Introduction to Markov state modeling with the PyEMMA software [Article v1.0]

Christoph Wehmeyer[1†*], **Martin K. Scherer**[1†], **Tim Hempel**[1†], **Brooke E. Husic**[1,2], **Simon Olsson**[1], **Frank Noé**[1,3*]

[1]Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany; [2]Department of Chemistry, Stanford University, 333 Campus Drive, Stanford, California 94305, USA; [3]Department of Chemistry, Rice University, 6100 Main Street, Houston, Texas 77005, USA

This LiveCoMS document is maintained online on GitHub at github.com/markovmodel/pyemma_tutorials; to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated November 29, 2018

Abstract This tutorial provides an introduction to the construction of Markov models of molecular kinetics from molecular dynamics trajectory data with the PyEMMA software. Using tutorial notebooks, we will guide the user through the basic functionality as well as the more common advanced mechanisms. Short exercises to self check the learning progress and a notebook on troubleshooting complete this basic introduction.

***For correspondence:**
christoph.wehmeyer@fu-berlin.de (CW); frank.noe@fu-berlin.de (FN)

[†]These authors contributed equally to this work

1 Introduction

PyEMMA [1] (http://emma-project.org) is a software for the analysis of molecular dynamics (MD) simulations using Markov state models (MSMs) [2–6]. The package is written in Python (http://python.org), relies heavily on NumPy/SciPy [7, 8], and is compatible with the scikit-learn [9] framework for machine learning.

1.1 Scope

In this tutorial, we assume that the reader is familiar with MD and standard analysis of MD simulations of peptides and proteins, such as computation of torsion angles and distances (see Ref. [10] for a review on the MD simulation of biomolecules, and Ref. [11] for a tutorial on MD simulations).

We further assume that the reader is familiar with the basic ideas and theory underlying Markov state modeling and will only give a brief reminder of the basic concepts in Sec. 2.

For those seeking further resources, the recent perspective "*Markov State Models: From an Art to a Science*" [12] provides a timeline of methods advances with relevant citations, while "*Markov models of molecular kinetics: Generation and validation*" [13] describes the basic MSM theory and methodology and details the underlying mathematics. Additionally, two textbooks have been published that focus on computational methods and applications [14] and mathematical theory [15].

In addition to publications on the theory and application of Markov state modeling [2, 6, 16–26], we also recommend the literature on time-lagged independent component analysis (TICA) [27–30], transition path theory (TPT) [31, 32], hidden

Received: 27 July 2018
Accepted: 27 November 2018

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

Markov state models (HMMs) [33–35], and variational techniques [36–38], as these topics play important roles within the standard MSM workflow.

The tutorial is divided into lessons on specific topics, each accompanied by a Jupyter [39] notebook containing code, instructions, and exercises. The lessons start with a showcase of the PyEMMA workflow and follow up with in-depth lessons on specific topics.

2 Prerequisites

In the following, we summarize the recommended theory and background knowledge of Markov state modeling for this tutorial. Then, we address the software required to work through the lessons.

2.1 Markov state models

Markov state modeling is a mathematical framework for the analysis of time-series data, often but not limited to high dimensional MD simulation datasets. In its standard formulation, the creation of an MSM involves decomposing the phase or configuration space occupied by a dynamical system into a set of disjoint, discrete states, and a transition matrix $\mathbf{P}(\tau) = [p_{ij}(\tau)]$ denoting the conditional probability of finding the system in state j at time $t + \tau$ given that it was in state i at time t. Let us make two remarks to avoid common misconceptions:

1. Equilibrium: While most analysis techniques require simulation trajectories to be long enough to sample from the equilibrium distribution, this is not required for MSMs. Because MSMs use the *conditional* probability $p_{ij}(\tau)$, they are useful for the analysis of short simulation trajectories with arbitrary starting points—see Ref. [25] for restrictions.
2. Markovianity: An MSM is a memoryless model. Early MSM papers have argued that accurate MSMs can be found if a few states with high barriers are captured by the MSM states so as to achieve a Mori-Zwanzig projection with fast-decaying memory [4, 5, 40]. The modern view, however, is that MSMs can be highly accurate if the MSM states discretize the collective coordinates of the slowest processes well [13]. This mainly requires that the system is characterized by only a few slow processes at lag time τ, which is true for cooperative systems such as most proteins, but not for highly frustrated systems such as glasses.

In order to create a Markov state model for a dynamical system, each data point in the time series is assigned to a state. Given an appropriate lag time, every pairwise transition at that lag time is counted and stored in a count matrix. Then, the count matrix is converted to a row-stochastic transition probability matrix $\mathbf{P}(\tau)$, which is defined for the specified lag time. For MD simulations in equilibrium, $\mathbf{P}(\tau)$ should obey detailed balance which is enforced by constraining the estimation of $\mathbf{P}(\tau)$ to the following equations:

$$\pi_i p_{ij} = \pi_j p_{ji}, \tag{1}$$

where π_i is the stationary probability of state i and p_{ij} is the probability of transitioning to state j conditional on being in state i. The constraints (1) are omitted if MD simulations are not conducted in equilibrium, e.g., for systems experiencing a pulling force or an external potential (see Ref. [41] for a recent review on nonequilibrium MSMs). For the remainder of this section we will simplify the matter by assuming the more common scenario of MD simulations without external forces, such that Eqn. (1) is assumed to hold.

When estimating an MSM it is critical to choose a lag time, τ, which is long enough to ensure Markovian dynamics in our state space, but short enough to resolve the dynamics in which we are interested. Plotting the implied timescales (ITS) as a function of τ can be a helpful diagnostic when selecting the MSM lag time [40]. The ITS t_i approximates the decorrelation time of the i^{th} process and is computed from the eigenvalues λ_i of the MSM transition matrix via,

$$t_i = \frac{-\tau}{\ln |\lambda_i(\tau)|}. \tag{2}$$

When the ITS become approximately constant with the lag time, we say that our timescales have converged and choose the smallest lag time with the converged timescales in order to maximize the model's temporal resolution.

Once we have used the ITS to choose the lag time, we can check whether a given transition probability matrix $\mathbf{P}(\tau)$ is approximately Markovian using the Chapman-Kolmogorov (CK) test [13, 18]. The CK property for a Markovian matrix is,

$$\mathbf{P}(k\tau) = \mathbf{P}^k(\tau), \tag{3}$$

where the left-hand side of the equation corresponds to an MSM estimated at lag time $k\tau$ and k is an integer larger than 1. The right-hand side of the equation is our estimated MSM transition probability matrix to the k^{th} power. By assessing how well the approximated transition probability matrix adheres to the CK property, we can validate the appropriateness of the Markovian assumption for the model (see Sec. IV.F in Ref. [13]).

Once validated, the transition matrix can be decomposed into eigenvectors and eigenvalues. The highest eigenvalue, $\lambda_1(\tau)$, is unique and equal to 1. Its corresponding left eigenvector is the stationary distribution, $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}^\top \mathbf{P}(\tau) = \boldsymbol{\pi}^\top. \tag{4}$$

The subsequent eigenvalues $\lambda_{i>1}(\tau)$ are real with absolute values less than 1 and are related to the *characteristic* or

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

implied timescales of dynamical processes within the system (eq. 2). The dynamical processes themselves (for $i > 1$) are encoded by the right eigenvectors ψ_i,

$$\mathbf{P}(\tau)\psi_i = \lambda_i(\tau)\psi_i, \tag{5}$$

where the eigenvalue-eigenvector pairs are indexed in decreasing order according to the eigenvalues. The coefficients of the eigenvectors represent the flux into and out of the Markov states that characterize the corresponding process. The right eigenvector ψ_1 is a vector consisting of 1's.

