TECHNISCHE UNIVERSITÄT BERLIN

Variational Tensor Approach for Approximating the Rare-Event Kinetics of Macromolecular Systems

Feliks Nüske **Reinhold Schneider**
Francesca Vitalini **Frank Noé**

Preprint 2015/30
Preprint-Reihe des Instituts für Mathematik
Technische Universität Berlin
`http://www.math.tu-berlin.de/preprints`

Preprint 2015/30 **November 2015**

Variational Tensor Approach for Approximating the Rare-Event Kinetics of Macromolecular Systems

Feliks Nüske, Reinhold Schneider, Francesca Vitalini, and Frank Noé

Freie Universität Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin

Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin and

*Freie Universität Berlin, Department of Chemistry, Takustr. 3, 14195 Berlin**

Abstract

Essential information about the stationary and slow kinetic properties of macromolecules is contained in the eigenvalues and eigenfunctions of the dynamical operator of the molecular dynamics. A recent variational formulation allows to optimally approximate these eigenvalues and eigenfunctions when a basis set for the eigenfunctions is provided. In this study, we propose that a suitable choice of basis functions is given by products of one-coordinate basis functions, which describe changes along internal molecular coordinates, such as dihedral angles or distances. A sparse tensor product approach is employed in order to avoid a combinatorial explosion of products, i.e. of the basis-set size. Our results suggest that the high-dimensional eigenfunctions can be well approximated with relatively small basis set sizes.

* feliks.nueske@fu-berlin.de; frank.noe@fu-berlin.de

I. INTRODUCTION

Molecular dynamics simulations (MD) have matured to be a standard tool to explore the conformations and conformational dynamics of biological macromolecules, such as proteins [1]. In MD simulations, the time integration of classical mechanics is combined with stochastic contributions, which ensure that certain thermodynamic quantities such as the temperature remain constant on average. For any atomic configuration $\mathbf{x}$ in the configuration space Ω, the classical force acting on the system is given by the gradient of an empirical potential energy function $V(\mathbf{x})$, which is the model for the molecular system under investigation. Such a potential energy function is usually the sum of many terms involving only a few atoms, typically depending on intramolecular distances ρ_{ij}, bond angles α_{ijk} and dihedral angles β_{ijkl}. A standard form of V is:

$$\begin{aligned} V(\mathbf{x}) = & \sum_{\text{bonds } (i,j)} V_{\text{bond}}(\rho_{ij}) \\ & + \sum_{\text{angles } (i,j,k)} V_{\text{angle}}(\alpha_{ijk}) \\ & + \sum_{\text{dihedrals } (i,j,k,l)} V_{\text{dih}}(\beta_{ijkl}) \\ & + \sum_{\text{nonbonded } (i,j)} V_{\text{nonbonded}}(\rho_{ij}) + V_{\text{other}}(\mathbf{x}). \end{aligned} \tag{1}$$

The nonbonded terms normally comprise the van-der-Waals and electrostatic interactions. The long-ranged part of the nonbonded forces, which might be computed through particle-field methods such as particle mesh Ewald, is included in $V_{\text{other}}(\mathbf{x})$.

The extraction of essential stationary and kinetic information from a set of trajectories $\{\mathbf{x}_t\}$, generated by MD simulations, has been an active field of research in recent years. A very powerful approach has been Markov modeling [2–7]: first, the sampled molecular conformations are clustered and the trajectories $\{\mathbf{x}_t\}$ are re-written into discrete time series. Second, the conditional transition probabilities between the clusters are estimated. These transition probabilities form a row-stochastic transition matrix best approximating the discretized dynamics. Software for the automatic construction and estimation of Markov models is available [8, 9]. Markov models have initiated a large amount of follow-up research, e.g. on the analysis of transition pathways through Markov models [10, 11], connecting molecular simulation and experimental observables via Markov

models [12–15], and applications to various molecular processes [10, 16–19]. A particularly interesting result from this line of research is that the essential information of the molecule's stationary and kinetic quantities is contained in the dominant eigenvalues and eigenfunctions of the dynamical operator of the molecular dynamics [2], and that the eigenvalues and eigenvectors of a Markov model transition matrix can be a good approximation thereof [20].

A more recent development has generalized this result. Instead of constructing a Markov model, one can define any set of basis functions that map molecular configurations to real values, and then try to approximate the eigenvalues and eigenfunctions of the MD operator by a linear combination of these basis functions [21]. This approach, named variational approach to conformation dynamics (VAC), may improve our ability to approximate and interpret the high-dimensional eigenfunctions. The construction of a Markov model requires a state space discretization that is typically obtained by some clustering routine. The discrete states obtained in this way lack a clear physical meaning in many cases. Within the more general variational approach, we can replace the discretization step by the use of functions that more faithfully describe configuration changes in molecules. We note that there is a close connection to quantum mechanics, where a craft of choosing suitable basis sets for modeling specific chemical transitions has evolved. However, it is yet unclear which choices of basis functions are suitable to model classical conformational changes of macromolecules. We were prompted to model such conformational changes in terms of the elementary internal coordinates ρ_{ij}, α_{ijk} and β_{ijkl}, as the potential in Eq. (1) depends on those, see [22] for some first results. However, these internal coordinates are coupled, thus we must allow our basis functions to represent couplings between internal coordinates in order to provide a good approximation of the high-dimensional eigenfunctions. In Ref. [23], a basis set for peptides with coupling between adjacent ϕ/ψ backbone angles was designed, and products between the basis functions were considered.

In general, this idea leads to the following approach: for all d elementary internal coordinates x_p, $p = 1, \ldots, d$ under consideration, we select n one-coordinate basis functions $f^p_{i_p}(x_p)$, $i_p = 1, \ldots, n$, defined on the single coordinate x_p. These functions should be chosen such as to be able to represent every significant variation along the single coordinates. A coupled basis set is then obtained by considering all possible products between one-coordinate basis functions:

$$\boldsymbol{\chi} = \left\{ f^1_{i_1}(x_1) f^2_{i_2}(x_2) \ldots f^d_{i_d}(x_d),\ i_1, i_2, \ldots, i_d = 1, \ldots, n \right\}. \tag{2}$$

Clearly, the size of the basis set Eq. (2) will combinatorially explode, as for n one-coordinate functions we have n^d possible products, making even the sheer enumeration of them unfeasible for relatively small molecules.

Consequently, we need a systematic method to determine a sparse representation of the eigenfunctions from the basis set Eq. (2). The high degree of redundancy among the set of intramolecular coordinates (for instance, among many distances) suggests that finding such a representation should be possible. Here we propose a new method based on the tensor-train (TT) format and its learning algorithm, the alternating linear scheme (ALS). Tensor-trains have been introduced by Refs. [24, 25]. However, they have been known in quantum physics as matrix product states since the 1990s, see e.g. Ref. [26]. A version of the ALS was introduced by Ref. [27] as the density matrix renormalization group algorithm, developed for the ground-state calculation of 1D spin chains. For a recent review on matrix product states and tensor product approximations in quantum physics, see Ref. [28] and references therein. It should also be noted that tensor-trains are special cases of hierarchical tensor representations as introduced by [29], also called tree tensor networks [28]. Please see the book [30] for a general overview of these concepts. A hierarchical tensor representation requires the choice of a dimension tree. One of the simplest and most straightforward choices is a linear and unbalanced tree, which gives rise to a tensor-train. In this work, we will stick to the TT-format, mainly because of its conceptual simplicity, and on the basis of tensor-trains and ALS having provided promising results for model systems [31].

The main contribution of this paper is to demonstrate the usefulness of the variational tensor approach for the analysis of data generated by MD simulations. To this end, after briefly reviewing the spectral theory of transfer operators and the variational approach in Sections II and III, we present some essential theory of tensor-trains in Section IV A, in a format which is suitable to our problem. In contrast to most of the previous applications of tensor product formats, the dynamical operator in our setting is only sampled by the simulation data, and no explicit low-rank representation of the operator is available. In Section IV B, we present a modification of the ALS which is suitable for our problem setting. The method is intuitive and appears to be useful in practical cases. We conclude by presenting the successful application of the method to MD simulations of two benchmark systems, deca-alanine and bovine pancreatic trypsin inhibitor (BPTI), for which reference solutions using Markov models are known. We also suggest a postprocessing method

that allows to identify a reduced subset of input coordinates which is representative for the slow dynamics. Its usefulness is confirmed by the applications.

II. THEORY

We first outline the basic assumptions made for our molecular dynamics (MD) implementation, following Ref. [6]. The MD simulation samples from a time- and space continuous, ergodic Markov process in a state space Ω. These dynamics have a unique stationary distribution π, given by the Boltzmann distribution:

$$\pi(x) \propto \exp\left(-\beta V(x)\right). \tag{3}$$

This means that independent of the starting point, every MD simulation that is sufficiently long will sample from this stationary distribution. Here, $\beta = (k_B T)^{-1}$ is the inverse temperature, k_B is the Boltzmann constant and T the temperature.

Another crucial ingredient for the system's description is the transition density function $p(x, y; \tau)$, which describes the conditional probability of the system travelling from x to y over a finite time step $\tau > 0$, also called the lag time. Although the transition density is not known to us, we can assume that it satisfies

$$\pi(x)p(x, y; \tau) = \pi(y)p(y, x; \tau) \quad \forall x, y \in \Omega, \tag{4}$$

which is the "detailed balance" condition. Equation (4) is physically motivated: It prohibits the existence of a preferred direction in the system, which could be used to produce work. Of course it must be enforced by the molecular dynamics integrator in order to hold in the simulation. Using the transfer density, we can understand how ensembles of trajectories evolve in time: If an ensemble of trajectories is started according to a distribution p_0 at time $t = 0$, then the corresponding distribution p_τ at time τ is given by

$$p_\tau(y) = \mathcal{P}(\tau)p_0(y) \tag{5}$$

$$= \int_\Omega p(x, y; \tau)\, p_0(x)\, \mathrm{dx}. \tag{6}$$

This defines a linear integral operator, the propagator $\mathcal{P}(\tau)$. Let us summarize the main properties of the propagator, see again Ref. [6] for details. If we introduce the weighted scalar product

$$\langle u, v\rangle_{\pi^{-1}} = \int_\Omega u(x)\, v(x)\, \pi^{-1}(x)\, \mathrm{dx}, \tag{7}$$

the propagator is bounded and self-adjoint (due to Eq. (4)) on the space $L^2_{\pi^{-1}}$ of functions u that satisfy $\langle u, u\rangle_{\pi^{-1}} < \infty$. It can be shown that its largest eigenvalue is $\lambda_1 = 1$, whereas all remaining eigenvalues are strictly smaller than one in absolute value. Moreover, there is a number M of dominant positive eigenvalues:

$$1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_M > 0. \tag{8}$$

The remaining spectrum is contained in an interval $[-R,\, R]$, with $R < \lambda_M$. The eigenfunction corresponding to the largest eigenvalue $\lambda_1 = 1$ is the stationary density π.

