JCTC Journal of Chemical Theory and Computation

Article

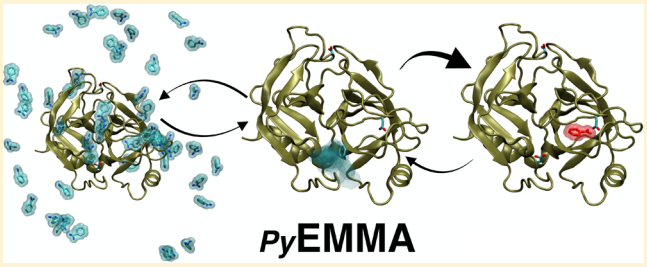
pubs.acs.org/JCTC

PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models

Martin K. Scherer,[†] Benjamin Trendelkamp-Schroer,[†] Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé*

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

ABSTRACT: Markov (state) models (MSMs) and related models of molecular kinetics have recently received a surge of interest as they can systematically reconcile simulation data from either a few long or many short simulations and allow us to analyze the essential metastable structures, thermodynamics, and kinetics of the molecular system under investigation. However, the estimation, validation, and analysis of such models is far from trivial and involves sophisticated and often numerically sensitive methods. In this work we present the open-source Python package PyEMMA (http://pyemma.org) that provides accurate and efficient algorithms for kinetic model construction. PyEMMA can read all common molecular dynamics data formats, helps in the selection of input features, provides easy access to dimension reduction algorithms such as principal component analysis (PCA) and time-lagged independent component analysis (TICA) and clustering algorithms such as *k*-means, and contains estimators for MSMs, hidden Markov models, and several other models. Systematic model validation and error calculation methods are provided. PyEMMA offers a wealth of analysis functions such that the user can conveniently compute molecular observables of interest. We have derived a systematic and accurate way to coarse-grain MSMs to few states and to illustrate the structures of the metastable states of the system. Plotting functions to produce a manuscript-ready presentation of the results are available. In this work, we demonstrate the features of the software and show new methodological concepts and results produced by PyEMMA.



INTRODUCTION

Ever since the introduction of atomistic molecular dynamics (MD) simulation, the sampling problem has been one of the fundamental challenges in the field. The sampling problem limits our ability to simulate sufficiently long MD trajectories in which the rare, macromolecular transitions are well-sampled. Special-purpose supercomputers such as Anton[1] have gone a long way in addressing this issue, but such resources are only accessible to a few researchers. Pioneered by worldwide distributed computing projects such as GPUgrid[2] and folding@home,[3] the generation of high-throughput simulation data is now facilitated through distributed simulations on commonly available compute clusters, using either high-performance CPU[4−6] or GPU codes.[7−9] Today, microsecond-long trajectories of explicitly solvated protein systems can be produced in a few days on single GPUs. Computing clusters and HPC resources packed with hundreds to tens of thousands of GPUs are being installed around the globe, enabling more and more researchers to generate macromolecular simulation data on the order of hundreds of microseconds to milliseconds. The field has thus started to close the gap of time scales between experimental and simulation approaches, at least for moderately sized protein systems, and is on its way toward a more detailed, quantitative, and insightful analysis of molecular processes that may serve to meet challenges in biotechnology, nanotechnology, and pharmaceutics.

Scientists employing high-throughput MD simulation data are challenged by two problems:

(1) How can simulation data in many short trajectories from often arbitrarily chosen starting points be analyzed in a statistically correct manner? Ideally, given milliseconds of simulation data, can we say something about millisecond kinetics even though our individual trajectories are just microseconds or shorter?

(2) How should the analysis be made such that we can conveniently obtain an accurate yet humanly readable model of the metastable states, the thermodynamics, and the kinetics? We want to avoid subjective choices of reaction coordinates and steps of manual selection as much as possible.

A particularly successful class of models addressing these challenges are discrete state kinetic models, such as master-equation models, Markov models, or Markov state models. For simplicity we will subsume these terms under the abbreviation MSMs in this work. While Markov chain theory is over a century old, the systematic use of MSMs for molecular dynamics and related theory has been introduced by Schütte and colleagues in the late 1990s.[10] With the availability of significant computing power, these ideas have been adopted and brought to maturity by a few groups in the MD simulation community

Received: August 3, 2015

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

since the mid 2000s.[11−19] MSMs and related techniques have been successfully used to unravel the thermodynamics and kinetics of complex molecular processes such as protein folding,[20−26] protein−ligand binding,[27−32] peptide dynamics,[15,33−37] peptide aggregation,[38] protein conformation changes,[19,31,39−48] and self-assembly.[49]

The past few years have seen a large number of developments and improvements in methodology for the construction, validation, and analysis of kinetic models. To give some examples, activities include the determination of suitable collective coordinates[36,50−54] and metrics,[16,54−58] the development of efficient clustering methods,[16,57,59−62] methods for estimating transition and rate matrices,[14,18,21,62] their statistical errors,[13,17,63−69] and their systematic errors,[62,70,71] as well as model selection[72] and coarse-graining methods.[10,73−81] The analysis of Markov models with transition path theory (TPT) was developed in refs 81−84 and 20. Several contributions have been made toward systematic comparison of Markov models with experimental observables.[85−91] See refs 62 and 92 for an introduction and overview of MSM methods. Recently, new model types have been introduced, such as Markov transition models (MTMs) that operate on continuous state spaces and avoid clustering[93] and hidden Markov state models (HMSMs) of the underlying kinetics.[78,94] The variational principle and variational approach for conformation dynamics (VAC) introduced in ref 95 and further developed in refs 36 and 96 provide a generic framework for describing many types of kinetic models, including MSMs, MTMs, and TICA, and open up a new approach to model construction by formulating it as a problem of finding eigenvalues and eigenfunctions of a dynamical operator by a combination of basis functions. Finally, in refs 97−100 MSM estimation methods for data produced at different thermodynamic states (e.g., temperatures and bias potentials) have been introduced, thus opening up a way to integrate enhanced sampling simulations and direct MD simulations.

It should be clear from the preceding summary that construction, validation, and analysis of kinetic models such as MSMs from simulation data sets is a complex task and requires reliable and efficient software tools. In particular, many of the methods noted previously are numerically sensitive, such that ad hoc implementations can easily lead to difficulties. So far, two relatively complete software packages were available for that task: Emma's Markov model algorithms (EMMA)[101] and MSMBuilder.[61] In the present work we describe the software package PyEMMA that replaces the previous EMMA program and contains a large collection of highly usable and efficient analysis algorithms for the construction of kinetic models from MD simulation data. Since both PyEMMA and MSMBuilder are being actively developed, we refrain from doing a comparison of available features or performances, as such information will be quickly outdated.

