and **b)**, respectively.

the Dynasome 1 fingerprints, we know that explicit knowledge of secondary structure entered the calculation of observables № 28–31 (see Tab. 1). For these reasons, we think that excluding these observables is the most valid way to transform the Dynasome 1 fingerprint into a purely dynamics based fingerprint and remove the strong dependency on secondary structure content.

To sum up, we found that the Dynasome 1 fingerprints also contain a significant amount of protein-specific information, partially on protein secondary structure. By discarding observables № 28–31 from the fingerprint, this SSI was removed, creating the 27D Dynasome 1 without SSI fingerprint. It solely contains information on dynamics, but still carries a significant amount of protein-specific information. Both these fingerprints are used as a benchmark for the MSM fingerprints in the next chapter.

4.4 Comparison of Fingerprints

Here, the protein specificity of the MSM, Dynasome 1 and Dynasome 1 without SSI fingerprints are compared, using the different measures introduced in Ch. 3.12.

4.4.1 Distance-based Analysis

The obtained β-values for the three fingerprint sets are listed in Tab. 4. The higher this number, the more protein-specific information is captured by the respective fingerprints. The associated errors represent 68 % confidence intervals (1σ). The factor $10^{-\beta}$, by which the mean intra-protein distance is smaller than the mean inter-protein distance, was computed from the β-values and is shown in the third column of Tab. 4.

Table 4: Distance-based analyses of the three fingerprint sets are summarized by assigning a β-value (Eq. (24)) to every set. The higher this number, the better the method performed. Third column shows the factor, by which the intra-protein distances are smaller than the inter-protein distances, on average.

Fingerprint	β	$10^{-\beta}$
Markov State Model	$0{,}049 \pm 0{,}006$	$0{,}893 \pm 0{,}012$
Dynasome 1	$0{,}056 \pm 0{,}005$	$0{,}88 \pm 0{,}01$
Dynasome 1 without SSI	$0{,}034 \pm 0{,}005$	$0{,}92 \pm 0{,}01$

The β-values of MSM and Dynasome 1 fingerprints are separated by a difference of $0{,}007 \approx 0{,}14\beta_{\mathrm{MSM}} \approx 1{,}17\sigma_{\mathrm{MSM}}$, whereas their 1σ intervals overlap. The two fingerprint sets therefore contain a similar amount of protein-specific information, with the Dynasome 1 fingerprints likely containing slightly more.

Comparing the MSM and Dynasome 1 without SSI fingerprints, their β-values are separated by a difference of $0{,}015 \approx 0{,}3\beta_{\mathrm{MSM}} \approx 2{,}5\sigma_{\mathrm{MSM}}$. Their 1σ intervals do not overlap, but their 2σ intervals do. Therefore, with high confidence $> 95\,\%$, the MSM fingerprints contain more protein-specific information than the Dynasome 1 without SSI fingerprints.

To examine the amount of protein-specific information lost from the Dynasome 1 fingerprints by removing SSI, the Dynasome 1 fingerprints with and without SSI are compared. Their β-values are separated by a difference of $0{,}022 \approx 0{,}4\beta_{\mathrm{Dyn1}} \approx 4{,}4\sigma_{\mathrm{Dyn1}}$, meaning SSI made up for $40\,\%$ of the separation of mean intra- and inter-protein distance of the Dynasome 1.

The amount of protein-specific information captured by the three fingerprints is statistically significant, but rather low. Intra-protein distances are shorter by factors of $0{,}893 \pm 0{,}012$, $0{,}88 \pm 0{,}01$ and $0{,}92 \pm 0{,}01$ compared to the inter-protein distances for the MSM, Dynasome 1 and Dynasome 1 without SSI fingerprints, respectively. These numbers indicate, that intra-protein distances and inter-protein distances are of similar size and on the same order of magnitude for all three fingerprint spaces. Proteins are therefore insufficiently separated in the fingerprint spaces to tell them apart given their fingerprints.

To sum up, we found that about $40\,\%$ of the Dynasome 1 fingerprint protein specificity originates from secondary structure information. The MSM fingerprints contain more dynamics-based protein-specific information than the Dynasome 1 fingerprints and a similar amount of overall protein-specific information (including SSI in Dynasome 1). The amount of protein-specific information captured by the three fingerprints is statist-

ically significant, but it is insufficient for telling proteins apart given their fingerprints. This insufficiency probably needs to be overcome to perform reliable function prediction with dynamics fingerprints, so there is room for further improvement in protein specificity.

4.4.2 Clustering Measures

In this chapter we present and discuss results of using conventional and nearest-neighbour-based clustering measures to examine protein specificity of the different fingerprint sets to validate our findings from the previous chapter.

The task of measuring protein specificity of fingerprints can also be described using the terminology of a closely related clustering problem: Given a set of 600 labeled data points $\mathbf{v} \in \mathbb{R}^d$ (fingerprints), assigned to 200 clusters (proteins) of three elements each, determine how well this clustering fits the data. Three established measures for clustering quality, namely the Davies-Bouldin index, Dunn index and the silhouette coefficient, were calculated for the three sets of fingerprints (Ch. 3.12.2).