The dominant eigenvalues λ_m and corresponding eigenfunctions ϕ_m are the key to understand the complex dynamics of the molecule. All eigenvalues except the first decay exponentially with the lag time τ, i.e.

$$\lambda_m(\tau) = e^{-\kappa_m \tau},\ m = 2, \ldots, M, \tag{9}$$

with some relaxation rates $\kappa_m > 0$. Using the eigenfunctions and eigenvalues, the action of the propagator on a probability density p_0 can be written as

$$\mathcal{P}(\tau)p_0 = \sum_{m=1}^{M} e^{-\kappa_m \tau} \langle \phi_m,\, p_0\rangle_{\pi^{-1}}\, \phi_m + \mathcal{P}_{\mathrm{fast}}\, p_0. \tag{10}$$

In most systems that we are interested in, the contribution of $\mathcal{P}_{\mathrm{fast}}$ vanishes quickly with the lag time. This implies that for large enough τ, only a finite number M of the terms in the expansion

(10) are still present, and the operator's action can be understood in terms of only finitely many processes:

$$\mathcal{P}(\tau)p_0 \approx \sum_{m=1}^{M} e^{-\kappa_m \tau} \langle \phi_m,\, p_0 \rangle_{\pi^{-1}}\, \phi_m. \tag{11}$$

Each of the dominant eigenfunctions $\phi_2, \ldots, \phi_m$ typically carries the same sign on some specific regions of the state space, but changes its sign between those regions. In this way, the function encodes parts of the state space where the system generally remains for a long time, while it rarely transitions between them [2]. This concept is known as metastability and is a typical feature of biomolecules, where the metastable regions are frequently associated with biological function of the molecule, e.g. the ability / inability to bind to a binding partner. Thus, each term in Eq. (11) corresponds to a relaxation process that transports probability between the metastable regions in order to equilibrate the system towards the stationary distribution. The relaxation time of each process is called the implied timescale and can be computed from the dominant eigenvalues:

$$t_m = \tfrac{1}{\kappa_m} = -\frac{\tau}{\log(\lambda_m(\tau))}. \tag{12}$$

An illustration of this concept is shown in Figure 1 for a very small toy system, the alanine dipeptide (Ac-Ala-NHMe). This molecule consists of a single amino acid, alanine, capped by two protection groups mimicking the behaviour of the amino acid in a chain, like in a protein. It has been used as a test case for many studies in recent years, e.g. Ref. [32], because it is easy to produce sufficient simulation data and consequently, its slow dynamics are very well-understood. Given this theoretical assessment, we now describe how to practically approximate the eigenvalues and eigenfunctions.

III. DISCRETIZATION

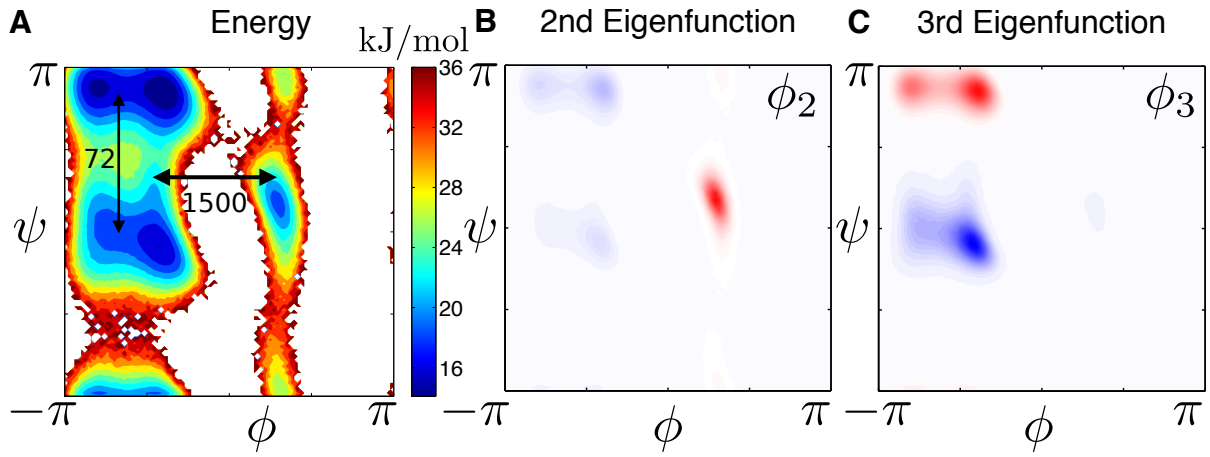

Figure 1. Conformation dynamics of the alanine dipeptide. In panel A, we show the energy of the system as a function of the two dihedral angles ϕ and ψ (not to be confused with the eigenfunctions) in units of kJ/mol. There are three minima of the energy landscape (blue color) in the upper left, central left, and central right region, where the system remains for most of the time. The two slow transitions are indicated by black arrows: the slowest is from the left to the right part of the plane, while the second slowest transition occurs between the upper left and central left minimum. The corresponding relaxation timescales t_2, t_3 in units of ps are printed next to the arrows. These two transitions are encoded in the sign structure of the second and third propagator eigenfunctions ϕ_2, ϕ_3, shown in panels B and C. It can be seen that ϕ_2 changes sign (red: positive, blue: negative) in the ϕ-direction, i.e. the slowest transition with timescale t_2 is associated with a switch of the ϕ-angle. The intensities are not equal on the left and on the right as the population of the two left minima is much higher compared to the right. In the same way, the second slowest transition can be recognized from the sign structure of ϕ_3. This figure has been adapted with permission from [22, Figure 4], copyright 2014, American Chemical Society.

A. Method of Linear Variation

In order to perform the approximation of eigenfunctions in practice, we consider a different operator, the transfer operator

$$\mathcal{T}(\tau)u(y) = \frac{1}{\pi(y)} \int_\Omega p(x, y; \tau)\, \pi(x)\, u(x)\, \mathrm{dx}. \tag{13}$$

The transfer operator is equivalent to the propagator in the sense that it possesses the same eigenvalues λ_m, and its corresponding eigenfunctions ψ_m can be computed from the propagator eigenfunctions ϕ_m via

$$\psi_m(x) = \pi^{-1}(x)\phi_m(x). \tag{14}$$

In particular, the sign structure of the ψ_m is identical to that of the ϕ_m. The transfer operator eigenfunctions ψ_m are orthonormal with respect to the scalar product

$$\langle u, v \rangle_\pi = \int_\Omega u(x)\, v(x)\, \pi(x)\, \mathrm{dx}, \tag{15}$$

and the operator is self-adjoint with respect to this inner product. We have the traditional variational formulation [33]:

$$\sum_{m=1}^{M} \lambda_m = \sup_{u_1,\ldots,u_M} \sum_{m=1}^{M} \langle \mathcal{T}(\tau) u_m, u_m \rangle_\pi, \tag{16}$$

$$\langle u_m, u_{m'} \rangle_\pi = \delta_{m,m'}. \tag{17}$$

The sum of the first M exact eigenvalues maximizes the partial trace $\sum_{m=1}^{M} \langle \mathcal{T}(\tau) u_m, u_m \rangle_\pi$, where the u_m are a set of M orthonormal functions. Equality is attained exactly for the true eigenfunctions $\psi_1, \ldots, \psi_M$. Our goal is to obtain an approximation of the true eigenfunctions from a finite space of trial functions $\chi_1, \ldots, \chi_N$. Therefore, one can restrict the above problem to the space spanned by the functions χ_i. It is also a well-known result [34] that the right-hand side of Eq. (16) is then maximized, subject to the constraint (17), by the functions

$$\hat{\psi}_m = \sum_{i=1}^{N} \mathbf{V}_{im} \chi_i. \tag{18}$$

The columns of the matrix $\mathbf{V}$ correspond to the first M eigenvectors of the generalized eigenvalue problem

$$\mathbf{C}^\tau \mathbf{V}(:, m) = \hat{\lambda}_m \mathbf{C}^0 \mathbf{V}(:, m),\ m = 1, \ldots, M, \tag{19}$$

$$c_{ij}^\tau = \langle \mathcal{T}(\tau)\chi_i, \chi_j \rangle_\pi, \tag{20}$$

$$c_{ij}^0 = \langle \chi_i, \chi_j \rangle_\pi. \tag{21}$$

The eigenvalues $\hat{\lambda}_m$ can be used as estimates of the true eigenvalues λ_m, and the functions $\hat{\psi}_m$ serve as approximations of the true eigenfunctions. We have that $\hat{\lambda}_m = \langle \mathcal{T}(\tau)\hat{\psi}_m, \hat{\psi}_m \rangle_\pi$, and it follows from Eq. (16) that:

$$\sum_{m=1}^{M} \hat{\lambda}_m \leq \sum_{m=1}^{M} \lambda_m. \tag{22}$$

The variational principle Eq. (22) also holds for the individual eigenvalues $\hat{\lambda}_m$, see e.g. Ref. [21]:

$$\hat{\lambda}_m < \lambda_m, \text{if} \hat{\psi}_m \neq \psi_m, \tag{23}$$

$$\hat{\lambda}_m = \lambda_m, \text{if} \hat{\psi}_m = \psi_m \quad m = 1, ..., M. \tag{24}$$

The matrix entries in Eqs. (20) and (21) are not exactly computable, because the transfer density defining $\mathcal{T}(\tau)$ is not known in an analytically practical form, and the integration spaces are very high-dimensional. An exception are models that directly parametrize the transfer density in a way that allows us to evaluate Eqs. (20) and (21), see e.g. Ref. [35].

However, as pointed out in Refs. [21, 22], we can still estimate the matrices $\mathbf{C}^0$ and $\mathbf{C}^\tau$ as correlation matrices between basis functions, as their entries are spatial expectation values. From a sufficiently long realization of the process, they can be estimated by time averages as follows:

$$c_{ij}^{\tau} \approx \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \chi_i(\mathbf{x}_t)\chi_j(\mathbf{x}_{t+\tau}), \tag{25}$$

$$c_{ij}^{0} \approx \frac{1}{T} \sum_{t=1}^{T} \chi_i(\mathbf{x}_t)\chi_j(\mathbf{x}_t), \tag{26}$$

where T is the total length of the realization. This is not true for the corresponding matrix approximation of the propagator $\mathcal{P}(\tau)$, as the matrix elements cannot be interpreted as correlations. It is worth noting that Markov State Models (MSMs) are a special case of the above formulation for the choice of characteristic functions of sets as basis functions [21, 22]. MSMs have the additional benefit that the estimators for Eqs. (25) and (26) are conditional estimators, and this allows us to use short trajectories that are not in global equilibrium [6]. This important property can also be achieved for other basis sets that are probability densities, such as Gaussian distributions [35]. For arbitrary basis functions, the estimators Eqs. (25) and (26) are incorrect if the data is not in global equilibrium. It is an open research question if these estimates can be used nevertheless.

B. Coordinates and Basis Sets

In order to apply the above approach, it is still necessary to identify the set of functions or a sequence of sets of functions to be used. This problem has in fact three aspects, as we need to select the coordinates that serve as domains of these functions, the type of functions, and the way that functions depending on different coordinates are being combined.

Finding informative input coordinates is an ongoing research problem, see e.g. [8, 36, 37]. However, this task is not the objective of the present study. In addition to that, we must decide what type of basis functions we use on the set of input coordinates, and how to choose their parameters. In [22], we have used a set of one-coordinate basis functions defined on the individual dihedral angles (but no tensor products of them) for small peptides. The one-coordinate functions themselves were chosen as Gaussian functions, their parameters were selected for each dihedral angle separately. Let us emphasize that the choice of Gaussian basis functions was just for the sake of illustration - other basis sets might have worked equally well. In fact, it would be desirable to use basis functions which carry some chemical or physical information. They might, for instance, encode

the torsional rotamer or whether a hydrogen bond is formed or dissociated. This is another open line of research, for some first results in this direction, see Ref. [23].