PyEMMA is written mainly in Python and compatible with Python 2.7, 3.3, and 3.4. Computationally expensive routines are implemented in C. PyEMMA runs on all main operating systems and can process input from all commonly used MD trajectory formats, as well as coordinate input from text and Python binary files. PyEMMA is an open-source package distributed under the GNU Lesser General Public License (LGPL) version 3.0.[102] LGPL is a permissive license, i.e., PyEMMA can be used in both free and propriety software, but if modified versions of PyEMMA are distributed, the source code must be made available in order to contribute improvements and developments back to the open-source community. The PyEMMA code is hosted on https://github.com/markovmodel/pyemma.

The installation instructions, documentation, and tutorials are found at pyemma.org. PyEMMA is used by writing Python scripts, or interactively via IPython notebooks. These two complementary approaches allow for maximal flexibility and make command line programs dispensable. The notebooks used to create the results and figures of this work are available on http://pyemma.org.

Before going into methodological details, their implementation, and results on protein dynamics, we will give a brief overview of a typical PyEMMA analysis. As an example, we use the simple pentapeptide WLALL shown in Figure 1. Twenty-five trajectories of 500 ns each were analyzed by using all ϕ, ψ, and χ_1 angles as input coordinates. We then find collective coordinates that optimally resolve the rare-event transitions in the system by employing time-lagged independent component analysis (TICA, Figure 1a).[36,52,103] After clustering to 250 microstates, a MSM model lag time τ is selected at which the computed relaxation time scales are constant within statistical error (Figure 1b). The MSM was estimated at a lag of 1 ns, meaning that we can run such an analysis with trajectories that are of length 1 ns or longer and predict the long-time-scale kinetics from that. Because our trajectories are much longer, we can systematically validate the model predictions (Figure 1c) and then analyze the model. Figure 1d shows the free energy landscape over the two slowest collective coordinates. These coordinates resolve the separation of the state space into four long-lived macrostates (labels shown in panel d). Figure 1e shows the probability density of seeing that system in any microstate given that it is in one of the four metastable states. A slight overlap between plotted densities is natural due to the fuzziness of the assignment and due to a projection of higher dimensional data onto two dimensions. Finally, we perform a systematic coarse-graining of the 200-state MSM to a four-state model using hidden Markov modeling techniques.[78] Figure 1f shows the resulting coarse-grained rate network: Representative structures of the four long-lived states are shown, their equilibrium probabilities are indicated by the area of the orange discs, and the transition rates between states are given in ns^{-1}. Numbers in parentheses indicate the size of the 2σ (95%) confidence interval.

METHODS

Overview. Figure 2 shows an overview of the current PyEMMA packages. Packages that will be included in the near future are also listed to give an overview of the software organization.

coordinates. Loading and saving of molecular coordinates, selection of features and order parameters (e.g., internal coordinates) to be analyzed, data transformation and dimension reduction, clustering methods. Contains transformation methods such as PCA and TICA and clustering methods such as *k*-means.

msm. Discrete state kinetic models with a Markovian kernel, i.e., a means to compute the transition probability p_{ij} or transition rate k_{ij} between two discrete sets of state space. This includes discrete state and discrete time MSMs, master-equation models, and HMSMs. Contains maximum-likelihood and Bayesian estimators for all such models.

plots. Convenience plotting functions to visualize important model properties, such as free energy plots, plots of the kinetic