The index values obtained for the different fingerprints are shown in Tab. 5. All these values indicate poor clustering. They consistently rank the Dynasome 1 fingerprints as best fitting the given cluster labels and the MSM fingerprints as worst. However, we should interpret these results with caution, because the applied measures were engineered to describe data sets with a small number of clusters compared to the number of elements within a cluster, and whose clusters are well manifested in the topological structure. The investigated fingerprints do not fulfil these two assumptions.

Table 5: Established conventional clustering measures describe how well the different fingerprints fit the clustering implied by protein names as labels. Note that these measures were developed for data sets with evident cluster structure and hence all these values imply poor clustering.

Fingerprint	Davies-Bouldin index lower is better	Dunn index higher is better	silhouette coefficient lower is better
Markov State Model	4,337	0,073	−0,316
Dynasome 1	2,679	0,128	−0,160
Dynasome 1 without SSI	2,903	0,124	−0,197

To examine how sensitive the conventional clustering measures are to violating these assumptions, they were applied to synthetic random data for comparison. For every set of fingerprints, 600 data points were drawn randomly from a normal distribution with a dimension equal to the fingerprint dimension, mean $\mu = 0$ and variance $\sigma^2 = 1$ for all

dimensions. These synthetic data were randomly assigned to 200 groups of three points each. Then the three conventional clustering measures were calculated for the artificial data sets. This process was repeated multiple times to get sufficient statistics. We found that, for all three conventional clustering measures, the dynamics fingerprints scored all significantly worse than random data of the respective dimension.

Within the random data, there is no information shared by points within the assigned groups that is not shared by all points, i.e. the equivalent to protein-specific information is 0. The dynamics fingerprints, in contrast, do contain such shared or protein-specific information, as shown in the previous chapters. In a measurement of protein-specific information, they should therefore score better compared with random data. They score worse in the conventional clustering measures though, so we deduce that those measures fail to capture protein-specific information from the fingerprints.

Furthermore, we calculated k-nearest-neighbor scores (Ch. 3.12.3) for the different sets of fingerprints. This score is calculated for every fingerprint by scanning its k nearest neighbours for fingerprints of the same protein. The number of such fingerprints found was averaged across the whole set and normalized. The k-NN score was calculated for $k \in [10, 20, .., 590]$. The obtained scores are shown in Fig. 13. Comparing the Dynasome 1 fingerprints with and without SSI, the Dynasome 1 fingerprints with SSI score higher for all values of k. MSM fingerprints score lowest of the three for $k < 330$ and highest for $k > 400$. For $k \in [330, 400]$, the ranking implied by the protein-specificity measure β in Ch. 4.4.1 is reproduced. The ranking implied by the conventional clustering measures is consistent with the k-NN scores with $k < 330$.

For data sets showing distinct clusters in their topological structure, we expect the k-NN score to be high even for very small k, because neighboring data points very likely belong to the same cluster in that case. For a comparison of such data sets, the k-NN score can only be a good measure of clustering if k is chosen small. For large k, all data points of the cluster are likely already included in the k-NN realm and therefore the k-NN score is less meaningful. In contrast, for data sets whose topological structure does not show distinct clusters, we expect a low score for small k, only slowly increasing with increasing k. For this kind of data, the k-NN score is less meaningful for small k, but for large k, it is a good measure of clustering.

For all three fingerprint sets, the scores are below 0.5 for $k \leq 100$, meaning that on average, within the 100 nearest neighbors of a fingerprint, there is less than one fingerprint of the same protein. Because the k-NN scores are low overall and only slowly increase with increasing k, we consider them meaningful only for large k.

To sum up, we aimed to validate the statements about protein-specific information of the different fingerprints obtained in Ch. 4.4.1, using conventional clustering measures

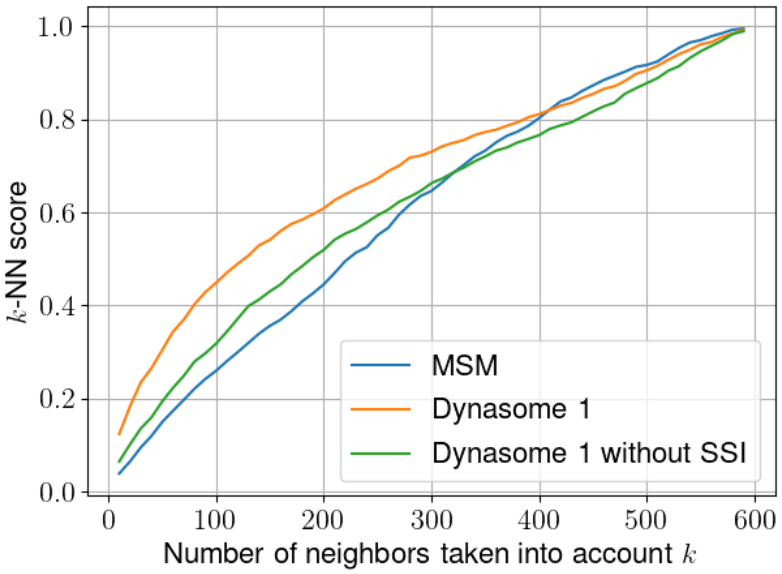


Figure 13: k-nearest neighbor score measured for different fingerprints. The Dynasome 1 fingerprints (orange) score highest for small k, the MSM fingerprints (blue) for large k.