2.2 Variational approach and TICA

The theory described in the previous section required the decomposition of the phase or configuration space occupied by a dynamical system into discrete, disjoint states. Starting from the output of an MD simulation of a protein, there are several steps that can be taken to obtain an MSM from the original configuration space:

- Featurization: The Cartesian coordinates characterizing each frame of the MD trajectory are transformed into an intuitive basis such as the protein's dihedral angles or contact distance pairs.
- Dimensionality reduction: Optionally, a basis set transformation can be performed that produces a linear (or nonlinear) combination of the features in the previous step. Frequently, TICA [27–29] is used to transform the features into a set of slow coordinates.
- Clustering: This is the step at which the state decomposition occurs. The features or TICs are grouped into a set of states using a clustering algorithm such as *k*-means.
- Transition matrix approximation: At this stage, transitions are counted at a pre-specified lag time, and the estimation and validation described in the previous section are performed.

It is apparent that there are many choices involved in MSM construction such as what features should be used and how many states should be chosen. In 2013, the variational approach to conformational dynamics (VAC) was derived, which enabled an objective comparison among different state decomposition choices for models built with the same Markovian lag time [36]. More recently, the more general variational approach to Markov processes (VAMP) has been developed in order to facilitate the approximation and comparison of reversible models for basis sets that are continuous, as opposed to discrete states [37]. The VAMP can thus be used to perform model selection. Specifically, we use the VAMP-2 score, which captures the kinetic variance explained by the model [29]. However, the MSM lag time cannot be optimized using VAMP, and must be chosen using a separate validation as described above [42].

A commonly used method for dimensionality reduction, TICA, is a particular implementation of the VAC [27]. To apply TICA, we need to compute instantaneous ($\mathbf{C}(0)$) and time-lagged ($\mathbf{C}(\tau)$) covariance matrices with elements

$$c_{ij}(0) = \left\langle \tilde{x}_i(t)\, \tilde{x}_j(t) \right\rangle_t \tag{6}$$

$$c_{ij}(\tau) = \left\langle \tilde{x}_i(t)\, \tilde{x}_j(t+\tau) \right\rangle_t, \tag{7}$$

where $\tilde{x}_i(t)$ denotes the i^{th} feature at time t after the mean has been removed. By default, PyEMMA estimates Eqns. (6) and (7) using symmetrization [27]. This symmetrization induces a significant bias when using nonequilibrium data from short trajectories [43]. As an alternative, the so-called Koopman reweighting estimator is available which avoids this bias, but comes at the cost of a large variance [43].

After estimating the covariance matrices, TICA solves the generalized eigenvalue problem

$$\mathbf{C}(\tau)\,\mathbf{u}_i = \mathbf{C}(0)\,\lambda_i(\tau)\,\mathbf{u}_i, \quad i = 1, \ldots, n \tag{8}$$

to obtain independent component directions $\mathbf{u}_i$ which approximate the reaction coordinates of the system, where the pairs of eigenvalues and independent components are sorted in descending order. A way to measure the contribution of each independent component to the kinetics is obtained by the kinetic distance [29] which assigns a cumulative variance fraction to the first d independent components:

$$c_d = \frac{\sum_{i=2}^{d} \lambda_i^2(\tau)}{\mathrm{TKV}}, \tag{9}$$

where

$$\mathrm{TKV} = \sum_{i=2}^{n} \lambda_i^2(\tau) \tag{10}$$

is the total kinetic variance explained by all n features.

If we further scale the independent components $\mathbf{u}_i$ by the corresponding eigenvectors $\lambda_i(\tau)$, we obtain a *kinetic map* [29] which is the default behavior in PyEMMA.

Note, though, that TICA requires the dynamics to be simulated at equilibrium conditions. To use TICA with nonequilibrium MD, e.g., subject to external forces, or simply to perform dimension reduction on short trajectory data without worrying about reweighting, we recommend to use VAMP, which does not require reversible dynamics [37].

For all of these approaches, dimensionality reduction is performed by projecting the (mean free) features $\tilde{\mathbf{x}}(t)$ onto the leading d independent components $\mathbf{U}_d = \left[\mathbf{u}_1 \ldots \mathbf{u}_d\right]$,

$$\mathbf{y}(t) = \mathbf{U}_d^{\top}\, \tilde{\mathbf{x}}(t), \tag{11}$$

where, in practice, d is preferably chosen such that a specific fraction of kinetic variance c_d is retained (e.g., 95 %).

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

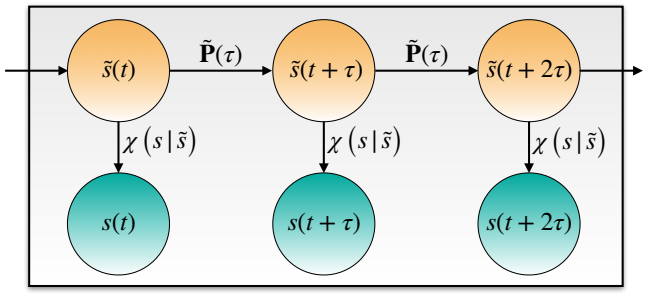

Figure 1. The HMM transition matrix $\tilde{\mathbf{P}}(\tau)$ propagates the hidden state trajectory $\tilde{s}(t)$ (orange circles) and, at each time step t, the emission into the observable state $s(t)$ (cyan circles) is governed by the emission probabilities $\chi\left(s(t)\,|\,\tilde{s}(t)\right)$.

2.3 Hidden Markov state models

The estimation of an MSM requires the dynamics between microstates to be Markovian. However, in case of a poor dimension reduction and/or discretization or short trajectories, we cannot anticipate this to be the case.

An alternative, which is much less sensitive to poor discretization, is to estimate a hidden Markov model (HMM) [33–35, 44]. HMMs are less sensitive to the discretization error as they sidestep the assumption of Markovian dynamics in the discretized space (illustrated in Fig. 1). Instead, HMMs assume that there is an underlying (hidden) dynamic process that is Markovian and gives rise to our observed data, i.e., the (n states) discretized trajectories $s(t)$. This is a powerful principle as we know that there is indeed an underlying process that is Markovian: our molecular dynamics trajectories.

To estimate an HMM, we need a spectral gap after the m^{th} timescale; in practice, a timescale separation of $t_m \geq 2t_{m+1}$ is sufficient [1]. The HMM then consists of a transition matrix $\tilde{\mathbf{P}}(\tau)$ between $m < n$ hidden states and a row-stochastic matrix ($\boldsymbol{\chi}$) of probabilities $\chi\left(s\,|\,\tilde{s}\right)$ to emit the discrete state s conditional on being in the hidden state $\tilde{s}$.

An HMM estimation always yields a model with a small number of (hidden) states in which each state is considered to be metastable and, thus, the number of hidden states is a new hyper-parameter which needs to be chosen carefully. As the HMMs—like MSMs—approximate the full phase-space dynamics, we can similarly compute the metastable kinetics, apply TPT, visualize the network, and obtain physical observables.