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

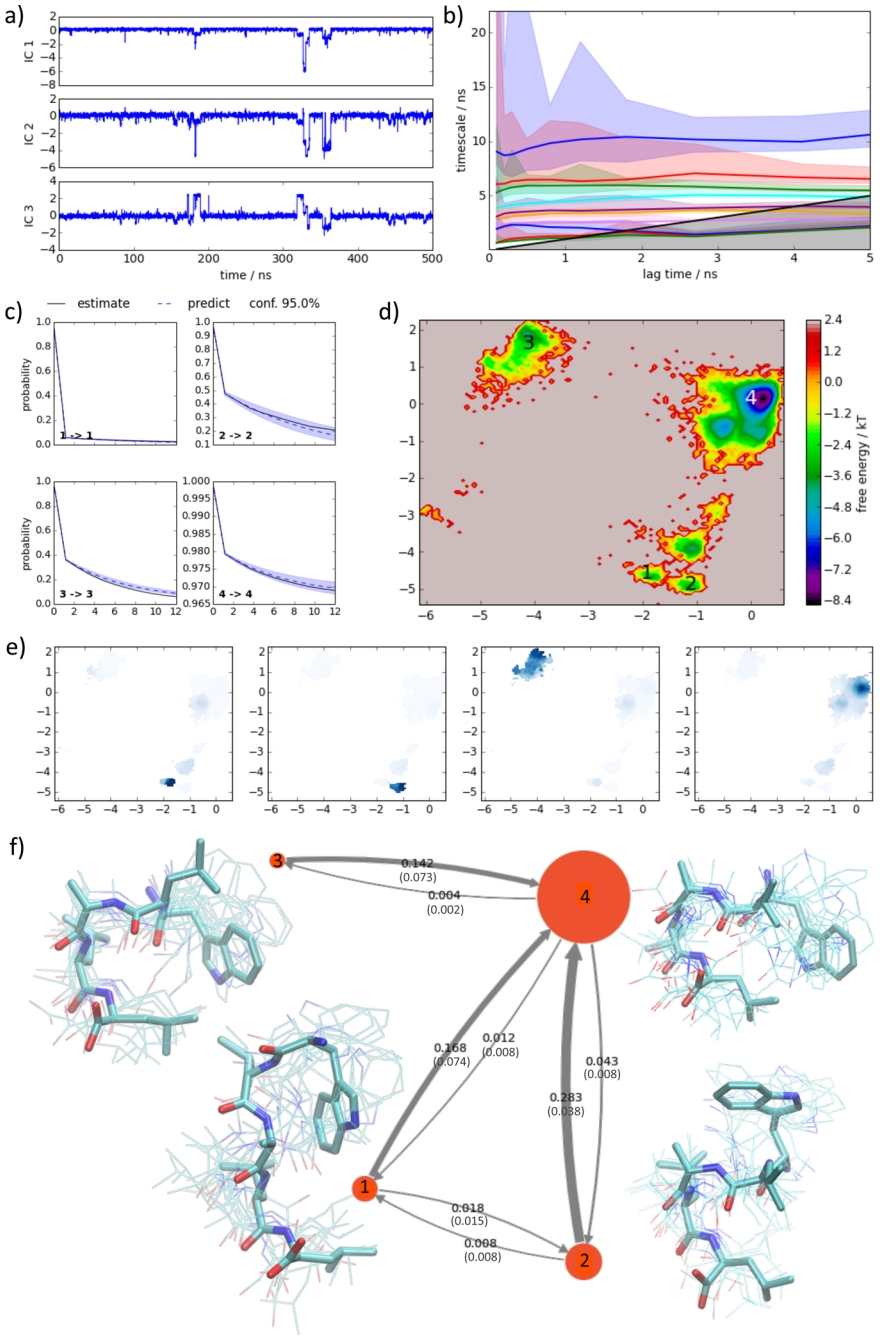


Figure 1. Illustrative Markov state model analysis of a pentapeptide. (a) Projection of a trajectory onto the slow collective coordinates (independent components, ICs) shows rare transitions between different metastable states. (b) Implied relaxation time scales as a function of the Markov model lag time τ shows that a lag time τ = 1 ns is suitable. Shown as shaded regions are 95% confidence intervals. (c) A Chapman−Kolmogorov test shows that the τ = 1 ns model accurately predicts the behavior on longer time scales. (d) The free energy landscape is computed from the MSM as a function of the two slowest ICs. The metastable states can be visually distinguished as free energy minima. (e) Probability distributions are given for the four longest living metastable states. The resulting assignment between energy minima and state numbers is shown in panel d. (f) The rate model is obtained from a hidden Markov model based coarse-graining of the MSM. Rates are given in nanoseconds; 10 structures have been sampled for each metastable state from the distributions shown in panel c. All subfigures except for the molecular structure images have been generated with PyEMMA. Molecular structures were rendered with VMD.[134]

relaxation time scales as a function of some model parameter, or a visual representation of the kinetic model as a network.

thermo (planned). Discrete state models with multiple thermodynamic states. This package is currently developed separately (https://github.com/markovmodel/thermotools) and will be integrated in PyEMMA in the near future. It will enable the analysis of data from biased simulations (e.g., umbrella sampling), or generalized ensemble simulations (e.g., replica-exchange MD), and combination of such data with direct MD via MSMs. Will contain estimators such as WHAM[104] and TRAM.[97,98,100]

mtms (planned). Markov transition models.[93] Continuous state kinetic models with a Markovian kernel, i.e., a means to compute the transition density $p(\mathbf{x},\mathbf{y})$ or rate density $k(\mathbf{x},\mathbf{y})$ between two continuous points of state space.

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

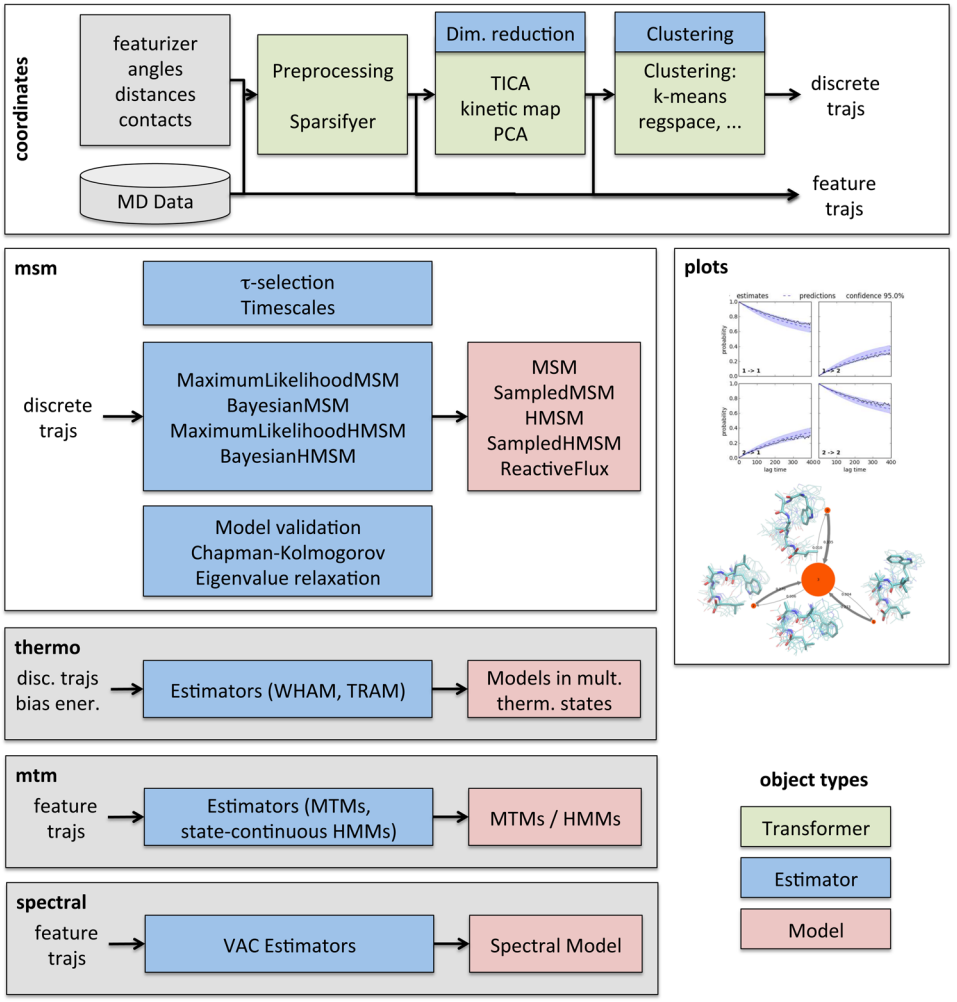


Figure 2. PyEMMA software structure: Representative packages and modules. White packages are currently available; gray packages will be included in subsequent releases and are shown here as part of the conceptual overview. The basic PyEMMA object types are Transformers that can perform on-disc or in-memory transformation of mass data and can be chained to form a processing pipeline, Estimators that produce kinetic models from state discrete or continuous trajectory data, and Models representing the results and providing analysis functionalities. All objects are made available through convenience functions on the package level; i.e., the user does not need any experience in object-oriented programming.

spectral (planned). Implementation of the variational approach of conformation dynamics (VAC)[95,96] for computing eigenvalue and eigenfunctions of dynamical operators, basis sets,[105] and tensor methods.[106] This package is currently developed separately (https://github.com/markovmodel/variational).

A typical analysis proceeds by first loading molecular topology information (e.g., from a pdb file) and then specifying the features or order parameters to be analyzed (e.g., intramolecular distances or contacts) and a list of the simulation trajectories one wants to analyze. Subsequently, one conducts a series of transformations, such as dimension reduction and decorrelation of the input features using TICA, and a discretization of the resulting collective coordinates using clustering. The resulting discretized trajectories are fed into an estimator, such as a Markov state model or a hidden Markov state model. This model is then analyzed for molecular or experimental observables of interest.