and a nearest-neighbor based score. We found that the conventional clustering measures did not recognize any protein-specific information in the fingerprints and therefore are not meaningful for this validation. The k-NN score with $k > 330$ ranks the MSM fingerprints as more protein specific than the Dynasome 1 without SSI fingerprints, which is consistent with the results in Ch. 4.4.1. We consider $k > 330$ a relevant regime of k for the reasons stated above. For $k > 400$, the k-NN score ranks the MSM fingerprints even more protein specific than the Dynasome 1 fingerprints with SSI.

This agreement of the distance-based and the nearest-neighbor based analyses reinforces us to conclude that our MSM fingerprints capture a similar amount of protein-specific information compared to the preceding approach (Dynasome 1) containing SSI, and capture more dynamics-based protein-specific information.

5 Outlook

In this chapter, we briefly discuss some implication of this work for future research on the dynasome and on protein dynamics based function prediction. We also present some preliminary results of recent follow-up work.

5.1 Sampling

In Ch. 4.1, we identified that one 1 μs MD trajectory does not cover all relevant protein motions, especially for larger proteins. Based on our results, we suspect that insufficient sampling is a bottleneck for the protein specificity of MSM fingerprints. Hence we discuss the implications of sampling on MSM fingerprints and how to address this bottleneck here.

In the limit of infinitely long trajectories, the trajectory-specific noise $\boldsymbol{\delta}_{\text{traj}}$ vanishes. Thereby, any set of fingerprints from different trajectories of the same protein converge to the same point in fingerprint space, given that the fingerprint extraction method does not introduce any additional noise $\boldsymbol{\delta}_{\text{meth}}$. It is unclear how the noise decays to zero here (monotonously, fast/slowly, ...), but in general, we expect that more data (i.e. more simulation time) leads to less noise.

This convergence depends not only on how much simulation time is used but also on how it is used (sampling strategy). One straightforward sampling strategy is to invest all effort into computing one trajectory and make it as long as possible. This strategy has the advantage of exploring many different regions of phase space, but the drawback that sampling within these regions might be poor (if ergodicity is not reached). A competing approach is to compute many short trajectories starting from the same structure. This strategy enhances sampling in the starting region, but has the drawback of not exploring distant phase space regions. These two strategies can be combined into a two step simulation procedure: First, a semi-long trajectory is computed to explore different regions of phase space. Second, starting from different regions in phase space, multiple short trajectories are computed to enhance sampling within all regions previously explored. It is so far unclear which strategy serves best to construct MSM fingerprints that capture much protein-specific information and little noise. Note that a poor sampling strategy might limit the amount of information that can be gathered from a trajectory. For example, if an enzyme can open and close its binding pocket and only very short trajectories of the closed state are considered, no information about the open state and the opening/closing motion is available. Fingerprints extracted from such trajectories might very precisely describe dynamics within the closed state, but can not capture the general dynamics of the protein.

To increase the amount of sampling, the most straightforward way is to dedicate more computation time to the MD simulations. Computation time is limited through being costly though. Hence we aim to increase the amount of sampling generated without increasing computation time, by employing an efficient *enhanced sampling* strategy. Many different, sophisticated enhanced sampling strategies like replica exchange [82], metadynamics [83] or generalized simulated annealing [84] are available, but there is no comparative analysis on which works best to construct MSMs. Therefore, it is so far unclear which would yield the best improvement in sampling for MSM fingerprints.

5.2 MSM Construction

In Ch. 2.1, we discussed different contributions to dynamics fingerprints (Eq. 1) and from there on, neglected the difference between trajectory-specific noise $\boldsymbol{\delta}_{\text{traj}}$ and method-specific noise $\boldsymbol{\delta}_{\text{meth}}$. In recent follow-up work to this study, we differentiated them and found that both are important to consider. Here we discuss how the MSM construction pipeline can be altered to reduce method-specific noise $\boldsymbol{\delta}_{\text{meth}}$.

When our MSM fingerprint construction pipeline was applied to the same trajectory multiple times, the resulting fingerprints still had a surprisingly large spread, although not as large as for different trajectories ($\boldsymbol{\delta}_{\text{traj}} > \boldsymbol{\delta}_{\text{meth}} \gg 0$). We identified the random initializations during k-means clustering as the source of that noise by using the same initialization multiple times and observing the noise vanished.

These observations imply that an alternative MSM construction method, that is less reliant on k-means, could yield dynamics fingerprints with an improved signal-to-noise ratio. We consider several promising methods to improve MSM construction: Using a *Minimally-Coupled Subspaces Approach* [85], the high-dimensional coordinate space is decomposed into lower-dimensional subspaces ($d < 15$), minimizing correlations between subspaces. K-means is then performed on these subspaces. We expect k-means to introduce less noise this way.