For an extensive discussion of details about HMM properties and the estimation algorithm in general, we suggest Ref. [45]. For its specific application to the discretization of MSMs using HMMs, we suggest Ref. [33]. A generalized extension for estimating this type of low dimensional projection from the data is given in Ref. [46]. One of our tutorial notebooks, to be discussed in the next section, provides an example of HMM analysis.

2.4 Software and installation

We utilize Jupyter [39] notebooks to show code examples along with figures and interactive widgets to display molecules. The user can install all necessary packages in one step using the `conda` command provided by the Anaconda Python stack (https://conda.io/miniconda.html). We recommend Anaconda because it resolves and installs dependencies and provides pre-compiled versions of common packages. The tutorial installation contains a launcher command to start the Jupyter notebook server as well as the notebook files.

The user can install the tutorial's dependencies in a new conda environment and start the notebook server via

```
conda create -n pyemma_tutorials
conda activate pyemma_tutorials
conda install -c conda-forge pyemma_tutorials
pyemma_tutorials
```

or refer to github.com/markovmodel/pyemma_tutorials for more detailed installation and usage instructions.

The data for the demonstrated test systems is downloaded upon the first use and is cached for future invocations of the tutorial.

The underlying software stack for running the tutorial consists of:

- **PyEMMA**: MSM/HMM estimation, validation, analysis, and visualization, and its dependencies [1]
- mdshare: A downloader for MD data from a public server
- notebook: The Jupyter [39] notebook tool used for running the tutorials, along with extension packages jupyter_contrib_nbextensions and nbexamples
- matplotlib: A plotting library [47]
- nglview: Widget for active viewing of molecular structures in Jupyter environments [48]

The tutorial software is currently supported for Python versions 3.5 and 3.6 on the operating systems Linux, OSX, and Windows.

Should the user prefer not to use Anaconda, a manual installation via the pip installer is possible. Alternatively, one can use the Binder service (https://mybinder.org) to view and run the tutorials online in any browser.

3 PyEMMA tutorials

This tutorial consists of nine Jupyter notebooks which introduce the basic features of PyEMMA. The first notebook (00), which we will summarize in the following, showcases the entire estimation, validation, and analysis workflow for a small

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

example system. The goal of this introductory notebook is to provide the user with the typical steps required to obtain a validated MSM analysis of protein or peptide simulation data. The seven subsequent notebooks (01–07) provide in-depth lessons on specific topics, and the last notebook (08) contains guidelines on how to deal with common problems during MSM estimation.

3.1 The PyEMMA workflow

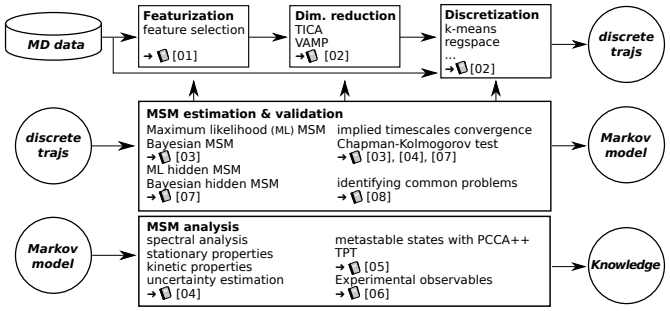

Figure 2. The PyEMMA workflow: MD trajectories are processed and discretized (first row). A Markov state model is estimated from the resulting discrete trajectories and validated (middle row). By iterating between data processing and MSM estimation/validation, a dynamical model is obtained that can be analyzed (last row).

In short, the workflow (Fig. 2) for a full analysis of an MD dataset might consist of

- extracting molecular features from the raw data (01),
- transforming those features into a suitable, low dimensional subspace (02),
- discretizing the low dimensional subsets into a state decomposition (02),
- estimating a maximum likelihood or Bayesian MSM from the discrete trajectories and performing validation tests (03),
- analyzing the stationary and kinetic properties of the MSM (04),
- finding metastable macrostates and applying transition path theory (TPT) to identify the pathways of conformational change (05),
- computing expectation values for experimental observables (06), and
- coarse-graining the MSM using a hidden Markov model approach (07).

For the remainder of this manuscript we will walk through the first notebook (00). In notebook 00 we analyze a dataset of the Trp-Leu-Ala-Leu-Leu pentapeptide (Fig. 3a), consisting of 25 independent MD trajectories conducted in implicit solvent with frames saved at an interval of 0.1 ns. We present the results obtained in this notebook, thereby providing an example of how results generated using PyEMMA can be integrated into research publications. The figures that will be displayed in the following sections are created in the showcase notebook 00 and can be easily reproduced.

Note that the modeler has to select hyper-parameters at most stages throughout the workflow. This selection must be done carefully as poor choices make it hard, or even impossible, to build a good MSM.

While there exist automated schemes [49] for cross-validated optimization in the full hyper-parameter space, we chose to adopt a sequential approach where only the hyper-parameters of the current stage are optimized. This approach is not only computationally cheaper but allows us to discuss the significance of the necessary modeling choices.

3.2 Feature selection

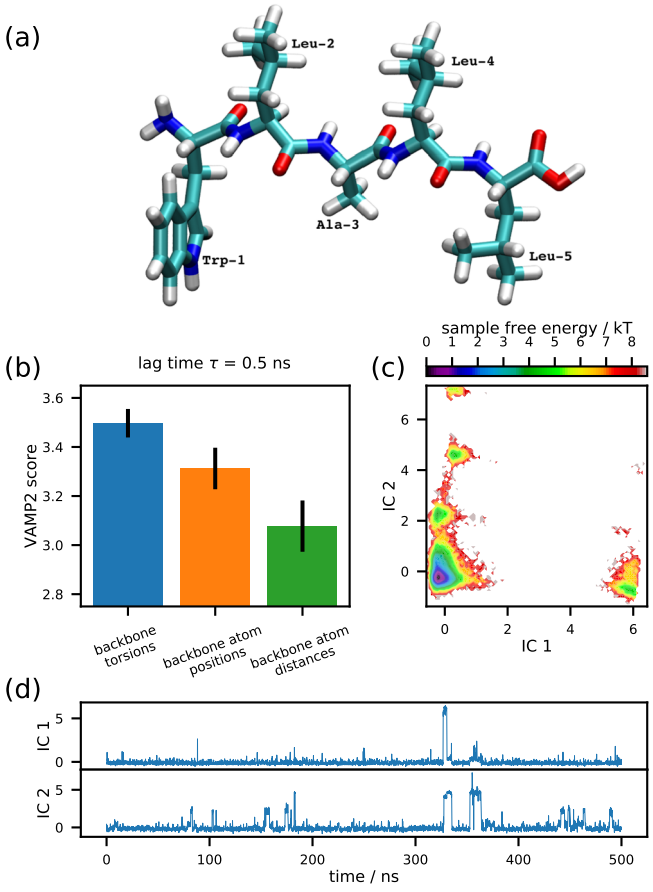

Figure 3. Example analysis of the conformational dynamics of a pentapeptide backbone: (a) The Trp-Leu-Ala-Leu-Leu pentapeptide in licorice representation [50]. (b) The VAMP-2 score indicates which of the tested featurizations contains the highest kinetic variance. (c) The sample free energy projected onto the first two time-lagged independent components (ICs) at lag time τ = 0.5 ns shows multiple minima and (d) the time series of the first two ICs of the first trajectory show rare transitions.