In this work, we are focusing on the coupling problem: In order to correctly model the dynamics of large systems where coordinates are coupled, we should use basis functions that are products of one-coordinate basis functions. Assuming that there are d input coordinates labeled $x_1, x_2, \ldots, x_d$, and for each coordinate x_p we have n one-coordinate basis functions $f^p_{i_p}(x_p)$, where $p = 1, \ldots, d$ and $i_p = 1, \ldots n$ (the theory can trivially deal with p-dependent n, but for simplicity of notation we assume a constant n here). For practical reasons, we assume that the first one-coordinate basis function is the constant, $f^p_1(x_p) \equiv 1$, although this is not needed for most of the theory. Then we try to approximate each eigenfunction ψ_m, $m = 1, 2, \ldots$ by a function $\hat{\psi}_m$ that is a linear combination of all possible products of the $f^p_{i_p}$:

$$\hat{\psi}_m(x_1, \ldots, x_d) = \sum_{i_1,\ldots,i_d} U_m(i_1, \ldots, i_d)\, f^1_{i_1}(x_1) \ldots f^d_{i_d}(x_d). \tag{27}$$

The basis functions for the variational approach are the products $f^1_{i_1}(x_1) \ldots f^d_{i_d}(x_d)$, and U_m is a d-dimensional array (tensor) containing the expansion coefficients of all these products. As we can immediately see, the number of basis functions used in this expansion is n^d. This number becomes impossible to cope with even for small n and moderate d, not to mention the evaluation of the correlation matrices in Eqs. (25) and (26) using long trajectories. However, our experience and the high degree of redundancy contained in intromolecular coordinates suggest that a small selection of these product functions should be sufficient to produce essentially the same results. Let us illustrate this by another example, a capped dimer of the two amino acids valine and alanine (Ac-Val-Ala-NHMe). Here, we have two pairs of dihedral angles, the dimension thus becomes $d = 4$. Since the coordinates are periodic angles, we used the real Fourier basis of sine and cosine functions. Setting $n = 5$, the full product basis comprises $5^4 = 625$ functions. In Figure 2A, we check the accuracy of the model by comparing the two slowest implied timescales t_2, t_3 to those obtained from a reference Markov model. This model is obtained by discretizing the dihedral plane of every residue into three states which were chosen according to known dynamics of the monomers, resulting in a total of $3^2 = 9$ states, see Ref. [23]. Both models perform comparably well. Clearly the Markov model is much more efficient, but its construction requires a priori knowledge of the

peptide dynamics that is not easily transferred to larger systems. Figure 2B) shows the cumulative sum of the squared coefficients of the estimated second eigenfunction $\hat{\psi}_2$ from the product basis. The coefficients were computed after transforming the product basis into an orthonormal basis with respect to the π-weighted inner product Eq. (15). We observe that only a small part of the 625 basis functions contribute with a high coefficient compared to all others. We conclude that it should be possible to find a much smaller subspace of the full product space and end up with essentially the same result. The efficient search for this subspace is the topic of the next section.

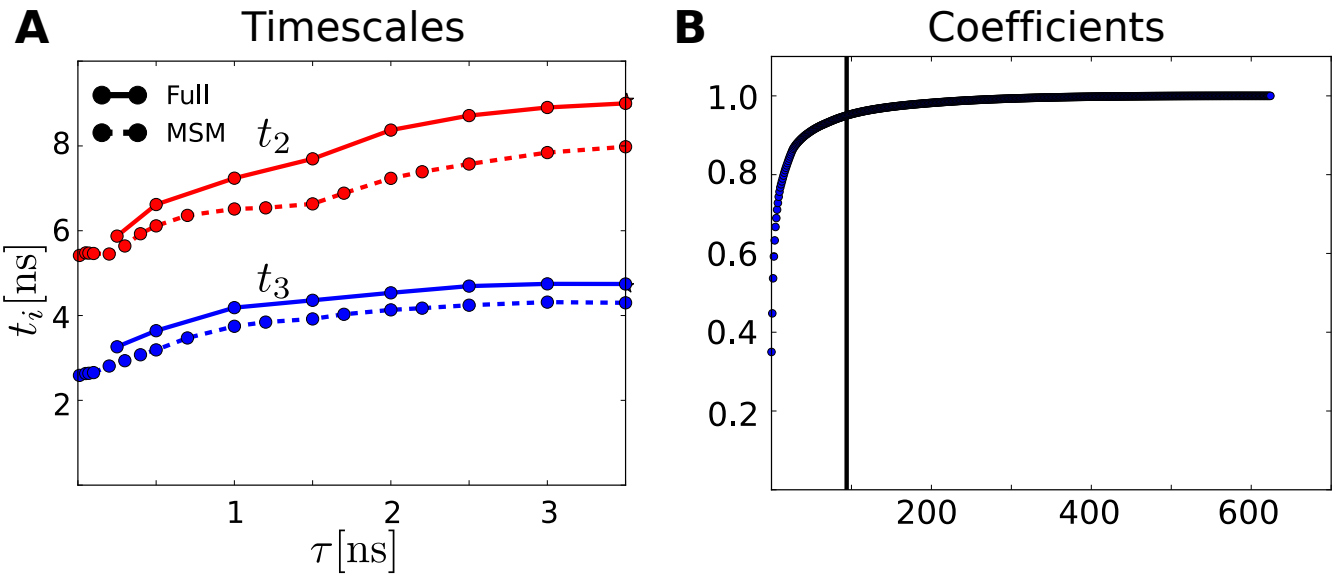

Figure 2. Illustration of low-dimensional subspaces carrying the relevant information for the dimer Ac-Val-Ala-NHMe: Panel A shows the two slowest implied timescales t_2, t_3, in red and blue, estimated by two different models: A reference MSM (dashed line, 9 states) and the full product expansion Eq. (27) (solid line, $n = 5$, 625 basis functions). Both models perform comparably well. In panel B, we present a cumulative plot of the squared expansion coefficients of the second eigenfunction $\hat{\psi}_2$, as estimated by the full product approach, expressed in an orthonormal basis w.r.t. the weighted inner product Eq. (15). It takes about 90 basis functions to reproduce 95 percent of the norm, as indicated by the black vertical line.

IV. TENSOR PRODUCT APPROXIMATIONS

A. Tensor-Train-Format

The problem of finding a computationally feasible approximation to a high-dimensional representation like Eq. (27) occurs across many fields, and significant progress has been made in recent years. Out of all the different approaches that have been suggested, we choose to present and use the tensor-train (TT) format, which has been introduced in [24, 25].

A function in TT-format still possesses a high-dimensional representation like Eq. (27), but the coefficient array U_m has a special structure as in Eq. (29) below [25]:

$$\hat{\psi}_m = \sum_{i_1,\ldots,i_d} U_m(i_1,\ldots,i_d)\, f_{i_1}^1(x_1)\ldots f_{i_d}^d(x_d) \tag{28}$$

$$= \sum_{i_1,\ldots,i_d} \left[\sum_{k_1=1}^{r_1} \ldots \sum_{k_{d-1}=1}^{r_{d-1}} U_1(i_1,k_1) U_2(k_1,i_2,k_2)\ldots U_d(k_{d-1},i_d) \right] f_{i_1}^1(x_1)\ldots f_{i_d}^d(x_d). \tag{29}$$

Here, $U_1 \in \mathbb{R}^{n\times r_1}$, $U_d \in \mathbb{R}^{r_{d-1}\times n}$ are matrices and $U_p \in \mathbb{R}^{r_{p-1}\times n\times r_p}$, $p = 2,\ldots,d-1$, are three-dimensional arrays. Consequently, for every choice of $i_1,\ldots,i_d$, the arrays U_1 and U_d turn into vectors $U_1(i_1)$, $U_d(i_d)$, whereas all other arrays $U_2,\ldots,U_{d-1}$ become matrices $U_2(i_2),\ldots,U_{d-1}(i_{d-1})$, and the coefficient $U_m(i_1,\ldots,i_d)$ can be computed by a repeated matrix-vector multiplication:

$$U_m(i_1,\ldots,i_d) = U_1(i_1)U_2(i_2)\ldots U_d(i_d). \tag{30}$$

Thus, only the arrays $U_1,\ldots,U_d$ need to be stored, and the number of parameters in these arrays is linear in the dimension d, see again Ref. [25].

The intuition behind this representation is that only limited information is passed on from one variable to the next in the sequence $x_1,\ldots,x_d$. To see this, consider the case $d = 4$, and re-order Eq. (29) as follows:

$$\hat{\psi}_m = \sum_{k_1,i_2,k_2} U_2(\mathbf{k_1}, i_2, \mathbf{k_2})\, f_{i_2}^2(x_2)\cdot \tag{31}$$

$$\left[\sum_{i_1} U_1(i_1, \mathbf{k_1}) f^1_{i_1}(x_1)\right] \cdot$$
$$\left[\sum_{i_3,i_4}\sum_{k_3} U_3(\mathbf{k_2}, i_3, k_3) U_4(k_3, i_4)\, f^3_{i_3}(x_3) \cdot f^4_{i_4}(x_4)\right]$$
$$= \sum_{k_1,i_2,k_2} U_2(\mathbf{k_1}, i_2, \mathbf{k_2})\, f^2_{i_2}(x_2) \cdot g^2_{\mathbf{k_1}}(x_1) \cdot h^2_{\mathbf{k_2}}(x_3, x_4). \tag{32}$$

The expressions shown in brackets in Eq. (31) contain exactly one free index k_1 and k_2, respectively, indicated by the bold-face letter. Thus, it makes sense to define functions $g^2_{k_1}$, $h^2_{k_2}$ by these expressions, which leads us to the representation in Eq. (32). The meaning of Eq. (32) is that the function $\hat{\psi}_m$ is represented by a linear combination of basis functions which can be separated into three parts: each basis function is a product of a function $f^2_{i_2}$ depending on the variable x_2, a function $g^2_{k_1}$ which depends on all variables up to x_2, and another function $h^2_{k_2}$ which depends on all unknowns following x_2. Thus, the information about all coordinates up to x_2 is encoded into a limited number of functions, and so is the information about all coordinates following x_2. The representation in Eq. (31) corresponds to panel B in Fig. 3. However, this is not the only way to re-order Eq. (29), as there are d equivalent ways to do so. All of these different re-orderings for the case $d = 4$ are displayed in the remaining parts of Fig. 3. In the general case, the re-ordering centered around coordinate x_p is given by

$$\hat{\psi}_m = \sum_{k_{p-1},i_p,k_p} U_p(\mathbf{k_{p-1}}, i_p, \mathbf{k_p})\, f^p_{i_p}(x_p) \cdot \tag{33}$$
$$\left[\sum_{i_1,\ldots,i_{p-1}}\sum_{k_1,\ldots,k_{p-2}} U_1(i_1,k_1)\ldots U_{p-1}(k_{p-2}, i_{p-1}, \mathbf{k_{p-1}}) f^1_{i_1}(x_1)\ldots f^{p-1}_{i_{p-1}}(x_{p-1})\right] \cdot$$
$$\left[\sum_{i_{p+1},\ldots,i_d}\sum_{k_{p+1},\ldots,k_{d-1}} U_{p+1}(\mathbf{k_p}, i_{p+1}, k_{p+1})\ldots U_d(k_{d-1}, i_d) f^{p+1}_{i_{p+1}}(x_{p+1})\ldots f^d_{i_d}(x_d)\right]$$
$$= \sum_{k_{p-1},i_p,k_p} U_p(\mathbf{k_{p-1}}, i_p, \mathbf{k_p})\, f^p_{i_p}(x_p) \cdot g^p_{\mathbf{k_{p-1}}}(x_1,\ldots,x_{p-1}) \cdot h^p_{\mathbf{k_p}}(x_{p+1},\ldots,x_d). \tag{34}$$

The underlying principle is the same: The information about the variables $x_1, \ldots, x_{p-1}$ is encoded into r_{p-1} functions $g^p_{k_{p-1}}$, which we call the left *interfaces* at position p. Also, the information

about the variables $x_{p+1}, \ldots, x_d$ is contained in r_p functions $h^p_{k_p}$, called right interfaces at p. The numbers $r_1, \ldots, r_{d-1}$ are called the *ranks* of the tensor-train. Furthermore, we note for later use that the interfaces satisfy the recursive relations

$$g^{p+1}_{k_p} = \sum_{k_{p-1}, i_p} U_p(k_{p-1}, i_p, k_p) g^p_{k_{p-1}} f^p_{i_p}, \tag{35}$$

$$h^{p-1}_{k_{p-1}} = \sum_{i_p, k_p} U_p(k_{p-1}, i_p, k_p) f^p_{i_p} h^p_{k_p}. \tag{36}$$

B. Alternating Linear Scheme

In order to make use of the tensor-train-format in practice, we need a method to determine the optimal components U_p, and a way to parametrize multiple eigenfunctions $\hat{\psi}_m$. To this end, we build on two major developments in the field of tensor-trains: first, the alternating linear scheme (ALS), which is an iterative learning algorithm that arises naturally from the TT-format, see Ref. [31]. Second, the block-TT-format from Refs. [38, 39], which is a modification of tensor-trains allowing for the simultaneous approximation of multiple functions using almost the same number of parameters. These concepts have led us to the algorithm outlined below. We present our optimization procedure as we have used it in the applications, and comment on its relation to the standard methods in the literature in Appendix B.