Recently, deep learning methods were developed for MD data which can be used to construct MSMs [86, 87]. A *Variational Dynamics Encoder* (VDE) utilizes a neural network architecture based on VAEs (Ch. 3.11) to map an input trajectory onto a meaningful reaction coordinate [87]. This mapping allows to perform k-means on a 1D representation. Again, we expect it to introduce less noise here.

VAMPnets [86] are deep neural networks performing a dynamics-based clustering of MD data. They utilize the variational approach for Markov processes (VAMP) [88, 89] and map MD trajectory frames onto discrete states. Therefore, no other clustering algorithm is needed within a VAMPnet MSM construction pipeline. Hence VAMPnets are a promising tool to obtain fingerprints with reduced noise.

Furthermore, k-means can be exchanged for another clustering algorithm (like Gaussian Mixture Models or density-based methods) within the MSM construction pipeline used here. However, results of a study by Husic and Pande [56] imply that a pipeline using k-means (or Ward clustering) produces more reproducible MSMs compared with other clustering methods.

5.3 Function-Specific Information

To investigate whether our dynamics fingerprints contain any function-specific information, we split them into two groups — enzymes and non-enzymes — each containing fingerprints of 100 proteins. Then, an analysis similar to the distance-based analysis for protein specificity (Ch. 3.12.1) was conducted: We examined, whether distances between fingerprints of enzyme-enzyme and non-enzyme-non-enzyme pairs are shorter than distances between enzyme-non-enzyme pairs, on average. For the MSM fingerprints, this was not the case. For the Dynasome 1 with and without SSI it was, intra-group distances were shorter by factors of $0{,}996 \pm 0{,}002$ and $0{,}9973 \pm 0{,}0011$, respectively. These factors reveal that differences between function groups are small compared to those between proteins (Tab. 4), as well as in absolute value ($< 1\,\%$). We aim to further develop dynamics fingerprints in future work to increase the amount of protein- and function-specific information gathered.

References

[1] Clare M. O'Connor and Jill U. Adams. *Essentials of Cell Biology*. Cambridge, MA: NPG Education, 2010.

[2] Fuhao Zhang et al. 'DeepFunc: A Deep Learning Framework for Accurate Prediction of Protein Functions from Protein Sequences and Interactions'. *Proteomics* 19.12 (2019), e1900019. DOI: 10.1002/pmic.201900019.

[3] Serkan Erdin, Andreas Martin Lisewski and Olivier Lichtarge. 'Protein function prediction: towards integration of similarity metrics'. *Current opinion in structural biology* 21.2 (2011), pp. 180–188. DOI: 10.1016/j.sbi.2011.02.001.

[4] Naihui Zhou, Yuxiang Jiang et al. 'The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens'. *Genome biology* 20.1 (2019), p. 244. DOI: 10.1186/s13059-019-1835-8.

[5] Stephen F. Altschul et al. 'Basic local alignment search tool'. *Journal of Molecular Biology* 215.3 (1990), pp. 403–410. DOI: 10.1016/S0022-2836(05)80360-2.

[6] Cyrus Chothia and Arthur M. Lesk. 'The relation between the divergence of sequence and structure in proteins'. *The EMBO Journal* 5.4 (1986), pp. 823–826. URL: http://www.ncbi.nlm.nih.gov/pubmed/3709526.

[7] Qingtian Gong, Wei Ning and Weidong Tian. 'GoFDR: A sequence alignment based method for predicting protein functions'. *Methods (San Diego, Calif.)* 93 (2016), pp. 3–14. DOI: 10.1016/j.ymeth.2015.08.009.

[8] John C. Kendrew et al. 'A three-dimensional model of the myoglobin molecule obtained by x-ray analysis'. *Nature* 181.4610 (1958), pp. 662–666. DOI: 10.1038/181662a0.

[9] Kurt Wüthrich. 'Protein structure determination in solution by nuclear magnetic resonance spectroscopy'. *Science (New York, N.Y.)* 243.4887 (1989), pp. 45–50. DOI: 10.1126/science.2911719.

[10] Werner Kühlbrandt. 'The resolution revolution'. *Science (New York, N.Y.)* 343.6178 (2014), pp. 1443–1444. DOI: 10.1126/science.1251652.

[11] Ka Man Yip et al. 'Atomic-resolution protein structure determination by cryo-EM'. *Nature* 587.7832 (2020), pp. 157–161. DOI: 10.1038/s41586-020-2833-4.

[12] Florencio Pazos and Michael J. E. Sternberg. 'Automated prediction of protein function and detection of functional sites from structure'. *Proceedings of the National Academy of Sciences of the United States of America* 101.41 (2004), pp. 14754–14759. DOI: 10.1073/pnas.0404569101.

[13] Michael Ashburner et al. 'Gene ontology: tool for the unification of biology. The Gene Ontology Consortium'. *Nature genetics* 25.1 (2000), pp. 25–29. DOI: 10.1038/75556.