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

In Markov state modeling, our objective is to model the slow dynamics of a molecular process. In order to approximate the slow dynamics in a statistically efficient manner, a lower dimensional representation of our simulation data is necessary. However, the features (e.g. torsion angles, distances or contacts) which best represent the slow dynamical modes of a given molecular system are unknown a priori [51]. Fortunately, the variational principle of conformational dynamics [36, 52] and the more general VAMP [37] provide a systematic means to quantitatively compare multiple representations of the simulation data. In particular, we can use a scalar score obtained using VAMP to directly compare the ability of certain features to capture slow dynamical modes in a particular molecular system. In notebook 01, we present in detail how to extract features from MD datasets and how to systematically compare them.

Throughout this tutorial, we utilize the VAMP-2 score, which maximizes the kinetic variance contained in the features [29]. We should always evaluate the score in a cross-validated manner to ensure that we neither include too few features (under-fitting) or too many features (over-fitting) [37, 38]. To choose among three different molecular features reflecting protein structure, we compute the (cross-validated) VAMP-2 score (notebook 00). Although we cannot optimize MSM lag times with a variational score[42], such as VAMP-2, it is important to ensure that the properties we optimize are robust as a function of lag time. Consequently, we compute the VAMP-2 score at several lag times (notebook 00). We find that the relative rankings of the different molecular features are highly robust as a function of lag time. We show one example of this ranking and the absolute VAMP-2 scores for lag time 0.5 ns in Fig. 3b. We find that backbone torsions contain more kinetic variance than the backbone heavy atom positions or the distances between them (Fig. 3b). This suggests that backbone torsions are the best of the options evaluated for MSM construction.

We note that deep learning approaches for feature selection have recently been developed that may eventually replace the feature selection step [53–55].

3.3 Dimensionality reduction

Subsequently, we perform TICA [27–29] in order to reduce the dimension from the feature space, which typically contains many degrees of freedom, to a lower dimensional space that can be discretized with higher resolution and better statistical efficiency. TICA is a special case of the variational principle [36, 52] and is designed to find a projection preserving the long-timescale dynamics in the dataset. Here, performing TICA on the backbone torsions at lag time 0.5 ns yields a four dimensional subspace using a 95 % kinetic variance cutoff (note that we perform a cos / sin-transformation of the torsions before TICA in order to preserve their periodicity). The sample free energy projected onto the first two independent components (ICs) exhibits several minima (Fig. 3c). Discrete transitions between the minima can be observed by visualizing the transformation of the first trajectory into these ICs (Fig. 3d). We thus assume that our TICA-transformed backbone torsion features describe one or more metastable processes.

We demonstrate how to apply TICA, suggest how to interpret the projected coordinates, and compare the results to other dimension reduction techniques in notebook 02.

3.4 Discretization

TICA yields a representation of our molecular simulation data with a reduced dimensionality, which can greatly facilitate the decomposition of our system into the discrete Markovian states necessary for MSM estimation. Here, we use the *k*-means algorithm to segment the four dimensional TICA space into $k = 75$ cluster centers. The number of cluster centers has been chosen to optimize the VAMP-2 score in a manner identical to how the feature selection was carried out above, which is shown in the showcase notebook 00. A detailed comparison between different clustering techniques is provided in notebook 02.

3.5 MSM estimation and validation

A necessary condition for Markovian dynamics in our reduced space is that the ITS are approximately constant as a function of τ; accordingly, we chose the smallest possible τ which fulfills this condition within the model uncertainty. The uncertainty bounds are computed using a Bayesian scheme [16, 23] with 100 samples. In our example, we find that the four slowest ITS converge quickly and are constant within a 95 % confidence interval for lag times above 0.5 ns (Fig. 4a). Using this lag time we can now estimate a (Bayesian) MSM with $\tau = 0.5$ ns.

To test the validity of our MSM, we perform a CK test. Visualizing the full transition probability matrix *T* is difficult; we therefore coarse-grain *T* into a smaller number of metastable states before performing the test. An appropriate number of metastable states can be chosen by identifying a relatively large gap in the ITS plot. For this analysis, we chose five metastable states. The CK test (Fig. 4b) shows that predictions from our MSM (blue-dashed lines) agrees well with MSMs estimated with longer lag times (black-solid lines) Thus, the CK test confirms that five metastable states is an appropriate choice and shows that the MSM we have estimated at lag time $\tau = 0.5$ ns indeed predicts the long-timescale behavior of our system within error (blue/shaded area).

In notebook 03, we demonstrate in detail how to estimate and validate MSMs with PyEMMA.

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

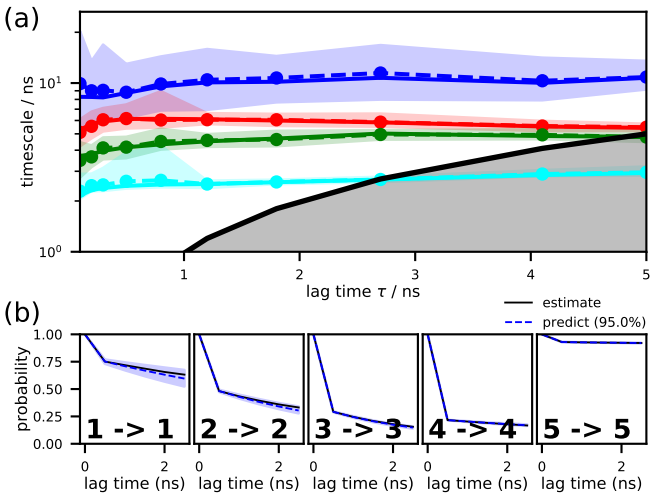

Figure 4. Example analysis of the conformational dynamics of a pentapeptide backbone: (a) The convergence behavior of the implied timescales associated with the four slowest processes. The solid lines refer to the maximum likelihood result while the dashed lines show the ensemble mean computed with a Bayesian sampling procedure [23]. The black line (marking equality of timescale and lag time) with grey area indicates the timescale horizon below which the MSM cannot resolve processes. As implied timescales are well-converged at τ = 0.5 ns, this lag time is chosen for subsequent MSM estimation. (b) CK test computed using an MSM estimated with lag time τ = 0.5 ns assuming 5 metastable states. Predictions from this model agree with higher lag time estimates within confidence intervals. Implied timescales convergence as well as a passing CK test are a necessary condition in MSM validation. In both panels, the (non-grey) shaded areas indicate 95 % confidence intervals computed with the aforementioned Bayesian sampling procedure.

