Introduction to Markov state modeling with the PyEMMA software — v0.3

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

[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

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 August 3, 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 [2, 3] (MSMs). The package is written in Python (http://python.org), relies heavily on NumPy/SciPy [4, 5], and is compatible with the scikit-learn [6] framework for machine learning.

1.1 Scope

In this tutorial, we assume the reader's familiarity with the basic theory behind the MSM approach (see Sec. 2.1) and focus on usage of PyEMMA. Nevertheless, we will mention important theoretical concepts when appropriate throughout the tutorial. We also assume the reader is familiar with MD analyses of proteins and peptides and commonly used structural components of these systems (for a review of the simulation of biomolecular macromolecules, see [7]).

The tutorial is divided into lessons on specific topics, each accompanied by a Jupyter [8] 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 theoretical background knowledge of Markov state modeling for this tutorial. Then, we address the software required to work through the lessons.

2.1 Background knowledge

For those unfamiliar with Markov state modeling, "*Markov State Models: From an Art to a Science*" [9] provides a recent

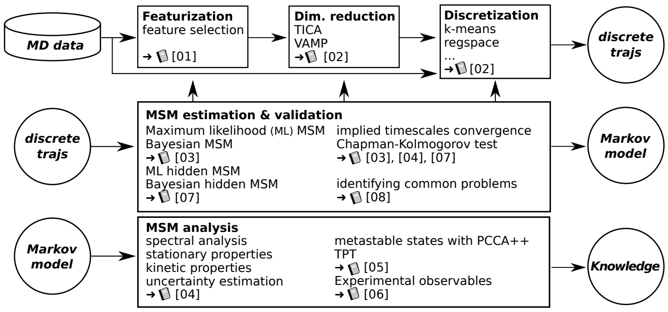

Figure 1. The PyEMMA workflow: MD trajectories are processed and discretized (first row). A Markov state model is estimated from the resulting discrete trajectories and validated (middle row). By iterating between data processing and MSM estimation/validation, a dynamical model is obtained that can be analyzed (last row).

overview, while "*Markov models of molecular kinetics: Generation and validation*" [10] describes the basic MSM theory and methodology in detail. Additionally, two textbooks exist that focus on computational methods and applications [11] and mathematical theory [12].

In addition to publications on theory and application of Markov state modeling [2, 13–24], we also recommend the literature on time-lagged independent component analysis (TICA) [25–28], transition path theory (TPT) [29, 30], hidden Markov state models (HMMs) [31–33], and variational techniques [34–36], as these topics play important roles within the standard MSM workflow.

2.2 Software/system requirements

We utilize Jupyter [8] 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://anaconda.com). We recommend Anaconda because it resolves and installs dependencies as well as provides precompiled versions of common packages.

The tutorial installation contains a launcher command to start the Jupyter notebook server as well as the notebook files. 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 notebook tool used for running the tutorials [8], along with extension packages jupyter_contrib_nbextensions and nbexamples
- nglview – Widget for active viewing of molecular structures in Jupyter environments [37]

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 to view and run the tutorials online in any browser.

3 Content and links

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 example system. The goal of this introductory notebook (00) 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

In short, the workflow 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. 2a), 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 the 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 are created in the showcase notebook (00) and can be easily reproduced.

3.2 Feature selection

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 [39]. Fortunately, the variational principle of conformational dynamics [34, 40] and the more general variational approach for Markov processes (VAMP) [35] 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.

Here, we utilize the VAMP-2 score, which maximizes the kinetic variance contained in the features [28]. 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) [35, 36]. To choose among three different molecular features relevant to protein structure, we compute the (cross-validated) VAMP-2 score at a lag time of 0.5 ns and find that backbone torsions contain more kinetic variance than the backbone's heavy atom positions or the distances between them (Fig. 2b).

We note that deep learning approaches for feature selection have recently been developed that may eventually replace the feature selection step [41–43].

3.3 Dimensionality reduction

Subsequently, we perform TICA [25, 28] 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 [34, 40] 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 density projected onto the first two independent components (ICs) exhibits several maxima (Fig. 2c). Discrete jumps between the maxima can be observed by visualizing the transformation of the first trajectory into these ICs (Fig. 2d). We