The idea of alternating optimization is that in each iteration step, we attempt to update one component U_p while keeping all others fixed. Starting from some initial guess for all U_p, the method will first update U_1 while $U_2, \ldots, U_d$ are fixed, then it will update U_2 with $U_1, U_3, \ldots, U_d$ fixed, and so on, until U_d is optimized. After completing this so called *forward sweep,* it will proceed backwards along the sequence of variables, which is called the *backward sweep.* This can be repeated until some convergence criterion is satisfied.

As outlined in the previous section, the component U_p can be read in two different ways: Either it is meant to optimally encode the information about all coordinates up to position p into r_p

left interfaces $g^{p+1}_{k_p}$, or to encode the information about all coordinates $x_p, \ldots, x_d$ into r_{p-1} right interfaces $h^{p-1}_{k_{p-1}}$. We will focus on the first reading during the forward sweep of the optimization, and on the second during the backward sweep. Consider the forward sweep case and assume that we attempt to optimize component U_p while all others are fixed. Following Sec. III A, and recalling the recursive definition Eq. (35), the optimal left interfaces $g^{p+1}_{k_p}$ would be the linear combinations

$$g^{p+1}_{k_p}(U_p) = \sum_{k_{p-1}, i_p} U_p(k_{p-1}, i_p, k_p) g^p_{k_{p-1}} f^p_{i_p}, \tag{37}$$

that maximize the eigenvalue sum

$$L_p(U_p) = \sum_{m=1}^{M} \hat{\lambda}_m(U_p) \tag{38}$$

resulting from the generalized eigenvalue problem Eq. (19) for the basis

$$g^{p+1}_{k_p}(U_p) f^{p+1}_{i_{p+1}} f^{p+2}_{i_{p+2}} \ldots f^d_{i_d}, \tag{39}$$

as it combines limited information about the first p coordinates with all possible basis functions of the remaining ones. As this problem is not tractable, we use the information we have already computed, and determine the interfaces $g^{p+1}_{k_p}$ which maximize the sum Eq. (38) for the reduced basis

$$g^{p+1}_{k_p}(U_p) f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, \tag{40}$$

see Fig. 4 for an illustration. This trick is inspired by the MALS [31] and the original DMRG algorithm.

Let us touch on the most important points of this optimization problem. First, we can set up a numerical optimization method for Eq. (38) if r_p is fixed, please see App. C for an explanation. Therefore, we sequentially determine the optimal component U_p for increasing values of the rank

r_p, and accept U_p as the solution if the eigenvalue sum $L_p(U_p)$ matches a reference value L_{ref} up to a tolerance ϵ_{rank}. If accepted, U_p becomes the new p-th component, the functions $g_{k_p}^{p+1}(U_p)$ become the new left interfaces at position $p+1$ and r_p is the new rank. Otherwise, r_p is increased by one and the above optimization is repeated. The reference L_{ref} is obtained as follows: As a first step, we always evaluate a four-fold product basis defined by the functions

$$g_{k_{p-1}}^{p} f_{i_p}^{p} f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1}, \tag{41}$$

and solve the generalized eigenvalue problem Eq. (19) for this basis. We compute the dominant eigenvalue sum resulting from this problem,

$$L_p = \sum_{m=1}^{M} \hat{\lambda}_m^{p,p+1}. \tag{42}$$

The variational principle Eq. (16) implies that for any U_p, the eigenvalue sum $L_p(U_p)$ is bounded from above by L_p. Thus, we keep track of the maximal value obtained for L_p during the entire optimization process, and store this maximum as the reference L_{ref}. Second, we enforce the first interface function g_1^{p+1} to be the constant function. This constraint ensures that the largest eigenvalue $\hat{\lambda}_1(U_p)$ is always equal to one, which turned out to be an important stabilization of the method. Third, the full optimization is considered converged if all of the objective functions $L_p(U_p)$ from two subsequent forward and backward sweeps do not differ by more than a tolerance ϵ_{iter}. A summary of the complete method is given in Algorithm 1.

V. RESULTS

In this section, we present two examples for the approximation of dominant eigenfunctions of molecular systems in the tensor-train-format. The first is the ten residue peptide deca-alanine (Ala_{10}), the second is the 58 residue protein BPTI. Equilibrium trajectories that are orders of magnitude longer than the slowest relaxation timescales are available for both of these systems.

The ALS-optimization is initialized as being completely uninformed, we set all ranks $r_p = 1$ and prepare the components U_p to parametrize just the constant function. We choose the rank acceptance threshold as $\epsilon_{\text{rank}} = 0.995$ and the overall stopping criterion as $\epsilon_{\text{iter}} = 0.01$. Both of these choices are based on our experience with the method so far, and a more systematic or automatic choice of parameters will be subject of further research. The setting for ϵ_{rank} ensures that no important information is lost along the course of the iteration. The setting for ϵ_{iter} reflects the general level of accuracy that we can achieve for the eigenvalues obtained from the analysis of MD data, based on the general experience we have.

Our analysis of the examples consists of four steps. First, we monitor the slowest implied timescale t_2 over the course of the optimization and compare it to reference values. Second, we analyse the structural transition encoded in the slowest eigenfunction $\hat{\psi}_2$. To this end, we evaluate the eigenfunction at all frames and histogram the resulting time series. Following the theory in Sec. II, we expect to find peaks of the population corresponding to the most negative and most positive values attained by the eigenfunction. As these peaks should correspond to metastable states, we extract representative structures for each of them in order to determine the structural transition described by the eigenfunction. Third, we attempt to identify coordinates which are relevant for the slow dynamics. To this end, we solve the following problem after every iteration step (we illustrate the problem for the forward sweep again, it works analogously for the backward sweep): after the new interface functions $g_{k_p}^{p+1}$ have been determined, we compute the best approximation to these functions in the least squares sense from the previous interfaces $g_{k_{p-1}}^{p}$ only, leaving out the one-coordinate basis for coordinate p. We record the average approximation error $E(p)$ for all of the new interface functions as a measure for the information contained in the basis at position p, see App. D for the details. Once the main iteration is completed, we re-run the ALS-iteration using only those coordinates p which satisfy that $E(p)$ is greater than a certain cutoff, and repeat this for various choices of the cutoff. By this procedure, we attempt to find a reduced set of coordinates which allows us to build an equally good model as the full one.

A. Deca-Alanine

Ala_{10} simply consists of ten residues of alanine in a chain. We use six equilibrium simulations of 500 ns each, accumulating to 3 μs total simulation time. The simulations were produced at temperature 300 K using the GROMACS 4.5.5 simulation package, the Amber03 force field and the TIP3P water model. See Appendix A for details.

The input coordinates used for this system are $d = 16$ backbone dihedral angles from the eight internal residues of the chain. We left out the two outermost residues as the chain was not capped in the simulation, increasing the flexibility of the outer residues. Our set of one-coordinate basis function used for each dihedral consisted of the first $n = 7$ real Fourier (sine and cosine) waves. The lag time used for our analysis was $\tau = 2$ ns. We can compare our results to a reference Markov state model, see e.g. Ref. [22]. This model was constructed by partitioning the Ramachandran plane of each pair of backbone dihedrals into three boxes corresponding to the minima of the single amino acid dynamics, see Fig. 1A. The Markov states were then defined by all combinations of these boxes, the total number of states is thus $8^3 = 6561$. It is found that the slowest dynamical process in the system is the formation of a helix and occurs at an implied timescale $t_2 \approx 7.5$-8ns.

Fig. 5A shows that the implied timescale t_2 as estimated by our model reaches the correct regime over the first forward sweep, then corrects slightly along the backward sweep, and remains more or less constant afterwards. Panel B displays the relative histogram of the second estimated eigenfunction $\hat{\psi}_2$ over all data points of the MD trajectory. We can identify a number of peaks of the population, of which we select the two outermost ones (around -1.3 ± 0.3 and 1.6 ± 0.2) to analyse the slow transition. An overlay of 200 random structures from each of these peaks confirms that the eigenfunction $\hat{\psi}_2$ encodes the transition from an extended structure to a helix, as expected. The final values of the least squares approximation error $E(p)$ (thus resulting from the final backward sweep) are shown in panel C. It can be observed that five interior ψ-angles from the chain display the largest least squares error, indicating that these coordinates are important. This is consistent with the slowest process being the formation of a helix, and is strengthened further by the analysis shown in panel D. Here, we find that these five coordinates allow us to build a model which equals the quality of the full model.

B. BPTI

We also study the 1.05 ms folded-state simulation of the 58-residue protein BPTI produced on the Anton supercomputer and provided by D.E. Shaw research [40]. This large dataset has become a benchmark system used in numerous studies in recent years. The slowest structural transition included in the C_α dynamics has been identified by other kinetic models to be on a timescale $t_2 \approx 40\,\mu$s, see Ref. [8, 41] for details.

The coordinates used in order to apply our method are the distances between all C_α atoms in the system which are at least three residues apart. For each distance, we construct a minimal basis set consisting of only $n = 2$ functions: The first is the constant, while the second is a smooth switching function indicating whether a contact between two C_α atoms has formed or not:

$$f_2^p(x_p) = \frac{1 - (x_p/r_0)^{64}}{1 - (x_p/r_0)^{96}}, \tag{43}$$

where x_p is the C_α distance under consideration and $r_0 = 0.7\,$nm is an empirically obtained cutoff distance. The function is mostly equal to one for $x_p < r_0$, indicating that a contact between the two atoms has formed, while it is mostly zero for $x_p > r_0$, thus indicating that the contact is broken. The function smoothly transitions between one and zero in a small neighborhood of r_0. With this basis, it is easy to reduce the number of input coordinates by checking if a contact has at least once transitioned from the formed to the broken state or vice versa, and only using those contacts while leaving out all others. For the given data set, this preprocessing reduces the number of contacts from initally around 1500 to $d = 258$. Still, this system is a lot larger than the previous one. We conduct our analysis at lag time $\tau = 5\,\mu$s.