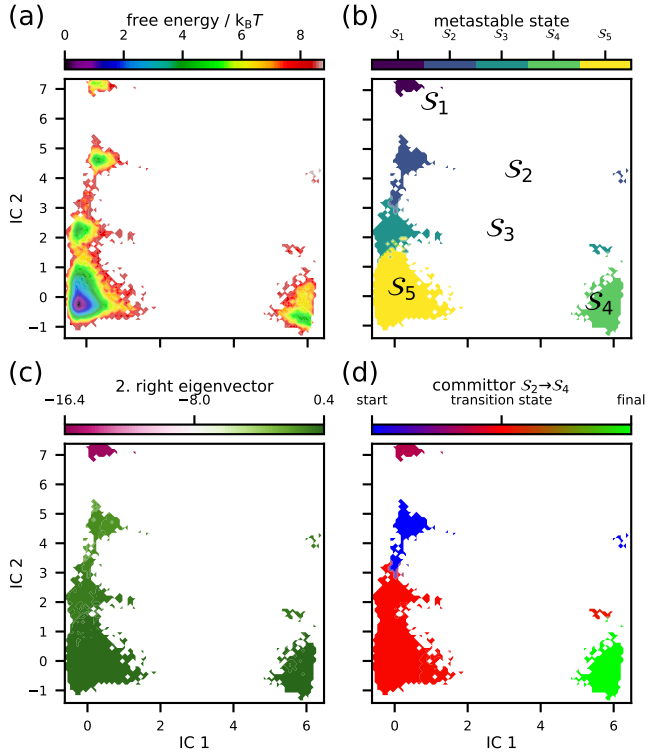

Figure 5. Example analysis of the conformational dynamics of a pentapeptide backbone: (a) The reweighted free energy surface projected onto the first two independent components exhibits five minima which (b) PCCA++ identifies as five metastable states. (c) The second right eigenvector shows that the slowest process shifts probability between the least probable state ($\mathcal{S}_1$) and the other states, in particular states ($\mathcal{S}_4$, $\mathcal{S}_5$), whereas (d) the committor $\mathcal{S}_2 \to \mathcal{S}_4$ indicates that states $\mathcal{S}_{(1,3,5)}$ act as a transition region between states $\mathcal{S}_2$ and $\mathcal{S}_4$.

3.6 MSM Analysis

We can now directly extract several thermodynamic and kinetic properties from the estimated and validated model. An example of the former is the free energy surface in the projection onto the first two TICA components (Fig. 5a) reweighted by the MSM stationary distribution.

A spectral clustering using the PCCA++ algorithm [56–58] allows us to coarse-grain the 75 *k*-means microstates into five metastable macrostates (Fig. 5b) $\mathcal{S}_i$, $i = 1, \ldots, 5$, for which we then approximate the stationary probabilities and relative free energies (defined up to an additive constant)

macrostate $\mathcal{S}_i$	$\pi_{\mathcal{S}_i}$	$G_{\mathcal{S}_i}/\mathrm{k_B}T$
$\mathcal{S}_1$	0.004	5.567
$\mathcal{S}_2$	0.014	4.293
$\mathcal{S}_3$	0.021	3.841
$\mathcal{S}_4$	0.021	3.875
$\mathcal{S}_5$	0.940	0.062

using the relation

$$G_{\mathcal{S}_i} = -\mathrm{k_B}T \ln \sum_{j \in \mathcal{S}_i} \pi_j, \qquad (12)$$

where π_j denotes the MSM stationary weight of the j^{th} microstate.

In order to interpret the slowest relaxation timescales, we refer to the (right) eigenvectors, as they are independent of the stationary distribution (see Sec. 2.1). This enables us to specifically study what conformational changes are happening on a particular timescale independently of the equilbrium distribution. The first right eigenvector corresponds to the stationary process and its eigenvalue is the Perron eigenvalue 1. The second right eigenvector, on the other hand, corresponds to the slowest process in the system. Note that the eigenvectors are real because detailed balance has been enforced during MSM estimation. The minimal and maximal components of the second right eigenvector indicate the microstates between which the process shifts probability density. The relaxation timescale of this exchange process is exactly the corresponding implied timescale, which can be computed from its corresponding eigenvalue using Eqn. (2). In the projection onto the first two TICA components, we identify the slowest MSM process as a probability shift between

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

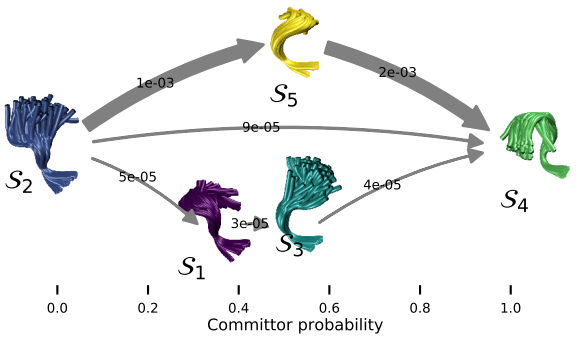

Figure 6. Example analysis of the conformational dynamics of a pentapeptide backbone: visualization of the transition paths from $\mathcal{S}_2$ to $\mathcal{S}_4$. Metastable states $\mathcal{S}_{(1-5)}$ are represented by an ensemble of representative structures and are arranged along the horizontal axis according to their committor probabilities. The three main transition pathways starting from $\mathcal{S}_2$ and ending in $\mathcal{S}_4$ are depicted by gray arrows with thickness proportional to the transition flux. The dominant pathway proceeds through $\mathcal{S}_5$.

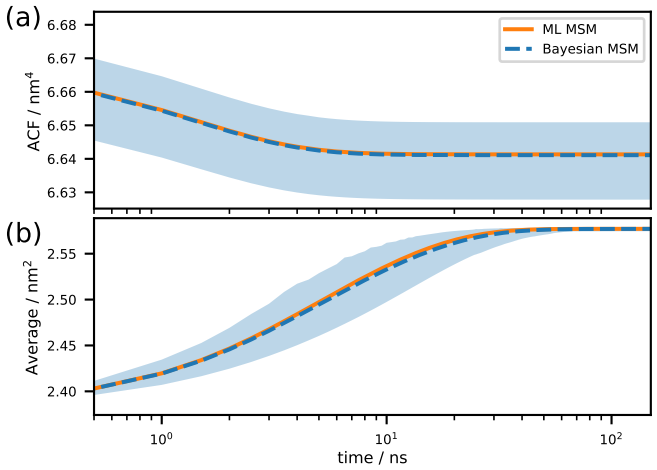

Figure 7. Example analysis of the conformational dynamics of a pentapeptide backbone: (a) the Trp-1 SASA autocorrelation function yields a weak signal which, however, (b) can be enhanced if the system is prepared in the nonequilibrium condition $\mathcal{S}_1$. The solid/orange lines denote the maximum likelihood MSM result; the dashed/blue lines and the the shaded areas indicate sample means and 95 % confidence intervals computed with a Bayesian sampling procedure [23].

macrostate $\mathcal{S}_1$ and the rest of the system, with macrostates $\mathcal{S}_4$ and $\mathcal{S}_5$ in particular (Fig. 5c).

The mean first passage times (MFPTs) out of and into the macrostate $\mathcal{S}_1$ compute to

direction	mean / ns		std / ns
$\mathcal{S}_1 \to \mathcal{S}_{(2,3,4,5)}$	9.0	±	1.9
$\mathcal{S}_{(2,3,4,5)} \to \mathcal{S}_1$	2496.4	±	470.0

using the Bayesian MSM.

TPT [31, 32] is a method used to analyze the statistics of transition pathways. TPT as implemented in Ref. [18] can be conveniently applied to the estimated MSM. Here, we compute the TPT flux between macrostates $\mathcal{S}_2$ and $\mathcal{S}_4$ (Fig. 5d). The committor projection onto the first two TICA components shows that it is constant within the metastable states defined above. Transition regions (macrostates $\mathcal{S}_{(1,3,5)}$) can be identified by committor values $\approx \frac{1}{2}$.

The transition network can be additionally visualized by plotting representative structures of the five metastable states $\mathcal{S}_{(1-5)}$ according to their committor probability (Fig. 6). It is easy to see from this depiction that the dominant pathway from $\mathcal{S}_2$ to $\mathcal{S}_4$ proceeds through $\mathcal{S}_5$.

More details about (spectral) properties of MSMs and how to analyze them with PyEMMA are discussed in notebook 04 and notebook 05.