Technical Features and Software Design. Before describing the practical steps in kinetic model construction, we give a brief overview of the main technical and software features of PyEMMA:

(a) PyEMMA requires a Python installation version 2.7, 3.3, or 3.4 and runs under Linux, Mac OS X, and Windows. It can be installed through the package managers pip and conda, although we strongly recommend using conda. PyEMMA is part of the Omnia suite (http://omnia.md) and uses the Omnia software channel to resolve PyEMMA's package dependencies.

(b) The progress of all heavy calculations in PyEMMA can be optionally shown by progress bars.

(c) PyEMMA reads all major MD formats by integrating MDTraj[107] (http://mdtraj.org) and is thus compatible with all major simulation packages.

(d) All major functionalities of PyEMMA are available using convenience functions at the package level (e.g., `coordinates.tica` or `msm.estimate_markov_model`). Thereby the user does not need to be familiar with object-oriented design and programming.

(e) PyEMMA is built on a modular and flexible object structure (see Figure 2), consisting of data `Transformers`, model `Estimators`, and `Models`. The designs of these basic object types were motivated by our previous Java framework `StaLLOne` and the machine learning package scikit-learn (http://scikit-learn.org). Our `Transformers`

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

and `Estimators` are compatible with scikit-learn, thus enabling access to the powerful machine learning algorithms in scikit-learn. Note that MSMBuilder 3 (http://msmbuilder.org) uses a similar design, suggesting that it is a natural representation for the MD analysis problem.

(f) All heavy data processing, such as reading, dimension reduction, or clustering, is done through `Transformer` objects. Transformers can either work in memory to optimize speed or stream data from disk to minimize memory consumption. Multiple transformers can be chained to a data processing pipeline. This allows also very long trajectories and large protein systems to be analyzed.

(g) Processed data are fed into an `Estimator` which produces the results in the form of a Model. For example, `MaximumLikelihoodMSM` estimates a single MSM of maximum likelihood, while a BayesianMSM estimates a model containing a single representative (mean or maximum-likelihood) MSM and a sample of MSMs generated from the posterior distribution to compute uncertainties. All estimators are used in the same fashion, which minimizes the code in PyEMMA needed for generic tasks such as parallel processing.

(h) All Model objects share a common interface which makes it easy to implement generic tasks such as parameter estimation, sampling, and model validation. For example, MSMs and HMSMs both have the method `timescales`, and thus it is possible to compute the relaxation time scale as a function of the model lag time for both of them using the same code. The common interface allows the user to easily switch their analysis from one model type to the other, thus avoiding code duplication.

(i) All of PyEMMA's low-level MSM computations, such as transition matrix estimation, sampling, and decompositions, have been implemented in the subsidiary package `msmtools`. `msmtools` has only basic dependencies and can be installed independently in order to facilitate further software development. `msmtools` supports efficient and robust Matrix operations that are specialized for the problem at hand (e.g., reversible versus nonreversible Markov models), and both sparse and dense matrix calculations are supported.

Input Feature Selection. Given simulation data of a macromolecular system of interest, what are the first steps toward an informative Markov model or other kinetic models? In contrast to classical analysis methods, the user does not need to define a restrictive set of collective coordinates on which the analysis is performed—our aim is rather to learn which collective coordinates best characterize the rare-event transitions as part of the results of the analysis.

Nonetheless, one needs to define the input coordinates one is interested in working on. For a macromolecular simulation, candidates include the following:

(a) Cartesian coordinates of a subset of selected atoms (such as protein Cα's, backbone atoms, or heavy atoms of the solute) after rototranslational alignment. This coordinate set may be applied if a large part of the solute structure is stable such that a meaningful reference structure exists.

(b) Flexible torsions, such as protein backbone ϕ/ψ angles or side chain torsions. Usually these angles are not used directly but rather their cos and sin values (thus doubling the number of angle coordinates), because, after the cos/sin transform, standard operations such as calculating and subtracting the mean can be performed.[55]

(c) Distances between pairs of protein Cα's or solute heavy atoms.

(d) Contacts between residues or other relevant chemical groups.

(e) Coordinates characterizing the solvation shell such as coordination numbers and density coordinates.[108−110]

(f) Minimal root-mean-square deviation (minRMSD) with respect to given reference structures, e.g., a protein in a folded state.

In PyEMMA, the coordinates to be analyzed are called *features*. Featurization consists of transforming every Cartesian coordinate frame $\mathbf{r}(t) \in \mathbb{R}^{3n_{\mathrm{atom}}}$ into a feature vector $\mathbf{x}(t) \in \mathbb{R}^{n_{\mathrm{feat}}}$ given as a one-dimensional array whose size equals the number of features:

$$\underset{\text{Cartesian coordinates}}{\mathbf{r}(t)} \quad \rightarrow \quad \underset{\text{features}}{\mathbf{x}(t)}$$

If Cartesian coordinates of n_{atom} atoms are used, $\mathbf{x}$ will be a vector of size $3n_{\mathrm{atom}}$. Many standard features are readily available for selection, but the user can also add custom features by implementing the preceding mapping in a custom function and passing this function to the Featurizer.

The first step of an analysis consists of loading topology information, e.g., from a pdb file, and then defining a `Featurizer` object that stores a list of all features that will be used in the analysis. This feature object will be employed whenever coordinates are read from disk. Feature mapping will always be applied to small chunks of the simulation trajectory. In this way, memory overflow is avoided even when very large feature vectors are employed.

Features can be simply concatenated; i.e., one could have a feature vector containing a mix of Cartesian coordinates, intramolecular distances, and angles, etc. In principle, one can select all of the features described previously, even if they contain redundant information. However, such an approach is computationally unfavorable. In the results (Figure 4), we discuss how the TICA method discussed later can be employed to select suitable features.

Coordinate Transformation and Dimension Reduction. The feature vectors selected previously can be very high dimensional. For example, there are nearly 5000 Cα distances in a small 100 residue protein. Trying to discretize very high dimensional spaces by clustering methods is inefficient and tends to produce low-quality discretizations.[54] Furthermore, features such as pairwise distances are highly redundant and thus by themselves not very interesting—in the preceding example the Cα positions are fully described by 300 coordinates.