Figure 6A shows that again, the second implied timescale t_2 rises to the appropriate regime over the course of the first forward sweep, improves further during the first backward sweep, and changes only slightly afterwards. The histogram of the data over the estimated second eigenfunction $\hat{\psi}_2$ displays two clearly distinguishable peaks at its extremal values (around -0.2 ± 0.5 and 6.5 ± 1.0). A set of 200 molecular structures extracted from these peaks confirm that $\hat{\psi}_2$ encodes the structural transition as it was determined previously, namely a re-folding of the loop on the N-terminal side of the backbone [8, 41]. The results of the least squares approximations are not as clear as in the previous example. It is apparent from Fig. 6C that more than 100 of the coordinates at the

end of the sequence are identified as completely unimportant, with $E(p) \approx 0$. This finding is in agreement with the fact that the part of the chain near the C-terminus is not involved in the slow transition. For the remaining 140 coordinates, $E(p)$ varies between 10^0 and 10^{-6}, but there is no obvious gap or cutoff which separates the important from the unimportant coordinates. However, such a cutoff can be determined by building various reduced models. We can conclude from Fig. 6D that choosing the cutoff as $E(p) \geq 10^{-3}$, we can determine a set of 58 coordinates which are sufficient to build a reduced model of the same quality as the full model, while using an even higher cutoff entails loss of information.

VI. CONCLUSIONS

We have proposed a new approach to approximate eigenvalues and eigenfunctions of molecular dynamics operators by products of one-coordinate basis functions. The one-coordinate basis functions are used to indicate local changes in internal coordinates of the molecular system, such as rotamer changes in the dihedrals, or formation / dissociation of contacts in the distances. Combining these one-coordinate functions by products allows us to encode complex couplings of coordinates (e.g. concerted changes of torsions, or simultaneous dissociation of multiple contacts). In order to avoid a combinatorial explosion of the products, we select a small subset of them using a sparse tensor product approach. Specifically, the tensor-train (TT) format and the method of linear variation are used in order to iteratively select a suitable subspace, from which the eigenvalues and eigenfunctions are approximated. We make use of the alternating linear scheme (ALS) as a learning algorithm, in a version which we have adapted to the problem setting.

Our results suggest that the TT-approach is suitable for selecting a sparse set of products of one-coordinate basis functions in order to approximate the high-dimensional eigenfunctions of molecular conformation spaces. As the resulting eigenfunction approximations are directly related to the molecular coordinates, they can be interpreted via postprocessing methods, and may serve as a way to select the most relevant molecular features that are good reaction coordinates. In the two examples presented, specific coordinates could be recognized as relevant for the slow kinetics or as irrelevant.

Although this research is still in its infancy, our work suggests that approximating eigenfunctions

by a sparse set of products of one-coordinate basis functions may be a promising direction to compute conformation dynamics of macromolecules. Future work will have to adress the question of how stably this method can perform for significantly larger systems. The success of our iterative scheme depends on the ranks r_p, as the computational effort grows with increasing ranks. It will be important to see how these ranks can be controlled for large systems. Also, we expect the ordering of input coordinates to play an important role in the future. Apart from that, we were able to use equilibrium trajectories in the examples presented so far. For large systems, it is usually impossible to provide equilibrium data because of the sampling problem. We will have to study the effects of non-equilibrium data on our current methods, and we will need to develop a framework where statistical uncertainty and discretization error can be balanced. However, we think that the methods presented in this work can be a useful foundation for the study of these problems.

ACKNOWLEDGMENTS

This work was funded by the Einstein foundation Berlin through ECMath, by Deutsche Forschungsgemeinschaft through SFB 1114, and by the European Commission through ERC starting grant "pcCell". We thank Benjamin Trendelkamp-Schroer for helpful discussions and Cecilia Clementi for suggesting the basis set used for BPTI.

Appendix A: Simulation Setup of Deca-Alanine

We performed all-atom molecular-dynamics simulations of deca-alanine, which is protonated at the amino terminus and deprotonated at the carboxy terminus, using the GROMACS 4.5.5 simulation package, the Amber03 force field [42] and the TIP3P water model. A completely elongated conformation was chosen as an initial structure.

The structure was solvated in a cubic box of volume $V = 232.6\,\mathrm{nm}^3$, with 7647 pre-equilibrated TIP3P water molecules. First, an equilibration run of 500 ps in the NVT ensemble with full position restraints, using the Velocity-Rescale thermostat, was carried out. This was followed by a 500 ps NPT equilibration run. The temperature was set to $T = 300\,\mathrm{K}$. The equilibration run

was followed by a 500 ns production run, again at $T = 300\,\mathrm{K}$. Two temperature coupling groups were used with a Velocity-Rescale thermostat and a time constant of 0.01 ps. Periodic boundary conditions were applied in the x, y and z direction. For the long range electrostatic interaction PME was used with a PME-order of 4 and a Fourier grid spacing of 0.15 nm. Covalent bonds to hydrogen bonds were constrained using the LINCS algorithm, allowing for a 2 fs timestep. The leap frog integrator was used. Data was saved every 1 ps, resulting in $5 \cdot 10^5$ data frames. Six independent simulations from the same equilibrated configuration were carried out resulting in $3\,\mu$s total data.

Appendix B: Relation to the Block-TT-Format

Our optimization method shown in Alg. 1 is built on the modification of the ALS (Ref. [31]) for the block-TT-format, see Refs. [38, 39]. The block-TT-format allows for the simultaneous parametrization of a number $M > 1$ functions using only a few additional parameters. A tensor is in block-p-format if there is exactly one component U_p which carries an additional index m, enumerating the different functions, while all remaining components retain their structure as before. Eqs. (29) and (33) then turn into

$$\hat{\psi}_{\mathbf{m}} = \sum_{i_1,\ldots,i_d} \left[\sum_{k_1=1}^{r_1} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} U_1(i_1,k_1)\ldots U_p(k_{p-1},i_p,k_p,\mathbf{m})\ldots U_d(k_{d-1},i_d) \right] f^1_{i_1}(x_1)\ldots f^d_{i_d}(x_d) \tag{B1}$$

$$\hat{\psi}_{\mathbf{m}} = \sum_{k_{p-1},i_p,k_p} U_p(k_{p-1},i_p,k_p,\mathbf{m})\, f^p_{i_p}(x_p)\cdot g^p_{k_{p-1}}(x_1,\ldots,x_{p-1})\cdot h^p_{k_p}(x_{p+1},\ldots,x_d), \tag{B2}$$

where we have highlighted the additional index in bold-face letters. The ALS-optimization of multiple eigenfunctions proceeds as follows: suppose we are on the forward sweep, the tensor is in block-p-format and we seek to update component U_p while all others are fixed. We observe that we can solve the eigenvalue problem Eq. (19) for the three-fold product basis in Eq. (B2), and we can update every slice $U_p(:,:,:,m)$ by the m-th eigenvector thus obtained. In order to proceed to the optimization of the next component U_{p+1}, however, the index m needs to be moved into U_{p+1} first. Otherwise, U_p would parametrize different left interfaces for every value of m, which violates

the idea of the TT-format. In the literature, it is suggested to perform the index move as follows.

- Re-shape the component U_p into a matrix in $\mathbb{R}^{r_{p-1}\cdot n\times r_p\cdot M}$ and compute a low-rank decomposition, e.g. by SVD or QR-decomposition:

$$U_p(k_{p-1}i_p, k_p m) = \sum_{k'_p=1}^{r'_p} V_p(k_{p-1}i_p, k'_p) W_p(k'_p, k_p m). \tag{B3}$$

- Contract the arrays W_p and U_{p+1} by summing over k_p:

$$\tilde{U}_{p+1}(k'_p, i_{p+1}, k_{p+1}, m) = \sum_{k_p=1}^{r_p} W_p(k'_p, k_p m) U_{p+1}(k_p, i_{p+1}, k_{p+1}). \tag{B4}$$

After this, the p-th component can be updated by V_p, which carries no more than three indices, while the $p+1$-st component can be updated by $\tilde{U}_{p+1}$, which now enumerates the index m. Furthermore, the p-th rank has changed to r'_p, thus allowing for rank-adaptivity during the iteration. Also note that the decomposition Eq. (B3) needs to be truncated, otherwise the ranks r_p can easily blow up.

Initially, we attempted to apply ALS using the above method, but the truncation step turned out to be problematic. The main obstacle was that decompositions like SVD do not respect the underlying structure of the problem, namely that the solutions $\hat{\psi}_m$ need to be orthogonal with respect to the weighted inner product Eq. (15). Even for large ranks r'_p, yielding close approximations to the full matrix U_p, the resulting functions $\hat{\psi}_m$ often failed to fulfill the orthogonality constraints. Consequently, we were facing either intolerably large ranks, or meaningless results.

Still, the optimization algorithm described in this work produces a tensor in the block-TT-format. Recall that the optimization of component U_p provides a new left interface $g^{p+1}_{k_p}(U_p)$. The eigenvectors of the generalized eigenvalue problem Eq. (19) parametrize M eigenfunctions in terms of the reduced basis Eq. (40), yielding a component $U_{p+1} \in \mathbb{R}^{r_p\times n\times r_{p+1}\times M}$. Thus, the tensor is in block-$p+1$-format after the optimization. However, this component is not used, as it is updated immediately afterwards by the next optimization step.

Appendix C: Optimization Problem for the Components U_p

Here, we formulate the optimization problem which needs to be solved for increasing ranks r_p in every iteration step of Alg. 1. We seek to determine the optimal component $U_p \in \mathbb{R}^{r_{p-1} \times n \times r_p}$, s.t. the eigenvalue sum Eq. (38) for the reduced basis Eq. (40) is maximal. This is an unconstrained optimization problem which can be solved numerically by a conjugate gradient method if we can provide the derivatives of the eigenvalues $\hat{\lambda}_m(U_p)$ w.r.t. the entries of U_p. These derivatives can be obtained as follows: The eigenvalues $\hat{\lambda}_m(U_p)$ solve the generalized eigenvalue problem Eq. (19) using the reduced correlation matrices $\mathbf{C}^\tau(U_p)$, $\mathbf{C}^0(U_p)$ between the basis functions Eq. (40). These correlation matrices can be computed from the larger correlation matrices $\mathbf{C}^\tau_{p,p+1}$, $\mathbf{C}^0_{p,p+1}$ of the four-fold product basis Eq. (41):

$$\mathbf{C}^\tau_{p,p+1} = \langle \mathcal{T}(\tau) g^{p-1}_{k_{p-1}} f^p_{i_p} f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, g^{p-1}_{l_{p-1}} f^p_{j_p} f^{p+1}_{j_{p+1}} h^{p+1}_{l_{p+1}} \rangle_\pi, \tag{C1}$$

$$\mathbf{C}^0_{p,p+1} = \langle g^{p-1}_{k_{p-1}} f^p_{i_p} f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, g^{p-1}_{l_{p-1}} f^p_{j_p} f^{p+1}_{j_{p+1}} h^{p+1}_{l_{p+1}} \rangle_\pi, \tag{C2}$$

by the formulas

$$[\mathbf{C}^\tau(U_p)]^{k_p, i_{p+1}, k_{p+1}}_{l_p, j_{p+1}, l_{p+1}} = \sum_{k_{p-1}, i_p, l_{p-1}, j_p} U_p(k_{p-1}, i_p, k_p) \left[\mathbf{C}^\tau_{p,p+1}\right]^{k_{p-1}, i_p, i_{p+1}, k_{p+1}}_{l_{p-1}, j_p, j_{p+1}, l_{p+1}} U_p(l_{p-1}, j_p, l_p), \tag{C3}$$

$$\left[\mathbf{C}^0(U_p)\right]^{k_p, i_{p+1}, k_{p+1}}_{l_p, j_{p+1}, l_{p+1}} = \sum_{k_{p-1}, i_p, l_{p-1}, j_p} U_p(k_{p-1}, i_p, k_p) \left[\mathbf{C}^0_{p,p+1}\right]^{k_{p-1}, i_p, i_{p+1}, k_{p+1}}_{l_{p-1}, j_p, j_{p+1}, l_{p+1}} U_p(l_{p-1}, j_p, l_p). \tag{C4}$$