3.7 Connecting the MSM with experimental data

MSMs can also be analyzed in the context of experimental observables. Connecting MSM analysis to experimental data can both serve as an accuracy test of our MSM as well as provide a mechanistic interpretation of observed experimental signals.

Since we have both the stationary and dynamic properties of the molecular system encoded in the MSM transition probability matrix, we can compute observables that involve both stationary ensemble averages as well as correlation functions.

As an example, we look at the fluorescence correlation of Trp-1, since this terminal tryptophan is a realistic experimental observable for our pentapeptide system. In order to compute the fluorescence correlation functions we require a microscopic, instantaneous value of the tryptophan fluorescence for each of the original 75 MSM microstates. To approximate the fluorescence signal in our pentapeptide system, we use the mdtraj library [59] to compute the solvent accessible surface area (SASA) [60] of Trp-1. Now that we have an approximation of the fluorescence in each of our MSM states, we can use PyEMMA to compute the fluorescence autocorrelation function (ACF) from our MSM (Fig. 7a). Note how the computed ACF has a very small response (i.e., signal amplitude).

Using PyEMMA, we can simulate the relaxation of an observable from a nonequilibrium initial condition. The experimental counterpart of such a prediction could be a temperature or pressure jump experiment or a stopped flow assay. To illustrate such an experiment, we initialize our molecular ensemble as the metastable distribution of $\mathcal{S}_1$ and follow the predicted fluorescence signal as it relaxes to equilibrium (Fig. 7b). We see that the predicted relaxation signal has a much larger amplitude for the nonequilibrium initialization, making it more likely to be experimentally measurable.

In addition to a detailed demonstration of the above, note-

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

book 06 demonstrates how to compute J-couplings and dynamic fingerprints from MSMs.

3.8 Summary of the showcase notebook

In this section, we have summarized how to conduct an MSM-based analysis of biomolecular dynamics data using PyEMMA. For the full analysis, please refer to the first notebook (00). All notebooks as well as detailed installation instructions are available on github.com/markovmodel/pyemma_tutorials.

3.9 Modeling large systems

When estimating MSMs for large systems, challenges may arise that are mostly system dependent.

A case in point is the curse of dimensionality: it is difficult to discretize a high dimensional feature space. While it is somewhat computationally demanding, more importantly, Euclidean distances become less meaningful with increasing dimensionality [61] and thus cluster assignments based on that norm may yield a poor discretization. Especially for large systems, it is particularly important to first find a suitable set of features, and to further apply dimensionality reduction techniques (e.g., TICA, VAMP, if applicable) to obtain a low dimensional representation of the slow dynamics. Hidden Markov models (HMMs) might further mitigate poor discretization to a certain extent [33].

Furthermore, the slowest process in a system as identified by an MSM or HMM might not be the one a modeler is interested in [62]. For instance, the slowest process might correspond to a biologically irrelevant side chain flip that only occurred once in the data set. This problem may be mitigated by choosing a more specific set of features.

Additional technical challenges for large systems include high demands on memory and computation time; we explain how to deal with those in the tutorials (notebook 01). More details on how to model complex systems with the techniques presented here are described, e.g., by Refs. [63, 64]. We further examine some symptoms that may indicate problematic or difficult datasets, and demonstrate how to deal with them in notebook 08.

3.10 Advanced Methods

The present tutorial presents the basics of modern Markov state modeling with PyEMMA. However, recent years have seen many extensions of the methodology, many of which are available within PyEMMA. We encourage interested readers to look into these methods in the software documentation and to make use of the specific Jupyter notebooks distributed with PyEMMA (http://emma-project.org).

Conventional Markov state modeling often relies on large simulation datasets to ensure proper convergence of thermodynamic and kinetic properties. In one extension, multi-ensemble Markov models (MEMMs) [65, 66], we can integrate unbiased and biased simulations in a systematic manner to speed up the convergence. MEMMs consequently enable users to combine enhanced sampling methods such as umbrella sampling or replica exchange with conventional molecular dynamics simulations to more efficiently study rare event kinetics [67]. MEMMs are implemented in PyEMMA. Since the many publications associated with the development of these methods are beyond the scope of this tutorial, we refer the reader to Sec. 8.3 of Ref. [12] and the references therein.

Another issue often faced during Markov state modeling is a lack of quantitative agreement with complementary experimental data. This issue is not intrinsic to the Markov state modeling approach as such, but rather is associated with systematic errors in the force field model used to conduct the simulation. Nevertheless, using Augmented Markov models (AMM) it is possible to build an integrative MSM which balances experimental and simulation data, taking into account their respective uncertainties [26]. AMMs are also implemented in PyEMMA.

Recently, there have been steps towards replacing the traditional user-directed pipeline (involving featurizing, reducing dimension, discretizing, MSM estimation and coarse-graining) by a single end-to-end deep learning method such as VAMPnets [53]. Other deep learning methods for performing the dimension reduction [54], finding reaction coordinates for enhanced sampling [55, 68, 69], and generative MSMs [70] have been put forward and are likely to spawn an active field of research in its own right. Implementations of some of these methods are available or are under development in the deeptime package github.com/markovmodel/deeptime.

4 Author Contributions

CW, MKS, TH, SO, and FN designed research. CW, MKS, TH, BEH, and SO developed and tested notebooks. MKS developed the software infrastructure, test, and install environment. CW, MKS, TH, BEH, SO, and FN wrote the manuscript.

For a more detailed description of author contributions, see the GitHub issue tracking and changelog at github.com/markovmodel/pyemma_tutorials.

5 Other Contributions

We are grateful to Nuria Plattner for providing the pentapeptide simulation data and Camilla Ventura Santos as well as the entire computational molecular biology group for valuable discussion and feedback.

For a more detailed description of contributions from the community and others, see the GitHub issue tracking and changelog at github.com/markovmodel/pyemma_tutorials.

https://doi.org/10.33011/livecoms.1.1.5965
Living J. Comp. Mol. Sci. 2018, 1, 1

6 Potentially Conflicting Interests

The authors declare no conflicting interests.

7 Funding Information

MKS acknowledges financial support from European Commission (ERC StG 307494 "pcCell"). TH acknowledges financial support from Deutsche Forschungsgemeinschaft (SFB/TRR 186, Project A12). SO acknowledges a postdoctoral fellowship from the Alexander von Humboldt Foundation. FN and MKS acknowledge funding from Deutsche Forschungsgemeinschaft (SFB 1114, Projects A04 and C03, NO 825/2-2). FN and BEH acknowledge funding from European Commission (ERC CoG 772230 "ScaleCell").

References

[1] **Scherer MK**, Trendelkamp-Schroer B, Paul F, Pérez-Hernández G, Hoffmann M, Plattner N, Wehmeyer C, Prinz JH, Noé F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. J Chem Theory Comput. 2015; 11(11):5525–5542. https://doi.org/10.1021/acs.jctc.5b00743.

[2] **Schütte C**, Fischer A, Huisinga W, Deuflhard P. A Direct Approach to Conformational Dynamics Based on Hybrid Monte Carlo. J Comput Phys. 1999; 151(1):146–168. https://doi.org/10.1006/jcph.1999.6231.

[3] **Singhal N**, Snow CD, Pande VS. Using path sampling to build better Markovian state models: Predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. J Chem Phys. 2004; 121(1):415. https://doi.org/10.1063/1.1738647.