In PyEMMA, we recommend to first reduce the dimension by conducting a linear coordinate transformation. In linear transformation methods, one seeks a set of basis vectors $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_m]$, where each vector $\mathbf{u}_i$ is a collective coordinate with m components. The new coordinates are given by the following projection:

$$\mathbf{y}(t) = \mathbf{U}^{\mathrm{T}}\mathbf{x}(t)$$

and have thus been reduced to n_{red} dimensions which is usually much smaller than n_{feat}. A common approach to finding such a basis is PCA.[111,112] In PCA, one first subtracts the means from each feature vector, obtaining $\tilde{\mathbf{x}}(t)$, and then computes the instantaneous correlation matrix with elements $c_{ij}(0) = \langle \tilde{x}_i(t)\, \tilde{x}_j(t) \rangle_t$. The principal directions $\mathbf{u}_i$ are obtained after solving the eigenvalue problem:

$$\mathbf{C}(0)\mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$$

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

The eigenvalues σ_i^2 measure the variance of the data along the principal directions:

$$\langle (y_i(t))^2 \rangle_t = \sigma_i^2$$

PCA transforms the data into an orthogonal basis, such that the new coordinates are uncorrelated for $i \neq j$:

$$\langle y_i(t)\, y_j(t) \rangle = 0$$

This property is very useful, because it transforms the possibly redundant feature set x into a nonredundant linear basis. Thus, PCA can remove many input dimensions by finding linear or near-linear dependencies among input coordinates and assigning them to directions with eigenvalues σ_i^2 near zero. The fraction of the variance retained by this dimension reduction is

$$V_m = \frac{\sum_{i=1}^{m} \sigma_i^2}{\mathrm{TV}}$$

where the total variance is $\mathrm{TV} = \sum_{i=1}^{n_{\mathrm{feat}}} \sigma_i^2$. It is well-known that PCA is optimal within linear transformation methods in terms of maximizing the retained variance V_m.

However, our aim is not necessarily to retain variance but to describe the molecular kinetics. Thus, we are most interested in preserving the slow motions, rather than the large-amplitude motions. As an example consider an unstructured peptide that has very flexible ends but undergoes a rare-event concerted torsion transition in its center. We would like to identify the rare-event transition rather than the fast and high-variance fluctuations of the termini.

Toward this end, the standard dimension reduction method in PyEMMA is the time-lagged independent component analysis (TICA), originally developed in ref 103 and introduced to the analysis of molecular kinetics in refs 36 and 52. It has been shown in ref 36 that TICA implements the variational approach of conformation dynamics[95] and is optimal among linear methods in recovering the slow reaction coordinates and their relaxation time scales. TICA is related to PCA but also uses a time-lagged correlation matrix $\mathbf{C}(\tau)$ with elements $c_{ij}(\tau) = \langle \tilde{x}_i(t)\, \tilde{x}_j(t+\tau) \rangle_t$. One then solves the generalized eigenvalue problem

$$\mathbf{C}(\tau)\mathbf{u}_i = \mathbf{C}(0)\, \lambda_i(\tau)\mathbf{u}_i$$

thus obtaining independent component directions $\mathbf{u}_i$ which are linear approximations to reaction coordinates of the system. Instead of having maximal variance, they have maximal autocorrelation, as measured by the eigenvalues λ_i:

$$\langle y_i(t)\, y_i(t+\tau) \rangle_t = \lambda_i(\tau)$$

Furthermore, we still retain the PCA property that the new coordinates are uncorrelated for $i \neq j$:

$$\langle y_i(t)\, y_j(t+\tau) \rangle = 0$$

Similar to that in PCA, we can define a kinetic variance.[54] The fraction of the kinetic variance retained by this dimension reduction is

$$\mathrm{KV}_m = \frac{\sum_{i=1}^{m} \lambda_i^2}{\mathrm{TKV}} \quad (1)$$

where the total kinetic variance is

$$\mathrm{TKV} = \sum_{i=1}^{n_{\mathrm{feat}}} \lambda_i^2$$

Note that the squared eigenvalues are in the range [0, 1], where values near 1 occur for slow processes and values near 0 occur for fast processes. Thus, roughly speaking the TKV measures the total number of slow processes found in the data.

We recommend TICA as a standard method for coordinate transformation and dimension reduction. We suggest selecting the retained dimension m using eq 1, i.e., such that the projection retains a user-defined fraction (e.g., 0.95) of the kinetic variance. Furthermore, we suggest scaling the TICA coordinates to obtain a kinetic map,[54] as the resulting coordinates define a metric space in which geometric distances are proportional to kinetic distances and are thus optimally prepared for geometric clustering. The aforementioned settings are defaults in PyEMMA.

Discretization. Many kinetic models, including MSMs and HMSMs, are built on discrete state spaces. Therefore, the transformed input data must be discretized. Since the dimension of the TICA subspace can still be high (often in the range of a few 10s of dimensions), employing any kind of grid discretization is unfeasible. A discretization that can be employed in high-dimensional state spaces is the Voronoi discretization,[113] where a set of k centers is determined, and then each data point $\mathbf{y}(t)$ is assigned to the closest center. At this point we usually use the standard Euclidean distance to measure distances, although other options such as minimal RMSD exist.[16] If TICA-based kinetic maps were used as described at the end of the last section, these distances approximate the kinetic distances.

The locations of the k centers are determined by a clustering method. The recommended method in PyEMMA is to use k-means[114] in combination with the k-means++ initialization procedure.[60] This approach is relatively fast and generates reproducible results with only a few (one to five) k-means iterations. Because the full k-means algorithm is linear in the amount of samples and centers, a minibatch[115] version of k-means has been implemented to support fast clustering of very large data sets. PyEMMA has efficient C implementations for these three methods.

For completeness, PyEMMA also has implementations of regular-space clustering and uniform-time clustering as discussed in ref 62. In regular-space clustering, points are distributed such that the distances to the nearest neighbors are approximately uniform, thus giving rise to an approximately regular coverage of the sampled space. We discourage the use of the regular-space algorithm as it tends to place cluster centers on outliers, which can lead to models that are poorly reproducible and have unnecessarily large statistical error. The uniform-time clustering algorithm samples cluster centers at regular time intervals. This approach leads to cluster centers being concentrated in the most stable region, and very few are assigned to transition states or to less populated metastable states. Such a cluster distribution is unfavorable for the estimation of MSMs[70] and is therefore discouraged as well. The k-means algorithm produces a balanced clustering which is "in between" the other two extreme clustering algorithms: k-means clusters are more regular in space than uniform-time clustering but selects less outliers than regular-space clustering.

Although implicit pairwise norms such as minRMSD[16,62,101] are available, we discourage the usage of clustering algorithms that employ pairwise minRMSD. In our experience, such approaches are computationally expensive and produce lower quality discretizations than the recommended approach of combining TICA and k-means.