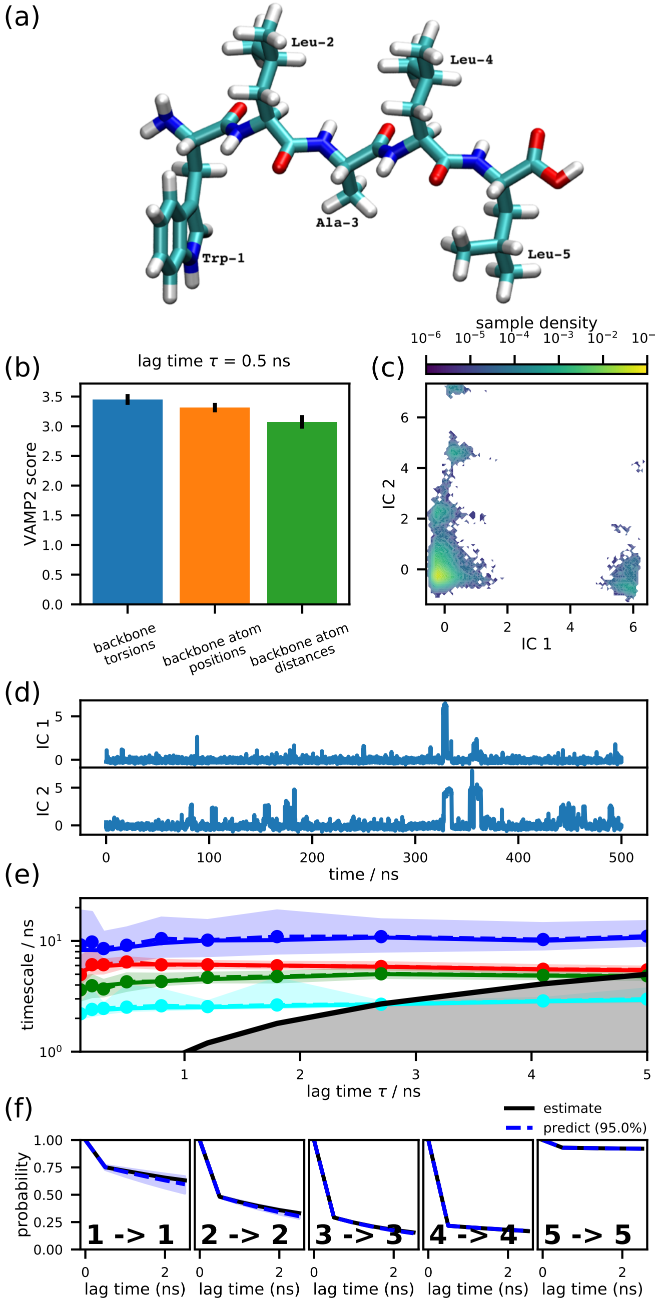

Figure 2. Exemplary analysis of the conformational dynamics of a pentapeptide backbone: (a) The Trp-Leu-Ala-Leu-Leu pentapeptide in licorice representation [38]. (b) The VAMP-2 score indicates which of the tested featurizations contains the highest kinetic variance. (c) The sample density projected onto the first two time-lagged independent components (ICs) at lag time τ = 0.5 ns shows multiple density maxima and (d) the time series of the first two ICs show rare transition events. (e) The convergence behavior of the first four implied timescales indicates that a lag time of τ = 0.5 ns is suitable for MSM estimation. (f) A Chapman-Kolmogorov test shows that an MSM estimated at lag time τ = 0.5 ns under the assumption of five metastable states accurately predicts the kinetic behavior on longer timescales. In (e) and (f), the shaded areas indicate 95% confidence intervals computed with a Bayesian sampling procedure.

thus assume that our TICA-transformed backbone torsion features describe one or more metastable processes.

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).

3.5 MSM estimation and validation

When estimating an MSM it is critical to choose a lag time, τ, which is long enough to ensure Markovian dynamics in our reduced 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 [44]. 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

$$t_i = \frac{-\tau}{\ln|\lambda_i(\tau)|}. \tag{1}$$

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 [14, 21] 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. 2e). 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 Chapman-Kolmogorov (CK) test. The CK test compares the right and the left side of the Chapman-Kolmogorov equation

$$T(k\tau) = T^k(\tau) \tag{2}$$

where T is the MSM transition matrix. The left-hand side of the equation corresponds to an MSM estimated at lag time $k\tau$, where k is an integer larger than 1, whereas the right-hand side of the equation is our estimated MSM to the k^{th} power. 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 5 metastable states. The CK test confirms that this 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 (Fig. 2f).

3.6 Analyzing the MSM

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. 3a) reweighted by the MSM stationary distribution.