[14] 'The Gene Ontology resource: enriching a GOld mine'. *Nucleic acids research* 49.D1 (2021), pp. D325–D334. DOI: 10.1093/nar/gkaa1113.

[15] Ronghui You et al. 'GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank'. *Bioinformatics (Oxford, England)* 34.14 (2018), pp. 2465–2473. DOI: 10.1093/bioinformatics/bty130.

[16] Michal Brylinski and Jeffrey Skolnick. 'A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation'. *Proceedings of the National Academy of Sciences* 105.1 (2008), pp. 129–134. DOI: 10.1073/pnas.0707684105.

[17] David Lee, Oliver Redfern and Christine Orengo. 'Predicting protein function from sequence and structure'. *Nature reviews. Molecular cell biology* 8.12 (2007), pp. 995–1005. DOI: 10.1038/nrm2281.

[18] Gregory R. Bowman, Vijay S. Pande and Frank Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Vol. 797. Dordrecht: Springer Netherlands, 2014. DOI: 10.1007/978-94-007-7606-7.

[19] Carsten Kutzner et al. 'More bang for your buck: Improved use of GPU nodes for GROMACS 2018'. *Journal of computational chemistry* 40.27 (2019), pp. 2418–2431. DOI: 10.1002/jcc.26011.

[20] A. Keith Dunker et al. 'Intrinsic disorder and protein function'. *Biochemistry* 41.21 (2002), pp. 6573–6582. DOI: 10.1021/bi012159.

[21] Martin Karplus and John Kuriyan. 'Molecular dynamics and protein function'. *Proceedings of the National Academy of Sciences of the United States of America* 102.19 (2005), pp. 6679–6685. DOI: 10.1073/pnas.0408930102.

[22] Ulf Hensen et al. 'Exploring Protein Dynamics Space: The Dynasome as the Missing Link between Protein Structure and Function'. *PloS one* 7.5 (2012), pp. 1–16. DOI: 10.1371/journal.pone.0033931.

[23] Vijay S. Pande, Kyle Beauchamp and Gregory R. Bowman. 'Everything you wanted to know about Markov State Models but were afraid to ask'. *Methods (San Diego, Calif.)* 52.1 (2010), pp. 99–105. DOI: 10.1016/j.ymeth.2010.06.002.

[24] Brooke E. Husic and Vijay S. Pande. 'Markov State Models: From an Art to a Science'. *Journal of the American Chemical Society* 140.7 (2018), pp. 2386–2396. DOI: 10.1021/jacs.7b12191.

[25] Robert T. McGibbon and Vijay S. Pande. 'Variational cross-validation of slow dynamical modes in molecular kinetics'. *The Journal of chemical physics* 142.12 (2015). DOI: 10.1063/1.4916292.

[26] Gregory R. Bowman, Xuhui Huang and Vijay S. Pande. 'Using generalized ensemble simulations and Markov state models to identify conformational states'. *Methods (San Diego, Calif.)* 49.2 (2009), pp. 197–201. DOI: 10.1016/j.ymeth.2009.04.013.

[27] Gregory R. Bowman et al. 'Progress and challenges in the automated construction of Markov state models for full protein systems'. *The Journal of chemical physics* 131.12 (2009), p. 124101. DOI: 10.1063/1.3216567.

[28] Max Born and Robert Oppenheimer. 'Zur Quantentheorie der Molekeln'. *Annalen der Physik* 389.20 (1927), pp. 457–484. DOI: 10.1002/andp.19273892002.

[29] David J. Griffiths. *Introduction to electrodynamics*. Fourth edition. Cambridge: Cambridge University Press, 2017. DOI: 10.1017/9781108333511.

[30] Tom Darden, Darrin York and Lee Pedersen. 'Particle mesh Ewald: An N ·log(N) method for Ewald sums in large systems'. *The Journal of chemical physics* 98.12 (1993), pp. 10089–10092. DOI: 10.1063/1.464397.

[31] Victor Gold, ed. *The IUPAC Compendium of Chemical Terminology*. Research Triangle Park, NC: International Union of Pure and Applied Chemistry (IUPAC), 2019. DOI: 10.1351/goldbook.

[32] J. Dana. Honeycutt and Hans C. Andersen. 'Molecular dynamics study of melting and freezing of small Lennard-Jones clusters'. *The Journal of physical Chemistry* 91.19 (1987), pp. 4950–4963. DOI: 10.1021/j100303a014.

[33] Kresten Lindorff-Larsen et al. 'Improved side-chain torsion potentials for the Amber ff99SB protein force field'. *Proteins* 78.8 (2010), pp. 1950–1958. DOI: 10.1002/prot.22711.

[34] Tim Meyer et al. 'MoDEL (Molecular Dynamics Extended Library): a database of atomistic molecular dynamics trajectories'. *Structure (London, England : 1993)* 18.11 (2010), pp. 1399–1409. DOI: 10.1016/j.str.2010.07.013.

[35] Sander Pronk et al. 'GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit'. *Bioinformatics (Oxford, England)* 29.7 (2013), pp. 845–854. DOI: 10.1093/bioinformatics/btt055.