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

An open question is how to select the number of clusters in k-means. Reference 72 suggests picking the number of clusters using cross-validation using the partial eigenvalue sum of the estimated MSM as a score which should be maximized according to the variational principle.[95] However, there was a large range of k within which the results were indistinguishable within statistical error. This coincides with our experiences, where the choice of k has a relatively modest effect compared to the model lag time τ that can have a dramatic effect (see later discussion). Therefore, we suggest the simple approach to choose the number of clusters as $\sqrt{N}$, where N is the total number of available samples in the input trajectories. This gives a number of clusters in the few hundreds to a few thousands for typical data sizes. $\sqrt{N}$ is the default parameter for clustering algorithms in PyEMMA that use a predefined number of clusters.

Estimation of Markov State Models. Markov state models (MSMs) are relatively simple and yet powerful kinetic models. An MSM approximates the molecular kinetics by a matrix of conditional transition probabilities among discrete states. Suppose the state space has been discretized into discrete trajectories $s(t)$ jumping between n microstates (n equals the number of cluster centers k in the previous section). The parameters of an MSM are the conditional transition probabilities between microstates at τ:

$$p_{ij}(\tau) = \mathbb{P}(s(t+\tau) = j | s(t) = i)$$

The MSM *predicts* kinetics at longer time scales in term of the powers of the transition matrix (Markov property):

$$\mathbb{P}(s(t+k\tau) = j | s(t) = i) = [\mathbf{P}^k(\tau)]_{ij}$$

Much of the power of MSMs comes from the fact that they use *conditional* transition probabilities. In order to parametrize an MSM, one thus only has to estimate the probability of going to j given that we start in state i. In other words we can use many short trajectories starting from different states i, rather than a single long trajectory visiting all of them (with the equilibrium frequency). We do not need to know in advance what the equilibrium probability of state i is—rather the MSM predicts that this equilibrium probability $\boldsymbol{\pi} = (\pi_i)$ in terms of its stationary vector:

$$\boldsymbol{\pi}^{\mathrm{T}} = \boldsymbol{\pi}^{\mathrm{T}}\mathbf{P}(\tau)$$

Thus, the MSM can be thought of as reweighing trajectories whose starting points have been selected from an out-of-equilibrium distribution. By virtue of this property, MSMs are very useful in reconciling short and independently generated trajectories and have thus been a method of choice for distributed simulation projects such as GPUGrid[2] and folding@home.[3]

PyEMMA currently has two estimator classes for MSMs: `MaximumLikelihoodMSM` seeks the maximum of the transition matrix likelihood:

$$\mathbb{P}(\mathbf{C}(\tau)|\mathbf{P}(\tau)) \propto \prod_{i,j=1}^{n} p_{ij}^{c_{ij}(\tau)}$$

The class `BayesianMSM` generates a sample of transition matrices $\mathbf{P}(\tau)$ according to the posterior distribution:

$$\mathbb{P}(\mathbf{P}(\tau)|\mathbf{C}(\tau)) \propto \mathbb{P}(\mathbf{P}) \prod_{i,j=1}^{n} p_{ij}^{c_{ij}(\tau)}$$

Here $c_{ij}(\tau)$ is the number of times a trajectory has been observed in state i at time t and in state j at time $t + \tau$. The maximum-likelihood estimator takes all of these transition counts into account by default, as it will converge to the correct transition matrix irrespective of the statistical correlations in subsequent count windows such as $t \rightarrow t + \tau$ and $t + 1 \rightarrow t + 1 + \tau$.[62] For the Bayesian estimator, by default only statistically effective counts will be used that are estimated based on computing the statistical inefficiency between transition counts.[69,116]

A crucial parameter for the accuracy of the Markov model is the lag time τ. τ should be picked such that the implied relaxation time scales

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

are approximately constant within the statistical error,[11] where $\lambda_i(\tau)$ is the ith largest eigenvalue of the MSM estimated at τ. To facilitate this, PyEMMA uses the meta-estimator `ImpliedTimescales` which can compute relaxation time scales as a function of the lag time for any estimator/model pair that can compute time scales, including MSMs. If a BayesianMSM is employed, then these time scales estimates will automatically have error bars as shown in several figures.

To illustrate, parts c–f of Figure 3 show results of an MSM estimation and validation for a double-well potential using a good and a poor discretization, respectively. The implied time scales and their error bars can be computed for a range of lag times and then visualized with one simple command. The implied time scales plot is not sufficient as a model test. Please see later how to systematically validate a model.

Both the maximum-likelihood and the Bayesian estimators can be run in either of three modes:

(1) *Nonreversible estimation*: $\mathbf{P}(\tau)$ is a transition matrix ($p_{ij} \geq 0$, $\sum_j p_{ij} = 1$), with no other constraints. In this case the maximum-likelihood estimator is analytically available ($\hat{p}_{ij} = c_{ij}/\sum_k c_{ik}$), and the posterior can be sampled by drawing from statistically independent Dirichlet distributions.[13] Nonreversible estimation is the fastest option but not recommended for MD simulation, unless one is deliberately simulating a system that violates detailed balance.

(2) *Reversible estimation*: $\mathbf{P}(\tau)$ is reversible with respect to its equilibrium distribution ($\pi_i p_{ij} = \pi_j p_{ji}$). This is the default option as reversibility is physically meaningful for most MD setups, reduces the statistical uncertainty, and permits a range of analysis methods of the equilibrium kinetics to be applied.[69] Bayesian estimation is conducted using the highly efficient Gibbs sampling procedure described in ref 69.

(3) *Reversible estimation with fixed* π: Same as the preceding mode, but the equilibrium distribution is an input. This option is useful for combining free MD simulations with knowledge of the equilibrium distribution obtained from other methods such as umbrella sampling or replica-exchange MD in order to estimate the kinetics of rare events that cannot be sampled in the unbiased MD simulations.[117]

Any discrete model will be constructed on a so-called *active set* of states, usually a proper subset of all observed states. The standard use of an active set is to mark the subset of states that are connected through trajectory data in both directions. If state labels are empty, for example because the clustering algorithm has not assigned any MD configurations to them, or if states are not reversible connected such as state 0 in the trajectory [0, 1, 2, 1, 2, 1], these disconnected states will be excluded in the