A spectral clustering using the PCCA++ algorithm [45–47] allows us to coarse-grain the 75 k-means microstates into five metastable macrostates (Fig. 3b) $\mathcal{S}_i$, $i = 1, \dots, 5$, 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

$$G_{\mathcal{S}_i} = -\mathrm{k_B}T \ln \sum_{j \in \mathcal{S}_i} \pi_j, \tag{3}$$

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 of the MSM as they contain information about what configurational changes are happening and their timescales. The first right eigenvector corresponds to the stationary process and its eigenvalue is the Perron eigenvalue 1. The second right eigenvector, however, corresponds to the slowest process (the eigenvector components are real because of the detailed balance constraint 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 (1). In the projection onto the first two TICA components, we identify the slowest MSM process as a probability shift between macrostate $\mathcal{S}_1$ and the rest of the system, with macrostates $\mathcal{S}_4$ and $\mathcal{S}_5$ in particular (Fig. 3c).

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

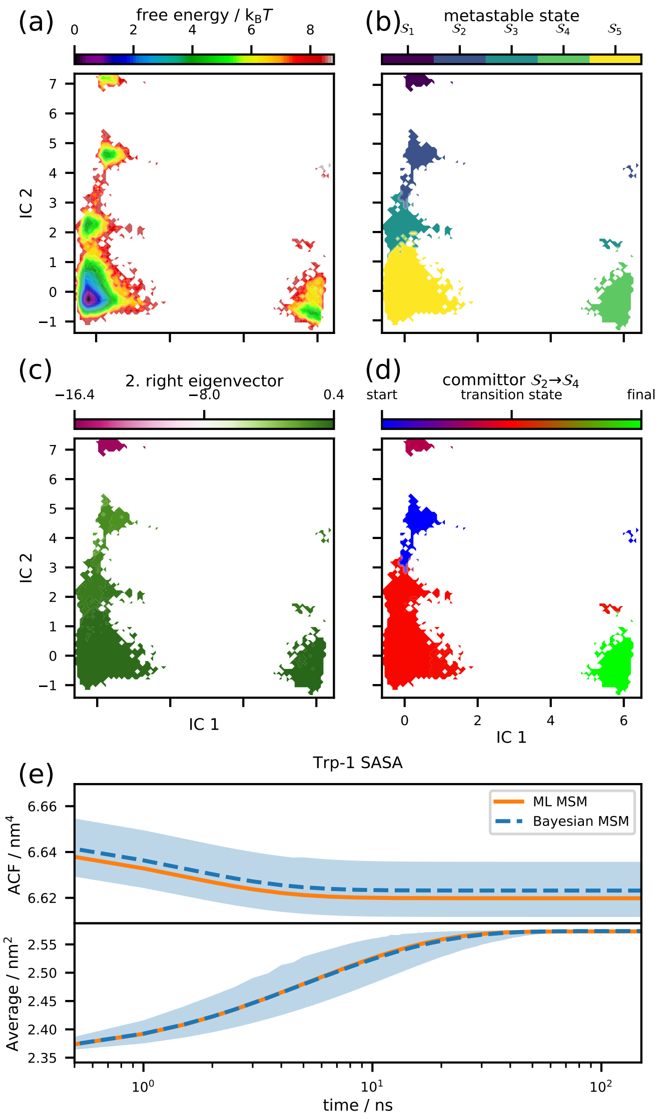

Figure 3. Exemplary analysis of the conformational dynamics of a pentapeptide backbone: (a) The reweighted free energy surface projected onto the first two independent components exhibits five minima which (b) PCCA++ identifies as five metastable states. (c) The second right eigenvector shows that the 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 $\mathcal{S}_2 \rightarrow \mathcal{S}_4$ indicates that states $\mathcal{S}_{(1,3,5)}$ act as a transition region between states $\mathcal{S}_2$ and $\mathcal{S}_4$. (e) The Trp-1 SASA autocorrelation function yields a weak signal (top) which, however, can be enhanced if the system is prepared in the nonequilibrium condition $\mathcal{S}_1$ (bottom).

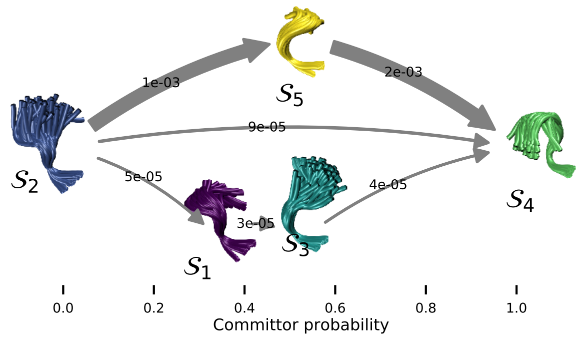

Figure 4. Visualization of the transition paths from $\mathcal{S}_2$ to $\mathcal{S}_4$: Metastable states $\mathcal{S}_{(1-5)}$ are represented by an ensemble of representative structures and are arranged along the horizonal axis according to their committor probabilities. The three main transition 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 $\mathcal{S}_5$.

using the Bayesian MSM.