[4] **Noé F**, Horenko I, Schütte C, Smith JC. Hierarchical analysis of conformational dynamics in biomolecules: Transition networks of metastable states. J Chem Phys. 2007; 126(15):155102. https://doi.org/10.1063/1.2714539.

[5] **Chodera JD**, Singhal N, Pande VS, Dill KA, Swope WC. Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. J Chem Phys. 2007; 126(15):155101. https://doi.org/10.1063/1.2714538.

[6] **Buchete NV**, Hummer G. Coarse Master Equations for Peptide Folding Dynamics†. J Phys Chem B. 2008; 112(19):6057–6069. https://doi.org/10.1021/jp0761665.

[7] **Oliphant TE**. Guide to NumPy. 2nd ed. USA: CreateSpace Independent Publishing Platform; 2015.

[8] **Jones E**, Oliphant T, Peterson P, et al., SciPy: Open source scientific tools for Python; 2001–. http://www.scipy.org/.

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

[10] **Dror RO**, Dirks RM, Grossman JP, Xu H, Shaw DE. Biomolecular Simulation: A Computational Microscope for Molecular Biology. Annu Rev Biophys. 2012; 41(1):429–452. https://doi.org/10.1146/annurev-biophys-042910-155245.

[11] **Braun E**, Gilmer J, Mayes HB, Mobley DL, Monroe JI, Prasad S, Zuckerman DM, Best Practices for Foundations in Molecular Simulations [Article v1.0]; 2018. https://doi.org/10.33011/livecoms.1.1.5957.

[12] **Husic BE**, Pande VS. Markov State Models: From an Art to a Science. J Am Chem Soc. 2018; 140(7):2386–2396. https://doi.org/10.1021/jacs.7b12191.

[13] **Prinz JH**, Wu H, Sarich M, Keller B, Senne M, Held M, Chodera JD, Schütte C, Noé F. Markov models of molecular kinetics: Generation and validation. J Chem Phys. 2011; 134(17):174105. https://doi.org/http://dx.doi.org/10.1063/1.3565032.

[14] **Bowman GR**, Pande VS, Noé F. An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation. Bowman GR, Pande VS, Noé F, editors, Springer Netherlands; 2014. https://doi.org/10.1007/978-94-007-7606-7.

[15] **Sarich M**, Schütte C. Metastability and Markov State Models in Molecular Dynamics. Courant Lecture Notes, American Mathematical Society; 2013.

[16] **Noé F**. Probability distributions of molecular observables computed from Markov models. J Chem Phys. 2008; 128(24):244103. https://doi.org/10.1063/1.2916718.

[17] **Bowman GR**, Beauchamp KA, Boxer G, Pande VS. Progress and challenges in the automated construction of Markov state models for full protein systems. J Chem Phys. 2009; 131(12):124101. https://doi.org/10.1063/1.3216567.

[18] **Noé F**, Schütte C, Vanden-Eijnden E, Reich L, Weikl TR. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. Proc Natl Acad Sci USA. 2009; 106(45):19011–19016. https://doi.org/10.1073/pnas.0905466106.

[19] **Sarich M**, Noé F, Schütte C. On the Approximation Quality of Markov State Models. Multiscale Model Simul. 2010; 8(4):1154–1177. https://doi.org/10.1137/090764049.

[20] **Noe F**, Doose S, Daidone I, Lollmann M, Sauer M, Chodera JD, Smith JC. Dynamical fingerprints for probing individual relaxation processes in biomolecular dynamics with simulations and kinetic experiments. Proc Natl Acad Sci USA. 2011; 108(12):4822–4827. https://doi.org/10.1073/pnas.1004646108.

[21] **Lindner B**, Yi Z, Prinz JH, Smith JC, Noé F. Dynamic neutron scattering from conformational dynamics. I. Theory and Markov models. J Chem Phys. 2013; 139(17):175101. https://doi.org/10.1063/1.4824070.

[22] **Chodera JD**, Noé F. Markov state models of biomolecular conformational dynamics. Curr Opin Struct Biol. 2014; 25:135–144. https://doi.org/10.1016/j.sbi.2014.04.002.

[23] **Trendelkamp-Schroer B**, Wu H, Paul F, Noé F. Estimation and uncertainty of reversible Markov models. J Chem Phys. 2015; 143(17):174101. https://doi.org/10.1063/1.4934536.

[24] **Olsson S**, Noé F. Mechanistic Models of Chemical Exchange Induced Relaxation in Protein NMR. J Am Chem Soc. 2016; 139(1):200–210. https://doi.org/10.1021/jacs.6b09460.

[25] **Nüske F**, Wu H, Prinz JH, Wehmeyer C, Clementi C, Noé F. Markov state models from short non-equilibrium simulations—Analysis and correction of estimation bias. J Chem Phys. 2017; 146(9):094104. https://doi.org/10.1063/1.4976518.

[26] **Olsson S**, Wu H, Paul F, Clementi C, Noé F. Combining experimental and simulation data of molecular processes via augmented Markov models. Proc Natl Acad Sci USA. 2017; 114(31):8265–8270. https://doi.org/10.1073/pnas.1704803114.

[27] **Pérez-Hernández G**, Paul F, Giorgino T, Fabritiis GD, Noé F. Identification of slow molecular order parameters for Markov model construction. J Chem Phys. 2013; 139(1):015102. https://doi.org/10.1063/1.4811489.

[28] **Schwantes CR**, Pande VS. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. J Chem Theory Comput. 2013; 9(4):2000–2009. https://doi.org/10.1021/ct300878a.

[29] **Noé F**, Clementi C. Kinetic Distance and Kinetic Maps from Molecular Dynamics Simulation. J Chem Theory Comput. 2015; 11(10):5002–5011. https://doi.org/10.1021/acs.jctc.5b00553.

[30] **Molgedey L**, Schuster HG. Separation of a mixture of independent signals using time delayed correlations. Phys Rev Lett. 1994; 72(23):3634–3637. https://doi.org/10.1103/physrevlett.72.3634.

[31] **E W**, Vanden-Eijnden E. Towards a Theory of Transition Paths. J Stat Phys. 2006; 123(3):503–523. https://doi.org/10.1007/s10955-005-9003-9.

[32] **Metzner P**, Schütte C, Vanden-Eijnden E. Transition Path Theory for Markov Jump Processes. Multiscale Model Simul. 2009; 7(3):1192–1219. https://doi.org/10.1137/070699500.

[33] **Noé F**, Wu H, Prinz JH, Plattner N. Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules. J Chem Phys. 2013; 139(18):184114. https://doi.org/10.1063/1.4828816.

[34] **Prinz JH**, Chodera JD, Noé F. Spectral Rate Theory for Two-State Kinetics. Phys Rev X. 2014; 4(1). https://doi.org/10.1103/physrevx.4.011020.

[35] **Chodera JD**, Elms P, Noé F, Keller B, Kaiser CM, Ewall-Wice A, Marqusee S, Bustamante C, Singhal Hinrichs N. Bayesian hidden Markov model analysis of single-molecule force spectroscopy: Characterizing kinetics under measurement uncertainty. arXiv preprint arXiv:11081430. 2011; https://arxiv.org/pdf/1108.1430.pdf.

[36] **Noé F**, Nüske F. A Variational Approach to Modeling Slow Processes in Stochastic Dynamical Systems. Multiscale Model Simul. 2013; 11(2):635–655. https://doi.org/10.1137/110858616.