[36] Hans W. Horn et al. 'Development of an improved four-site water model for biomolecular simulations: TIP4P-Ew'. *The Journal of chemical physics* 120.20 (2004), pp. 9665–9678. DOI: 10.1063/1.1683075.

[37] Helen M. Berman et al. 'The Protein Data Bank'. *Nucleic acids research* 28.1 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235.

[38] Giovanni Bussi, Davide Donadio and Michele Parrinello. 'Canonical sampling through velocity rescaling'. *The Journal of chemical physics* 126.1 (2007), p. 014101. DOI: 10.1063/1.2408420.

[39] Herman J. C. Berendsen et al. 'Molecular dynamics with coupling to an external bath'. *The Journal of chemical physics* 81.8 (1984), pp. 3684–3690. DOI: 10.1063/1.448118.

[40] M. Parrinello and A. Rahman. 'Polymorphic transitions in single crystals: A new molecular dynamics method'. *Journal of Applied Physics* 52.12 (1981), pp. 7182–7190. DOI: 10.1063/1.328693.

[41] Shuichi Miyamoto and Peter A. Kollman. 'Settle: An analytical version of the SHAKE and RATTLE algorithm for rigid water models'. *Journal of Computational Chemistry* 13.8 (1992), pp. 952–962. DOI: 10.1002/jcc.540130805.

[42] Berk Hess et al. 'LINCS: A linear constraint solver for molecular simulations'. *Journal of Computational Chemistry* 18.12 (1997), pp. 1463–1472. DOI: 10.1002/(SICI)1096-987X(199709)18:12{\textless}1463::AID-JCC4{\textgreater}3.0.CO;2-H.

[43] Douglas Brune and Sangtae Kim. 'Predicting protein diffusion coefficients'. *Proceedings of the National Academy of Sciences of the United States of America* 90.9 (1993), pp. 3835–3839. DOI: 10.1073/pnas.90.9.3835.

[44] Svante Wold, Kim Esbensen and Paul Geladi. 'Principal component analysis'. *Chemometrics and Intelligent Laboratory Systems* 2.1-3 (1987), pp. 37–52. DOI: 10.1016/0169-7439(87)80084-9.

[45] Hervé Abdi and Lynne J. Williams. 'Principal component analysis'. *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4 (2010), pp. 433–459. DOI: 10.1002/wics.101.

[46] Molgedey and Schuster. 'Separation of a mixture of independent signals using time delayed correlations'. *Physical review letters* 72.23 (1994), pp. 3634–3637. DOI: 10.1103/PhysRevLett.72.3634.

[47] Guillermo Pérez-Hernández et al. 'Identification of slow molecular order parameters for Markov model construction'. *The Journal of chemical physics* 139.1 (2013), p. 015102. DOI: 10.1063/1.4811489.

[48] 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* 134.6 (2011), p. 065101. DOI: 10.1063/1.3554380.

[49] Frank Noé and Cecilia Clementi. 'Kinetic distance and kinetic maps from molecular dynamics simulation'. *Journal of chemical theory and computation* 11.10 (2015), pp. 5002–5011. DOI: 10.1021/acs.jctc.5b00553.

[50] William C. Swope, Jed W. Pitera and Frank Suits. 'Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory †'. *The Journal of Physical Chemistry B* 108.21 (2004), pp. 6571–6581. DOI: 10.1021/jp037421y.

[51] Wei Wang et al. 'Constructing Markov State Models to elucidate the functional conformational changes of complex biomolecules'. *WIREs Computational Molecular Science* 8.1 (2018). DOI: 10.1002/wcms.1343.

[52] Hongli Liu et al. 'The misfolding mechanism of the key fragment R3 of tau protein: a combined molecular dynamics simulation and Markov state model study'. *Physical chemistry chemical physics : PCCP* 22.19 (2020), pp. 10968–10980. DOI: 10.1039/c9cp06954b.

[53] Andreas Tosstorff, Günther H. J. Peters and Gerhard Winter. 'Study of the interaction between a novel, protein-stabilizing dipeptide and Interferon-alpha-2a by construction of a Markov state model from molecular dynamics simulations'. *European journal of pharmaceutics and biopharmaceutics : official journal of Arbeitsgemeinschaft fur Pharmazeutische Verfahrenstechnik e.V* 149 (2020), pp. 105–112. DOI: 10.1016/j.ejpb.2020.01.020.

[54] Xiaojun Zeng et al. 'Unfolding mechanism of thrombin-binding aptamer revealed by molecular dynamics simulation and Markov State Model'. *Scientific reports* 6 (2016), p. 24065. DOI: 10.1038/srep24065.

[55] Martin K. Scherer et al. 'PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models'. *Journal of Chemical Theory and Computation* 11 (Oct. 2015), pp. 5525–5542. DOI: 10.1021/acs.jctc.5b00743.

[56] Brooke E. Husic and Vijay S. Pande. 'Ward Clustering Improves Cross-Validated Markov State Models of Protein Folding'. *Journal of chemical theory and computation* 13.3 (2017), pp. 963–967. DOI: 10.1021/acs.jctc.6b01238.