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

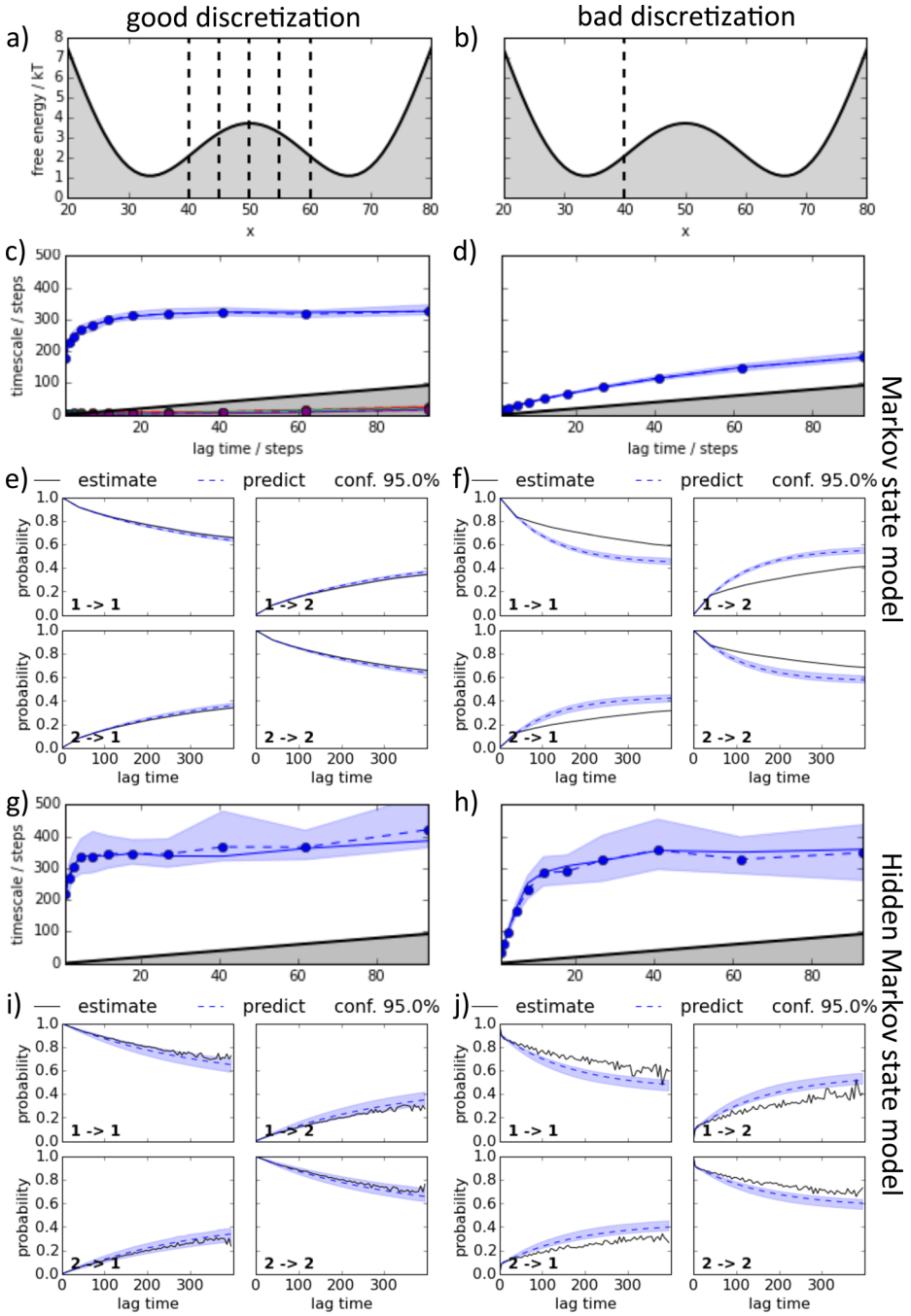


Figure 3. Lag time selection and model validation: Comparison of implied time scale plots and generalized Chapman−Kolmogorov tests for a metastable two state system (a, b) with good discretization (left) and bad discretization (right). The model quality of Markov state models (c−f) is compared to hidden Markov state models (g−j). Estimates are shown with 95% confidence intervals (blue). Generalized Chapman−Kolmogorov tests are shown for MSM estimates at lag τ = 20 steps and HMM estimates at lag τ = 5 steps.

estimation procedure and the subsequent model analysis. Note that when estimating with respect to a given equilibrium distribution, the data only need to be simply connected; i.e., in this case the trajectory [0, 1, 2, 1, 2, 1] would have the active set {0, 1, 2}.

Estimation of Hidden Markov State Models. A limitation of MSMs is that even with a relatively good discretization, the required τ to achieve a model with acceptable quality may be large. It has been shown in ref 118 that the MSM implied time scales only converge linearly in τ, and this poor convergence behavior is clearly seen, e.g., in Figure 4b. However, when τ exceeds the time scales of interest, or the affordable trajectory lengths, then an MSM is of little use.

An interesting alternative may be to construct a hidden Markov model on the discrete state space, here termed hidden Markov state model (HMSM). Hidden Markov models[119,120] are widely used in machine learning.[121] However, it has only been shown recently[78] that HMMs can be excellent approximations of the kinetics on discretized molecular state spaces. If a time scale separation exists after m eigenvalues—and typically a situation where t_m is twice t_{m+1} is sufficient for that—then the transition probability between n microstates at any lag time $k\tau$