[37] **Wu H**, Noé F. Variational approach for learning Markov processes from time series data. arXiv preprint arXiv:170704659. 2017; https://arxiv.org/pdf/1707.04659.pdf.

[38] **McGibbon RT**, Pande VS. Variational cross-validation of slow dynamical modes in molecular kinetics. J Chem Phys. 2015; 142(12):124105. https://doi.org/10.1063/1.4916292.

[39] **Kluyver T**, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, Kelley K, Hamrick J, Grout J, Corlay S, Ivanov P, Avila D, Abdalla S, Willing C. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: Loizides F, Schmidt B, editors. *Positioning and Power in Academic Publishing: Players, Agents and Agendas* IOS Press; 2016. p. 87–90.

[40] **Swope WC**, Pitera JW, Suits F. Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory†. J Phys Chem B. 2004; 108(21):6571–6581. https://doi.org/10.1021/jp037421y.

[41] **Koltai P**, Wu H, Noé F, Schütte C. Optimal Data-Driven Estimation of Generalized Markov State Models for Non-Equilibrium Dynamics. Computation. 2018; 6(1):22. https://doi.org/10.3390/computation6010022.

[42] **Husic BE**, Pande VS. Note: MSM lag time cannot be used for variational model selection. J Chem Phys. 2017; 147(17):176101. https://doi.org/10.1063/1.5002086.

[43] **Wu H**, Nüske F, Paul F, Klus S, Koltai P, Noé F. Variational Koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. J Chem Phys. 2017; 146(15):154104. https://doi.org/10.1063/1.4979344.

[44] **Baum LE**, Petrie T, Soules G, Weiss N. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Ann Math Stat. 1970; 41(1):164–171. http://www.jstor.org/stable/2239727.

[45] **Rabiner LR**. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE. 1989; 77(2):257–286. https://doi.org/10.1109/5.18626.

[46] **Wu H**, Prinz JH, Noé F. Projected metastable Markov processes and their estimation with observable operator models. J Chem Phys. 2015; 143(14):10B610_1.

[47] **Hunter JD**. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007; 9(3):90–95. https://doi.org/10.1109/MCSE.2007.55.

[48] **Nguyen H**, Case DA, Rose AS. NGLview–interactive molecular graphics for Jupyter notebooks. Bioinformatics. 2017; 34(7):1241–1242. https://doi.org/10.1093/bioinformatics/btx789.

[49] **Husic BE**, McGibbon RT, Sultan MM, Pande VS. Optimized parameter selection reveals trends in Markov state models for protein folding. J Chem Phys. 2016; 145(19):194103. https://doi.org/10.1063/1.4967809.

[50] **Humphrey W**, Dalke A, Schulten K. VMD: Visual molecular dynamics. J Mol Graph. 1996; 14(1):33–38. https://doi.org/10.1016/0263-7855(96)00018-5.

[51] **Noé F**, Clementi C. Collective variables for the study of long-time kinetics from molecular trajectories: theory and methods. Curr Opin Struct Biol. 2017; 43:141–147. https://doi.org/10.1016/j.sbi.2017.02.006.

[52] **Nüske F**, Keller BG, Pérez-Hernández G, Mey ASJS, Noé F. Variational Approach to Molecular Kinetics. J Chem Theory Comput. 2014; 10(4):1739–1752. https://doi.org/10.1021/ct4009156.

[53] **Mardt A**, Pasquali L, Wu H, Noé F. VAMPnets for deep learning of molecular kinetics. Nat Commun. 2018; 9(1). https://doi.org/10.1038/s41467-017-02388-1.

[54] **Wehmeyer C**, Noé F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. J Chem Phys. 2018; 148(24):241703. https://doi.org/10.1063/1.5011399.

[55] **Hernández CX**, Wayment-Steele HK, Sultan MM, Husic BE, Pande VS. Variational encoding of complex dynamics. Phys Rev E. 2018; 97(6). https://doi.org/10.1103/physreve.97.062412.

[56] **Röblitz S**, Weber M. Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification. Adv Data Anal Classif. 2013; 7(2):147–179. https://doi.org/10.1007/s11634-013-0134-6.

[57] **Deuflhard P**, Weber M. Robust Perron cluster analysis in conformation dynamics. Linear Algebra Appl. 2005; 398:161–184. https://doi.org/10.1016/j.laa.2004.10.026.

[58] **Kube S**, Weber M. A coarse graining method for the identification of transition rates between molecular conformations. J Chem Phys. 2007; 126(2):024103. https://doi.org/10.1063/1.2404953.

[59] **McGibbon RT**, Beauchamp KA, Harrigan MP, Klein C, Swails JM, Hernández CX, Schwantes CR, Wang LP, Lane TJ, Pande VS. MDTraj: A Modern Open Library for the Analysis of Molecular Dynamics Trajectories. Biophys J. 2015; 109(8):1528 – 1532. https://doi.org/10.1016/j.bpj.2015.08.015.

[60] **Shrake A**, Rupley JA. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. J Mol Biol. 1973; 79(2):351–371. https://doi.org/10.1016/0022-2836(73)90011-9.

[61] **Aggarwal CC**, Hinneburg A, Keim DA. On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: Van den Bussche J, Vianu V, editors. *Database Theory — ICDT 2001* Lecture Notes in Computer Science, Springer Berlin Heidelberg; 2001. p. 420–434.

[62] **Banushkina PV**, Krivov SV. Nonparametric variational optimization of reaction coordinates. J Chem Phys. 2015; 143(18):184108. https://doi.org/10.1063/1.4935180.

[63] **Plattner N**, Noé F. Protein conformational plasticity and complex ligand-binding kinetics explored by atomistic simulations and Markov models. Nat Commun. 2015; 6:7653. https://doi.org/10.1038/ncomms8653.

[64] **Plattner N**, Doerr S, Fabritiis GD, Noé F. Complete protein–protein association kinetics in atomic detail revealed by molecular dynamics simulations and Markov modelling. Nat Chem. 2017; 9(10):1005. https://doi.org/10.1038/nchem.2785.

[65] **Wu H**, Mey ASJS, Rosta E, Noé F. Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. J Chem Phys. 2014; 141(21):214106. https://doi.org/10.1063/1.4902240.

[66] **Wu H**, Paul F, Wehmeyer C, Noé F. Multiensemble Markov models of molecular thermodynamics and kinetics. Proc Natl Acad Sci USA. 2016; 113(23):E3221–E3230. https://doi.org/10.1073/pnas.1525092113.

[67] **Paul F**, Wehmeyer C, Abualrous ET, Wu H, Crabtree MD, Schöneberg J, Clarke J, Freund C, Weikl TR, Noé F. Protein-peptide association kinetics beyond the seconds timescale from atomistic simulations. Nat Commun. 2017; 8(1). https://doi.org/10.1038/s41467-017-01163-6.

[68] **Sultan MM**, Wayment-Steele HK, Pande VS. Transferable Neural Networks for Enhanced Sampling of Protein Dynamics. J Chem Theory Comput. 2018; 14(4):1887–1894. https://doi.org/10.1021/acs.jctc.8b00025.

[69] **Ribeiro JML**, Bravo P, Wang Y, Tiwary P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). J Chem Phys. 2018; 149(7):072301. https://doi.org/10.1063/1.5025487.

[70] **Wu H**, Mardt A, Pasquali L, Noe F. Deep Generative Markov State Models. arXiv preprint arXiv:180507601. 2018; https://arxiv.org/pdf/1805.07601.pdf.