[57] Yue Guo, Mojie Duan and Minghui Yang. 'The Observation of Ligand-Binding-Relevant Open States of Fatty Acid Binding Protein by Molecular Dynamics Simulations and a Markov State Model'. *International journal of molecular sciences* 20.14 (2019). DOI: 10.3390/ijms20143476.

[58] S. Lloyd. 'Least squares quantization in PCM'. *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[59] Frank Noé. 'Statistical inefficiency of Markov model count matrices' (2015). URL: https://core.ac.uk/download/pdf/267951465.pdf.

[60] Kyle A. Beauchamp et al. 'MSMBuilder2: Modeling Conformational Dynamics at the Picosecond to Millisecond Scale'. *Journal of chemical theory and computation* 7.10 (2011), pp. 3412–3419. DOI: 10.1021/ct200463m.

[61] Jan-Hendrik Prinz et al. 'Markov models of molecular kinetics: generation and validation'. *The Journal of chemical physics* 134.17 (2011), p. 174105. DOI: 10.1063/1.3565032.

[62] Benjamin Trendelkamp-Schroer et al. 'Estimation and uncertainty of reversible Markov models'. *The Journal of chemical physics* 143.17 (2015), p. 174101. DOI: 10.1063/1.4934536.

[63] Frank Noé. 'Probability distributions of molecular observables computed from Markov models'. *The Journal of chemical physics* 128.24 (2008), p. 244103. DOI: 10.1063/1.2916718.

[64] Frank Noé et al. 'Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations'. *Proceedings of the National Academy of Sciences* 106.45 (2009), pp. 19011–19016. DOI: 10.1073/pnas.0905466106.

[65] Nuria Plattner and Frank Noé. 'Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models'. *Nature communications* 6 (2015), p. 7653. DOI: 10.1038/ncomms8653.

[66] Riccardo Scalco and Amedeo Caflisch. 'Equilibrium distribution from distributed computing (simulations of protein folding)'. *The journal of physical chemistry. B* 115.19 (2011), pp. 6358–6365. DOI: 10.1021/jp2014918.

[67] Susanna Röblitz and Marcus Weber. 'Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification'. *Advances in Data Analysis and Classification* 7.2 (2013), pp. 147–179. DOI: 10.1007/s11634-013-0134-6.

[68] Diederik P. Kingma and Max Welling. 'Auto-Encoding Variational Bayes'. *The 2nd International Conference for Learning Representations, ICLR* (2013). URL: https://arxiv.org/pdf/1312.6114.

[69] Carl Doersch. 'Tutorial on Variational Autoencoders' (2016). URL: https://arxiv.org/pdf/1606.05908.

[70] Günter Klambauer et al. 'Self-Normalizing Neural Networks'. *Advances in Neural Information Processing Systems 30 (NIPS)* (2017). URL: https://arxiv.org/pdf/1706.02515.

[71] Diederik P. Kingma and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. *The 3rd International Conference for Learning Representations, San Diego, 2015* (2015). URL: https://arxiv.org/pdf/1412.6980.

[72] David L. Davies and Donald W. Bouldin. 'A Cluster Separation Measure'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (1979), pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[73] J. C. Dunn†. 'Well-Separated Clusters and Optimal Fuzzy Partitions'. *Journal of Cybernetics* 4.1 (1974), pp. 95–104. DOI: 10.1080/01969727408546059.

[74] Peter J. Rousseeuw. 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis'. *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

[75] Bradley Efron. 'Bootstrap Methods: Another Look at the Jackknife'. In: *Breakthroughs in statistics*. Ed. by Samuel Kotz. Springer series in statistics Perspectives in statistics. New York and Berlin: Springer, 1993, pp. 569–593. DOI: 10.1007/978-1-4612-4380-9_41.

[76] T. Nakatsu, H. Kato and J. Oda. 'Crystal structure of asparagine synthetase reveals a close evolutionary relationship to class II aminoacyl-tRNA synthetase'. *Nature structural biology* 5.1 (1998), pp. 15–19. DOI: 10.1038/nsb0198-15.

[77] Marcus Jäger et al. 'Structure-function-folding relationship in a WW domain'. *Proceedings of the National Academy of Sciences of the United States of America* 103.28 (2006), pp. 10648–10653. DOI: 10.1073/pnas.0600511103.

[78] Schrödinger, LLC. 'The PyMOL Molecular Graphics System, Version 1.8'. Nov. 2015.

[79] Hess. 'Similarities between principal components of protein dynamics and random diffusion'. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics* 62.6 Pt B (2000), pp. 8438–8448. DOI: 10.1103/physreve.62.8438.

[80] Berk Hess. 'Convergence of sampling in protein simulations'. *Physical review. E, Statistical, nonlinear, and soft matter physics* 65.3 Pt 1 (2002), p. 031910. DOI: 10.1103/PhysRevE.65.031910.