TPT [29, 30] is a method used to analyze the statistics of transition pathways. The TPT version of [16] can be conveniently applied to the estimated MSM. Here, we compute the TPT flux between macrostates $\mathcal{S}_2$ and $\mathcal{S}_4$ (Fig. 3d). 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. 4). It is easy to see from this depiction that the dominant pathway from $\mathcal{S}_2$ to $\mathcal{S}_4$ proceeds through $\mathcal{S}_5$.

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, here 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 [48] to compute the solvent accessible surface area (SASA) 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 (3e, upper). Note how the computed ACF has a very small response (i.e., signal amplitude).

Using PyEMMA, we can simulate the relaxation of an observable if we had prepared our molecular system in 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 (3e, lower). We see that the predicted relaxation signal has a much larger amplitude for the nonequilibrium initialization, making it more likely to be experimentally measurable.

3.8 Summary

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 Advanced Methods

While the present tutorial is intended to cover Markov State Modeling 101, we encourage the user to explore other, more recent extensions of the methodology. Multi-ensemble Markov models (MEMMs) [49, 50] can be used to combine unbiased and biased simulations so as to probe kinetics of very rare events [51]; MEMMs are 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 [41]. Other deep learning methods for performing the dimension reduction [42], finding reaction coordinates for enhanced sampling [43, 52, 53], and generative MSMs [54] have been put forward and are likely to spawn an active field of research on 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.

6 Potentially Conflicting Interests

The authors declare no conflicting interests.

7 Funding Information

TH acknowledges financial support by SFB/TRR 186. SO acknowledges a postdoctoral fellowship from the Alexander von Humboldt Foundation.

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 nov; 11(11):5525–5542. http://dx.doi.org/10.1021/acs.jctc.5b00743, doi: 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 may; 151(1):146–168. https://doi.org/10.1006/jcph.1999.6231, doi: 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, doi: 10.1063/1.1738647.

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

[5] **Jones E**, Oliphant T, Peterson P, et al., SciPy: Open source scientific tools for Python; 2001–. http://www.scipy.org/, [Online; accessed 2018-07-20].

[6] **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.

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

[8] **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.

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

[10] **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. http://scitation.aip.org/content/aip/journal/jcp/134/17/10.1063/1.3565032, doi: http://dx.doi.org/10.1063/1.3565032.

[11] **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%2F978-94-007-7606-7, doi: 10.1007/978-94-007-7606-7.

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

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

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

[15] **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 sep; 131(12):124101. https://doi.org/10.1063/1.3216567, doi: 10.1063/1.3216567.

[16] **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 nov; 106(45):19011–19016. https://doi.org/10.1073/pnas.0905466106, doi: 10.1073/pnas.0905466106.

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

[18] **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 mar; 108(12):4822–4827. https://doi.org/10.1073/pnas.1004646108, doi: 10.1073/pnas.1004646108.

[19] **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 nov; 139(17):175101. https://doi.org/10.1063/1.4824070, doi: 10.1063/1.4824070.

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

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

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

[23] **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 mar; 146(9):094104. https://doi.org/10.1063/1.4976518, doi: 10.1063/1.4976518.

[24] **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 jul; 114(31):8265–8270. https://doi.org/10.1073/pnas.1704803114, doi: 10.1073/pnas.1704803114.

[25] **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 jul; 139(1):015102. https://doi.org/10.1063/1.4811489, doi: 10.1063/1.4811489.

[26] **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 Apr; 9(4):2000–2009. http://dx.doi.org/10.1021/ct300878a, doi: 10.1021/ct300878a.

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

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

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

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

[31] **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 nov; 139(18):184114. https://doi.org/10.1063/1.4828816, doi: 10.1063/1.4828816.

[32] **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.

[33] **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, doi: 10.1109/5.18626.

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

[35] **Wu H**, Noé F. Variational approach for learning Markov processes from time series data. ArXiv e-prints. 2017 Jul; .

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

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

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

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

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

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

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

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

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

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

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

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

[48] **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. doi: 10.1016/j.bpj.2015.08.015.

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

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

[51] **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 Oct; 8(1). http://dx.doi.org/10.1038/s41467-017-01163-6, doi: 10.1038/s41467-017-01163-6.

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

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

[54] **Wu H**, Mardt A, Pasquali L, Noe F. Deep Generative Markov State Models. ArXiv e-prints. 2018 May; .