Using these formulas, we can differentiate the matrix entries of $\mathbf{C}^\tau(U_p)$ and $\mathbf{C}^0(U_p)$ w.r.t. the variables U_p:

$$\frac{\partial \left[\mathbf{C}^\tau(U_p)\right]^{k_p, i_{p+1}, k_{p+1}}_{l_p, j_{p+1}, l_{p+1}}}{\partial U_p(k'_{p-1}, i'_p, k'_p)} = \sum_{l_{p-1}, j_p} \left[\mathbf{C}^\tau_{p,p+1}\right]^{k'_{p-1}, i'_p, i_{p+1}, k_{p+1}}_{l_{p-1}, j_p, j_{p+1}, k_{p+1}} U_p(l_{p-1}, j_p, l_p) \delta_{k_p, k'_p} \tag{C5}$$

$$+ \sum_{k_{p-1}, i_p} \left[\mathbf{C}^\tau_{p,p+1}\right]^{k_{p-1}, i_p, i_{p+1}, k_{p+1}}_{k'_{p-1}, i'_p, j_{p+1}, k_{p+1}} U_p(k_{p-1}, i_p, k_p) \delta_{l_p, k'_p}, \tag{C6}$$

$$\frac{\partial \left[\mathbf{C}^0(U_p)\right]_{l_p,j_{p+1},l_{p+1}}^{k_p,i_{p+1},k_{p+1}}}{\partial U_p(k'_{p-1},i'_p,k'_p)} = \sum_{l_{p-1},j_p} \left[\mathbf{C}^0_{p,p+1}\right]_{l_{p-1},j_p,j_{p+1},k_{p+1}}^{k'_{p-1},i'_p,i_{p+1},k_{p+1}} U_p(l_{p-1},j_p,l_p)\delta_{k_p,k'_p} \tag{C7}$$

$$+ \sum_{k_{p-1},i_p} \left[\mathbf{C}^0_{p,p+1}\right]_{k'_{p-1},i'_p,j_{p+1},k_{p+1}}^{k_{p-1},i_p,i_{p+1},k_{p+1}} U_p(k_{p-1},i_p,k_p)\delta_{l_p,k'_p}. \tag{C8}$$

What remains is to compute derivatives of the eigenvalues $\hat{\lambda}_m(U_p)$ w.r.t. the matrix entries of $\mathbf{C}^\tau(U_p)$, $\mathbf{C}^0(U_p)$. For isolated eigenvalues $\hat{\lambda}_m(U_p)$ and positive definite $\mathbf{C}^0(U_p)$, matrix perturbation theory yields the results:

$$\frac{\partial \hat{\lambda}_m(U_p)}{\partial \mathbf{C}^\tau(U_p)(i,j)} = \mathbf{v}_m(i)\mathbf{v}_m(j)\left(2-\delta_{ij}\right) \tag{C9}$$

$$\frac{\partial \hat{\lambda}_m(U_p)}{\partial \mathbf{C}^0(U_p)(i,j)} = -\hat{\lambda}_m(U_p)\mathbf{v}_m(i)\mathbf{v}_m(j)\left(2-\delta_{ij}\right), \tag{C10}$$

where $\mathbf{v}_m$ is the m-th eigenvector corresponding to $\hat{\lambda}_m(U_p)$. Combining Eqs. (C9)-(C10) and (C6)-(C8), we find the derivatives of $\hat{\lambda}_m(U_p)$ w.r.t. the variables U_p. Eqs. (C9)-(C10) can be obtained from perturbation theory. Consider an analytic perturbation of $\mathbf{C}^\tau = \mathbf{C}^\tau(U_p)$ and $\mathbf{C}^0 = \mathbf{C}^0(U_p)$:

$$\tilde{\mathbf{C}}^\tau = \mathbf{C}^\tau + \epsilon\mathbf{C}_1^\tau + \dots \tag{C11}$$

$$\tilde{\mathbf{C}}^0 = \mathbf{C}^0 + \epsilon\mathbf{C}_1^0 + \dots \tag{C12}$$

Then, the proof of [43, Theorem 1] can be imitated for the positive definite generalized eigenvalue problem to show that also the eigenvalue $\tilde{\lambda}_m$ of $\tilde{\mathbf{C}}^\tau$, $\tilde{\mathbf{C}}^0$ can be computed by a series expansion in a small neighborhood of $\mathbf{C}^\tau$, $\mathbf{C}^0$:

$$\tilde{\lambda}_m = \hat{\lambda}_m(U_p) + \epsilon\hat{\lambda}_m^1 + \dots \tag{C13}$$

Moreover, the proof of this theorem also provides an expression for the first order correction $\hat{\lambda}_m^1$. For the positive definite generalized eigenvalue problem, the correction becomes

$$\hat{\lambda}_m^1 = (\mathbf{v}_m)^T\left(\mathbf{C}_1^\tau - \hat{\lambda}_m(U_p)\mathbf{C}_1^0\right)\mathbf{v}_m. \tag{C14}$$

Eqs. (C9)-(C10) now follow if we use the perturbations $\mathbf{C}_1^\tau = \mathbf{C}_1^0 = \mathbf{E}^{ij}$, where $\mathbf{E}^{ij}$ is a matrix whose elements (i,j) and (j,i) are equal to one while all others are zero. Note that the factor $2-\delta_{ij}$ accounts for the symmetry of the matrices $\mathbf{C}^\tau(U_p)$, $\mathbf{C}^0(U_p)$.

Appendix D: Least Squares Approximation of Interfaces

In order to evaluate the contribution of the one-coordinate basis $f_{i_p}^p$ to the full solution, we suggest the following simple method. As before, we explain the method in the context of the forward iteration. The interface functions $g_{k_p}^{p+1}$ encode the relevant information about coordinates $x_1,\ldots,x_p$ into a limited number r_p of functions. If coordinate x_p was relevant for the slow dynamics, these interfaces should differ from the ones computed previously, i.e. from the functions $g_{k_{p-1}}^p$. Therefore, after the interfaces $g_{k_p}^{p+1}$ have been optimized, we approximate these functions in the least squares sense from the basis of previous interfaces $g_{k_{p-1}}^p$. The expansion coefficient vector $\mathbf{u}^{k_p}$ of the best approximation for the interface $g_{k_p}^{p+1}$,

$$f_{k_p} = \sum_{l_{p-1}} \mathbf{u}^{k_p}(l_{p-1}) g_{l_{p-1}}^p, \tag{D1}$$

is found as the solution of the linear system

$$\mathbf{A}^p \mathbf{u}^{k_p} = \mathbf{b}^{k_p} \tag{D2}$$

$$\mathbf{A}^p(k_{p-1}, l_{p-1}) = \langle g_{k_{p-1}}^p, g_{l_{p-1}}^p \rangle_\pi \tag{D3}$$

$$\mathbf{b}^{k_p}(k_{p-1}) = \langle g_{k_{p-1}}^p, g_{k_p}^{p+1} \rangle_\pi. \tag{D4}$$

These quantites can be obtained from the correlation matrix $\mathbf{C}_{p,p+1}^0$ in Eq. (C2). The matrix $\mathbf{A}^p$ is just a submatrix of $\mathbf{C}_{p,p+1}^0$, whereas the vector $\mathbf{b}^{k_p}$ can be computed via

$$\mathbf{b}^{kp}(k_{p-1}) = \sum_{l_{p-1}, j_p} U_p(l_{p-1}, j_p, k_p) \langle g_{k_{p-1}}^{p-1}, g_{l_{p-1}}^{p-1} f_{j_p}^p \rangle_\pi, \tag{D5}$$

where we have used the recursion formula Eq. (35). Next, we can compute the approximation error for $g_{k_p}^{p+1}$ via

$$\left(E(p)_{k_p}\right)^2 = \langle g_{k_p}^{p+1} - f_{k_p}, g_{k_p}^{p+1} - f_{k_p}\rangle_\pi \tag{D6}$$

$$= 1 - 2\langle g_{k_p}^{p+1}, f_{k_p}\rangle_\pi + \langle f_{k_p}, f_{k_p}\rangle_\pi \tag{D7}$$

$$= 1 - 2\left(\mathbf{b}^{k_p}\right)^T \mathbf{u}^{k_p} + \left(\mathbf{u}^{k_p}\right)^T \mathbf{A}^p \mathbf{u}^{k_p}. \tag{D8}$$

Finally, we compute the average approximation error $E(p) = \frac{1}{r_p}\sum_{k_p=1}^{r_p} E(p)_{k_p}$ and use it as a measure of the importance of coordinate x_p.

---

[1] W. van Gunsteren, J. Dolenc, and A. Mark. Molecular Simulation as an Aid to Experimentalists. *Current Opinion in Structural Biology*, 18:149–153, 2008.

[2] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A Direct Approach to Conformational Dynamics based on Hybrid Monte Carlo. *Journal of Computational Physics*, 151:146–168, 1999.

[3] N. Singhal and V. S. Pande. Error Analysis and Efficient Sampling in Markovian State Models for Molecular Dynamics. *Journal of Chemical Physics*, 123:204909, 2005.

[4] F. Noé, I. Horenko, C. Schütte, and J. C. Smith. Hierarchical Analysis of Conformational Dynamics in Biomolecules: Transition Networks of Metastable States. *Journal of Chemical Physics*, 126:155102, 2007.

[5] J. D. Chodera, K. A. Dill, N. Singhal, V. S. Pande, W. C. Swope, and J. W. Pitera. Automatic Discovery of Metastable States for the Construction of Markov Models of Macromolecular Conformational Dynamics. *Journal of Chemical Physics*, 126:155101, 2007.

[6] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. Markov Models of Molecular Kinetics: Generation and Validation. *Journal of Chemical Physics*, 134:174105, 2011.

[7] W. C. Swope, J. W. Pitera, F. Suits, M. Pitman, and M. Eleftheriou. Describing Protein Folding

Kinetics by Molecular Dynamics Simulations. 2 Example Applications to Alanine Dipeptide and Beta-Hairpin Peptide. *Journal of Physical Chemistry B*, 108:6582–6594, 2004.

[8] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *Journal of Chemical Theory and Computation*, 2015.

[9] K. A. Beauchamp, G. R. Bowman, T. J. Lane, L. Maibaum, I. S. Haque, and V. S. Pande. MSMBuilder2: Modeling Conformational Dynamics at the Picosecond to Millisecond Scale. *Journal of Chemical Theory and Computation*, 7:3412–3419, 2011.

[10] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T.R. Weikl. Constructing the Full Ensemble of Folding Pathways from Short Off-Equilibrium Simulations. *Proceedings of the National Academy of Sciences*, 106:19011–19016, 2009.

[11] A. Berezhkovskii, G. Hummer, and A. Szabo. Reactive Flux and Folding Pathways in Network Models of Coarse-Grained Protein Dynamics. *Journal of Chemical Physics*, 130:205102, 2009.