[81] Wolfgang Demtröder. *Experimentalphysik: 1: Mechanik und Wärme*. 7., neu bearb. und aktualisierte Aufl. Berlin and Heidelberg: Springer Spektrum, 2015. DOI: 10.1007/978-3-662-46415-1.

[82] Yuji Sugita and Yuko Okamoto. 'Replica-exchange molecular dynamics method for protein folding'. *Chemical Physics Letters* 314.1-2 (1999), pp. 141–151. DOI: 10.1016/S0009-2614(99)01123-9.

[83] Alessandro Barducci, Massimiliano Bonomi and Michele Parrinello. 'Metadynamics'. *WIREs Computational Molecular Science* 1.5 (2011), pp. 826–843. DOI: 10.1002/wcms.31.

[84] Rafael C. Bernardi, Marcelo C. R. Melo and Klaus Schulten. 'Enhanced sampling techniques in molecular dynamics simulations of biological systems'. *Biochimica et biophysica acta* 1850.5 (2015), pp. 872–877. DOI: 10.1016/j.bbagen.2014.10.019.

[85] Ulf Hensen, Oliver F. Lange and Helmut Grubmüller. 'Estimating absolute configurational entropies of macromolecules: the minimally coupled subspace approach'. *PloS one* 5.2 (2010), e9179. DOI: 10.1371/journal.pone.0009179.

[86] Andreas Mardt et al. 'VAMPnets for deep learning of molecular kinetics'. *Nature communications* 9.1 (2018), p. 5. DOI: 10.1038/s41467-017-02388-1.

[87] Carlos X. Hernández et al. 'Variational encoding of complex dynamics'. *Physical review. E* 97.6-1 (2018), p. 062412. DOI: 10.1103/PhysRevE.97.062412.

[88] Feliks Nüske et al. 'Variational Approach to Molecular Kinetics'. *Journal of chemical theory and computation* 10.4 (2014), pp. 1739–1752. DOI: 10.1021/ct4009156.

[89] Hao Wu and Frank Noé. 'Variational Approach for Learning Markov Processes from Time Series Data'. *Journal of Nonlinear Science* 30.1 (2020), pp. 23–66. DOI: 10.1007/s00332-019-09567-y.

6 Appendix

List A: PDB [37] codes of the proteins selected for this work.

11AS	1A3H	1AT0	1B04	1B5U	1B75	1B79	1B7Y	1BA3
1BFD	1BGW	1BHD	1BS2	1BSG	1C3G	1C3P	1CD5	1CEQ
1CQY	1DDG	1DJ0	1DTW	1DVG	1E6Z	1EHE	1ENH	1EO0
1EO9	1EP0	1EVL	1F17	1FAS	1FM7	1FQN	1G2R	1G6L
1GC7	1GH9	1GHH	1GSO	1H3L	1H8H	1HD8	1HZG	1I2T
1I39	1I6A	1ILW	1IMF	1IMT	1IMU	1IRX	1ITV	1IUH
1IUR	1IZM	1J22	1JBI	1JHF	1JI8	1JPU	1JR2	1JRM
1JW3	1K0S	1K3C	1K6K	1K8K	1KPT	1LB6	1LBV	1LFP
1M1L	1MW7	1MZG	1N2J	1N6Z	1NIJ	1NKG	1NO5	1NPR
1NYN	1O0W	1O99	1OAG	1OUO	1P74	1P7A	1P99	1PB6
1PBY	1PNO	1POZ	1PU1	1PV5	1PVE	1PVS	1PVT	1Q60
1QAU	1QAZ	1QPM	1QQH	1QW2	1R0D	1R6U	1RKI	1RL6
1RLH	1RLK	1RWC	1RYK	1RYU	1RZW	1S2J	1S2O	1S35
1S3A	1S4K	1S7E	1SGV	1SNO	1SRV	1SU6	1T3B	1TJN
1TLB	1TLQ	1TM9	1TSF	1U24	1U56	1U5U	1U61	1U84
1UDG	1UG2	1UJ8	1UK3	1UNE	1USG	1UW0	1V0F	1V7L
1V7O	1V9K	1V9V	1VAJ	1VCL	1VDH	1VK5	1VMG	1VQZ
1WB7	1WE8	1WEK	1WFT	1WFW	1WFY	1WGF	1WHB	1WHC
1WHK	1WHR	1WHZ	1WIX	1WIZ	1WJ5	1WJW	1WN9	1WOT
1WQ4	1WWR	1X7F	1X9B	1XD3	1XDN	1XHS	1XJH	1XN8
1XO8	1XQO	1XVI	1XWM	1YB3	1YEL	1YEZ	1YGY	1YS9
1YWU	1Z5B	2AHC	2BES	2F21	2FFM	2HBB	2MOB	2PGI
2PTH	4MAT							

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die von mir angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder sinngemäß aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht. Die Richtlinien zur Sicherung der guten wissenschaftlichen Praxis an der Universität Göttingen wurden von mir beachtet. Mir ist bewusst, dass bei Verstoß gegen diese Grundsätze die Prüfung mit nicht bestanden bewertet wird.

Göttingen, den 9. Juli 2021

Nicolai Kozlowski