[12] W. Zhuang, R. Z. Cui, D.-A. Silva, and X. Huang. Simulating the T-Jump-Triggered Unfolding Dynamics of trpzip2 Peptide and Its Time-Resolved IR and Two-Dimensional IR Signals Using the Markov State Model Approach. *Journal of Physical Chemistry B*, 115:5415–5424, 2011.

[13] F. Noé, S. Doose, I. Daidone, M. Löllmann, J. D. Chodera, M. Sauer, and J. C. Smith. Dynamical Fingerprints for Probing Individual Relaxation Processes in Biomolecular Dynamics with Simulations and Kinetic Experiments. *Proceedings of the National Academy of Sciences*, 108:4822–4827, 2011.

[14] B. Lindner, Z. Yi, J.-H. Prinz, J. C. Smith, and F. Noé. Dynamic Neutron Scattering from Conformational Dynamics I: Theory and Markov Models. *Journal of Chemical Physics*, 139:175101, 2013.

[15] K. A. Beauchamp, D. L. Ensign, R. Das, and V. S. Pande. Quantitative Comparison of Villin Headpiece Subdomain Simulations and Triplet–Triplet Energy Transfer Experiments. *Proceedings of the National Academy of Sciences*, 108:12734–12739, 2011.

[16] G. R. Bowman and P. L. Geissler. Equilibrium Fluctuations of a Single Folded Protein Reveal a Multitude of Potential Cryptic Allosteric Sites. *Proceedings of the National Academy of Sciences*, 109:11681–11686, 2012.

[17] K. A. Beauchamp, R. McGibbon, Y. S. Lin, and V. S. Pande. Simple Few-State Models Reveal Hidden Complexity in Protein Folding. *Proceedings of the National Academy of Sciences*, 109:17807–17813,

2012.

[18] I. Buch, T. Giorgino, and G. de Fabritiis. Complete Reconstruction of an Enzyme-Inhibitor Binding Process by Molecular Dynamics Simulations. *Proceedings of the National Academy of Sciences*, 108:10184–10189, 2011.

[19] S. K. Sadiq, F. Noé, and G. de Fabritiis. Kinetic Characterization of the Critical Step in HIV-1 Protease Maturation. *Proceedings of the National Academy of Sciences*, 109:20449–20454, 2012.

[20] M. Sarich, F. Noé, and C. Schütte. On the Approximation Quality of Markov State Models. *SIAM Multiscale Modeling and Simulation*, 8:1154–1177, 2010.

[21] F. Noé and F. Nüske. A Variational Approach to Modeling Slow Processes in Stochastic Dynamical Systems. *SIAM Multiscale Modeling and Simulation*, 11:635–655, 2013.

[22] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé. Variational Approach to Molecular Kinetics. *Journal of Chemical Theory and Computation*, 10:1739–1752, 2014.

[23] F. Vitalini, F. Noé, and B. G. Keller. A Basis Set for Peptides for the Variational Approach to Conformational Kinetics. *Journal of Chemical Theory and Computation*, 11:3992–4004, 2015.

[24] I. Oseledets and E. Tyrtyshnikov. Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions. *SIAM Journal on Scientific Computing*, 31:3744–3759, 2009.

[25] I. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317, 2011.

[26] S. Ostlund and S. Rommer. Thermodynamic Limit of Density Matrix Renormalization. *Physical Review Letters*, 75:3537–3540, 1995.

[27] S. R. White. Density Matrix Formulation for Quantum Renormalization Groups. *Physical Review Letters*, 69:2863–2866, 1992.

[28] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza. Tensor Product Methods and Entanglement Optimization for ab initio Quantum Chemistry. *International Journal of Quantum Chemistry*, 115:1342–1391, 2015.

[29] W. Hackbusch and S. Kühn. A New Scheme for the Tensor Representation. *Journal of Fourier Analysis and Applications*, 15:706–722, 2009.

[30] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Berlin Heidelberg, 2012.

[31] S. Holtz, T. Rohwedder, and R. Schneider. The Alternating Linear Scheme for Tensor Optimization

in the Tensor Train Format. *SIAM Journal on Scientific Computing*, 34:A683–A713, 2012.

[32] J. Chodera, W. Swope, J. Pitera, and K. Dill. Long-Time Protein Folding Dynamics from Short-Time Molecular Dynamics Simulations. *SIAM Multiscale Modeling and Simulation*, 5:1214–1226, 2006.

[33] K. Fan. On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations. *Proceedings of the National Academy of Sciences of the United States of America*, 35:652–655, 1949.

[34] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*. Dover Publications, Mineola, NY, 1982.

[35] H. Wu and F. Noé. Gaussian Markov Transition Models of Molecular Kinetics. *The Journal of Chemical Physics*, 142:084104, 2015.

[36] G. Perez-Hernandez, F. Paul, T. Giorgino, G. de Fabritiis, and Frank Noé. Identification of Slow Molecular Order Parameters for Markov Model Construction. *Journal of Chemical Physics*, 139:015102, 2013.

[37] C. R. Schwantes and V. S. Pande. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. *Journal of Chemical Theory and Computation*, 9:2000–2009, 2013.

[38] S.V. Dolgov, B.N. Khoromskij, I.V. Oseledets, and D.V. Savostyanov. Computation of Extreme Eigenvalues in Higher Dimensions Using Block Tensor Train Format. *Computer Physics Communications*, 185:1207 – 1216, 2014.

[39] D. Kressner, M. Steinlechner, and A. Uschmajew. Low-Rank Tensor Methods with Subspace Correction for Symmetric Eigenvalue Problems. *SIAM Journal on Scientific Computing*, 36:A2346–A2368, 2014.

[40] D.E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R.O. Dror, M.P. Eastwood, J.A. Bank, J.M. Jumper, J.K. Salmon, Y. Shan, and W. Wriggers. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science*, 330:341–346, 2010.

[41] F. Noé, H. Wu, J.-H. Prinz, and N. Plattner. Projected and Hidden Markov Models for Calculating Kinetics and Metastable States of Complex Molecules. *The Journal of Chemical Physics*, 139:184114, 2013.

[42] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. Wang, and P. Kollman. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins based on Condensed-Phase Quantum Mechanical Calculations. *Journal of*

Computational Chemistry, 24:1999–2012, 2003.

[43] J. R. Magnus. On Differentiating Eigenvalues and Eigenvectors. *Econometric Theory*, 1:179–191, 1985.

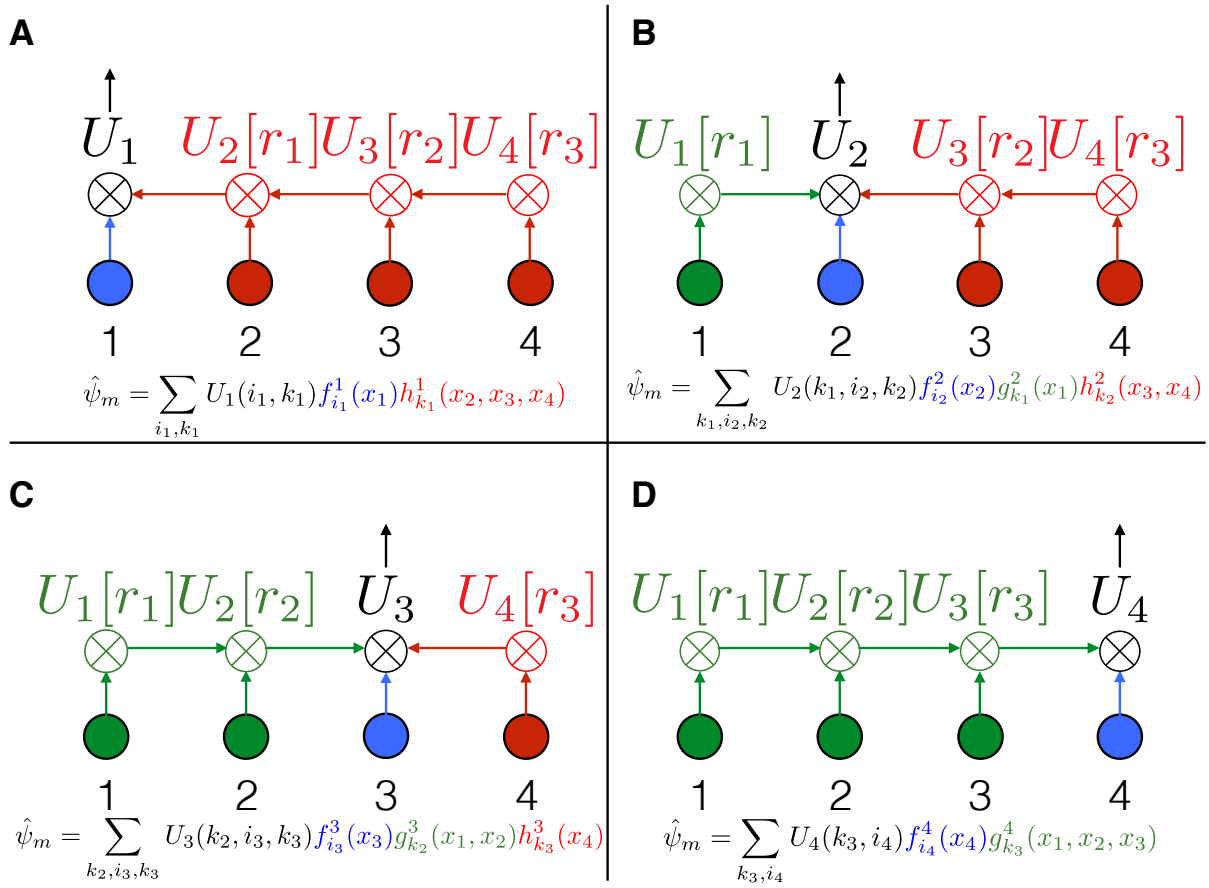

Figure 3. Illustration of a function $\hat{\psi}_m$ of $d = 4$ variables in tensor-train-format. The solid dots at the bottom represent the sets of one-coordinate basis functions $f^p_{i_p}$. Dots with the tensor product symbol $\otimes$ contain all products of the incoming bases, indicated by the arrows. The arrays denoted by $U_p[r_p]$ select r_p linear combinations of the products, to form a new basis. We see that there are d equivalent representations of the function as a linear combination of a reduced and structured basis. If we center the representation around coordinate x_p, then the arrays $U_1, \ldots, U_{p-1}$ encode the information about the variables $x_1, \ldots, x_{p-1}$ into r_{p-1} functions. This process is shown in the green part of each panel. The arrays $U_{p+1}, \ldots, U_d$ encode the information about the variables $x_{p+1}, \ldots, x_d$ into r_p functions, which is shown in the red part of each panel. Both basis sets are combined with the one-coordinate functions $f^p_{i_p}$ (shown in blue), and a linear combination of these products is selected by U_p, which is the final representation of $\hat{\psi}_m$.

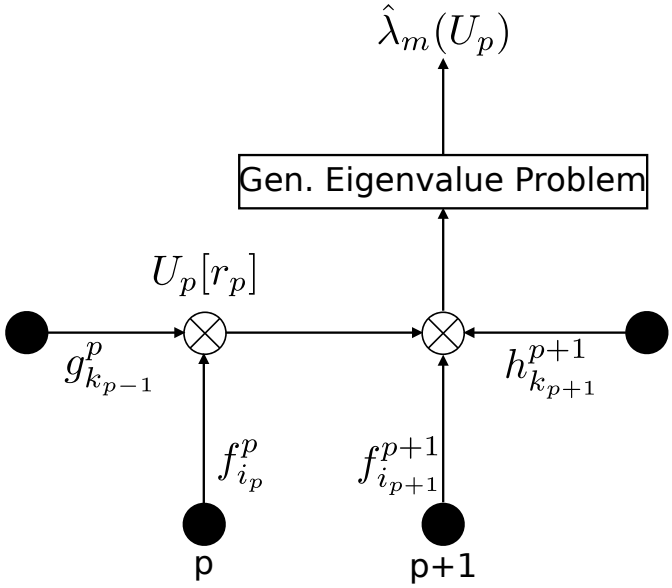

Figure 4. Schematic representation of the optimization problem for the component U_p. This array selects r_p linear combinations of the products $g^p_{k_{p-1}} \cdot f^p_{i_p}$ (see Eq. (35)) to form a new basis $g^{p+1}_{k_p}(U_p)$ in an optimal way. Optimality is defined as follows: We combine the basis $g^{p+1}_{k_p}(U_p)$ with the one-coordinate functions $f^{p+1}_{i_{p+1}}$ and with the right interfaces $h^{p+1}_{k_{p+1}}$ at position $p+1$, to form the basis Eq. (40). For this basis, we solve the generalized eigenvalue problem Eq. (19) to obtain dominant eigenvalues $\hat{\lambda}_m(U_p)$. Optimality of the U_p is defined by maximizing the sum Eq. (38) of the $\hat{\lambda}_m(U_p)$.

$q = 0$

repeat

  $q+ = 1$

  for p=1,. . . ,d-2 **do**

    Solve Eq. (19) for the four-fold basis Eq. (41), obtain eigenvalues $\hat{\lambda}_m^{p,p+1}$.

    Update reference eigenvalue sum L_{ref}.

    for $r_p = 1, \ldots$ **do**

      Optimize coefficients $U_p(k_{p-1}, i_p, k_p)$ s.t. $L_p(U_p)$ defined by Eq. (38) is maximal.

      if $L_p(U_p) \geq \epsilon_{\text{rank}} \cdot L_{\text{ref}}$ **then**

        Update U_p, $g_{k_p}^{p+1}$, r_p

        break

      end if

    end for

  end for

  Repeat this in backward direction for $p = d, \ldots, 3$

until $|L_p^q(U_p) - L_p^{q-1}(U_p)| < \epsilon_{\text{iter}} \quad \forall p$

Algorithm 1. Summary of optimization algorithm.

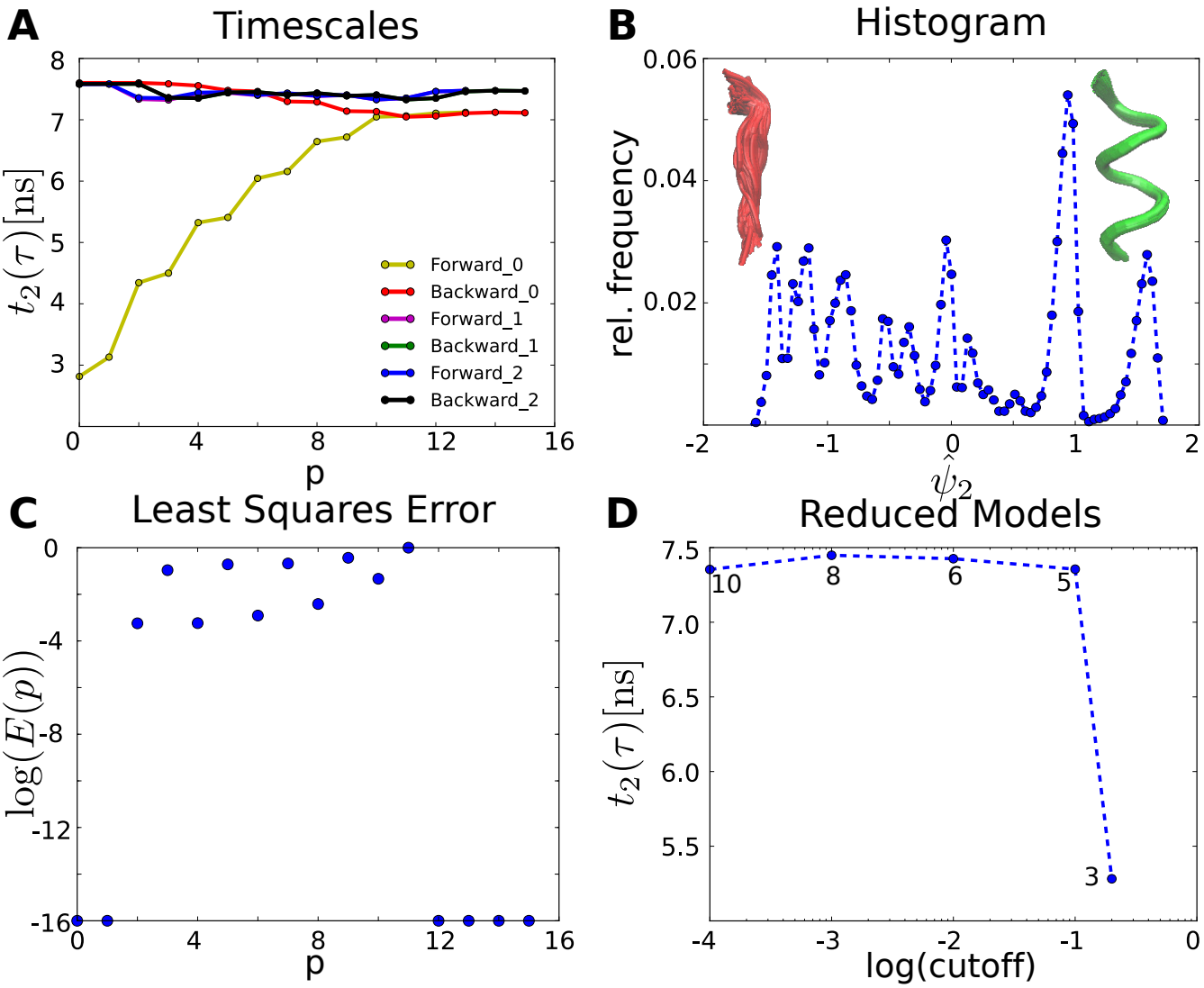

Figure 5. Results for deca-alanine peptide. A: Second implied timescale t_2 in ns along the three forward and backward sweeps of the ALS-iteration. B: Relative histogram of the simulation data along the $\hat{\psi}_2$-coordinate. We identify two peaks of the population corresponding to the most negative (around -1.3 ± 0.3) and the most positive values (1.6 ± 0.2) of the coordinate. Extracting 200 random frames from each of these peaks and superimposing their molecular structures shows that the $\hat{\psi}_2$-coordinate encodes the transition from an elongated conformation to the helix. C: Average approximation error $E(p)$ for the newly determined interface functions at position p, normalized by the maximum error over all coordinates p. D: Second implied timescale t_2 estimated by ALS using only the coordinates satisfying that $E(p)$ is greater than the cutoff given on the horizontal axis. The small numbers next to the data points indicate the number of coordinates used in each model.

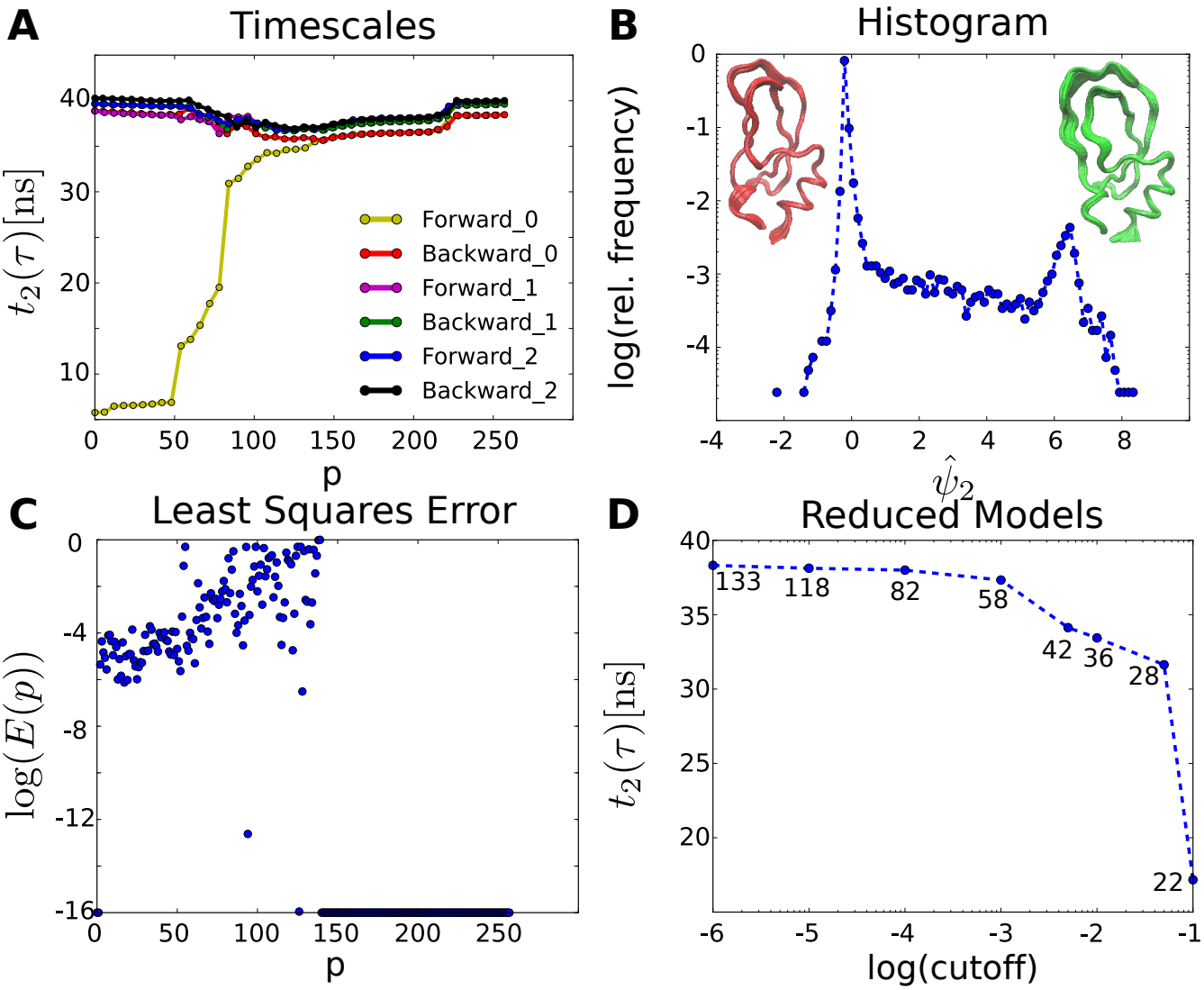

Figure 6. Results for BPTI. A: Second implied timescale t_2 in μs along the three forward and backward sweeps of the ALS-iteration. B: Relative histogram of the simulation data along the $\hat{\psi}_2$-coordinate. We identify two peaks of the population. Extracting 200 random frames from each of these peaks and superimposing their molecular structures shows that the $\hat{\psi}_2$-coordinate encodes the structural transition observed previously in the literature. C: Average approximation error $E(p)$ for the newly determined interface functions at position p, normalized by the maximum error over all coordinates p. D: Second implied timescale t_2 estimated by ALS using only the coordinates satisfying that $E(p)$ is greater than the cutoff given on the horizontal axis. The small numbers next to the data points indicate the number of coordinates used in each model.