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

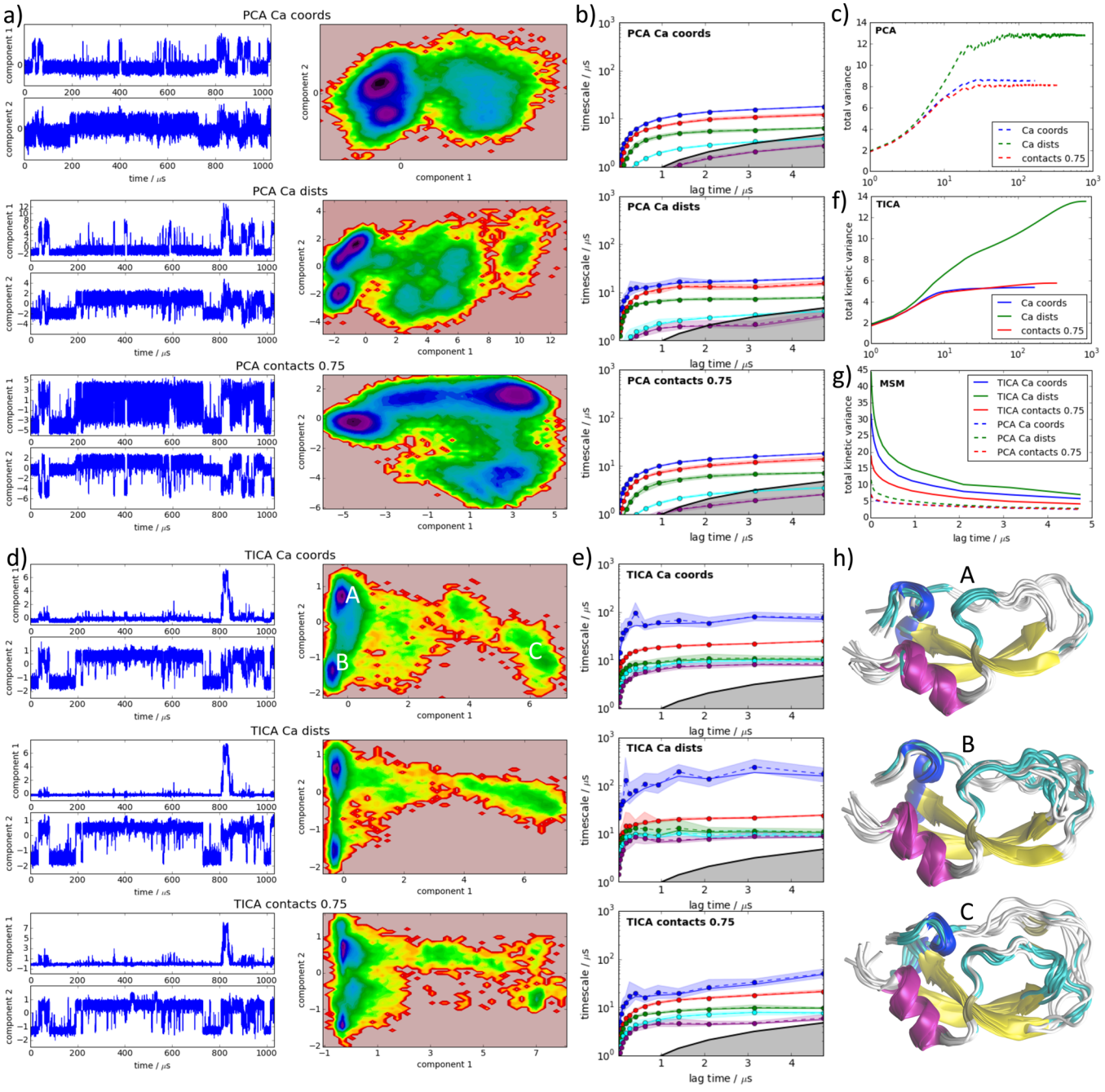


Figure 4. Input feature selection and transformation method. Three types of input features (rototranslationally oriented Cα coordinates, Cα−Cα distances, and residue contacts based on closest heavy-atom distances) and two transformation methods (PCA, TICA) are compared in their abilities to capture the slow processes in BPTI conformational dynamics based on a 1 ms simulation trajectory.[1] (a, d) Projection of the trajectory onto the two largest principal or independent components (left) and density plot of both coordinates in log-space (density maxima, blue/purple; low densities, yellow/red). (b, e) Implied relaxation time scales of MSMs built from these data, always using k-means with 100 cluster centers. (c, f) Cumulative variance captured by PCA and the cumulative variance captured by TICA using the three different input features. (g) Total kinetic MSM variance plotted as a function of lag time for different input features and projection methods. (h) Three representative sets of structures for the metastable states indicated as A, B, and C in panel d.

can be accurately described by the so-called projected Markov model:

$$\mathbf{P}(k\tau) = \mathbf{\Pi}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{\Lambda}^{k}(\tau)\mathbf{Q}$$

where $\mathbf{\Lambda}(\tau) \in \mathbb{R}^{m\times m}$ are the first m true eigenvalues, $\mathbf{\Pi} = \mathrm{diag}(\pi_1, ..., \pi_n)$ is the diagonal matrix of stationary probabilities, and $\mathbf{Q} \in \mathbb{R}^{m\times n}$ is a matrix of true eigenfunctions projected onto the n microstates.[78] While the preceding formulation is not immediately useful, it can be approximated by

$$\mathbf{P}(k\tau) = \mathbf{\Pi}^{-1}\chi^{\mathrm{T}}\tilde{\mathbf{\Pi}}\tilde{\mathbf{P}}^{k}(\tau)\chi$$

which is a hidden Markov model.[78] Here, $\tilde{\mathbf{P}} \in \mathbb{R}^{m\times m}$ is a small transition matrix between m metastable states of the system, and $\chi \in \mathbb{R}^{m\times n}$ contains probability vectors in its rows, each

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

of which determines the probability that a metastable state will appear in one of the n microstates.

Consider the double-well system in Figure 3 as an example. In the poor discretization (right column), a MSM cannot describe the kinetics accurately for any acceptable lag time τ (panel d). The problem lies in the fact that the dynamics are very non-Markovian on this particular discretization. However, a HMM can describe the kinetics on the poor discretization very well using short lag times (panel h). The reason is that the system itself is excellently described by a transition matrix between two states, because there is only one slow time scale. The effect of poor discretization can be captured by the output probability matrix χ, which in the example of Figure 3 is approximately

$$\chi = \begin{bmatrix} 0.88 & 0.12 \\ 0 & 1 \end{bmatrix}$$

This matrix corrects the fact that the poor discretization shown in Figure 3b has mixed up the true metastability structure of the double well: The true metastable state 1 (left well) shows up mostly (88%) in the left bin, but also with considerable probability (12%) in the right bin. On the other hand, the right well almost always shows up in the right bin. This example illustrates why the concept of an output probability is meaningful—it simply corrects for problems introduced by poor discretizations or poor choices of the input features.

Since a HMSM is a generalization of an MSM, all physicochemically relevant quantities that can be computed from an MSM can also be computed from an HMSM (see later discussion).

Like for MSMs, PyEMMA has two Estimators for hidden Markov state models, the `MaximumLikelihoodHMSM` that employs the standard Baum–Welch algorithm[119,120] and a `BayesianHMSM` estimator that samples the posterior distribution of HMSMs by generating pairs of transition matrices and output matrices, $(\tilde{\mathbf{P}}(\tau), \chi)$ using the Gibbs sampler described in ref 122. Both estimators employ the bhmm package (https://github.com/bhmm) that is installed as a dependency with PyEMMA. Using the BayesianHMSM estimator, all HMM quantities can be computed with statistical error estimation.

Kinetic Model Validation. Kinetic models are approximations of the full phase-space dynamics and therefore have nonzero systematic error. For example, consider a hypothetical simulation where we start a large number of long simulations in conformation A and measure the time each simulation takes to reach conformation B, and average those numbers to obtain the mean transition time. Alternatively, we build an MSM from a large amount of simulation data (either short or long simulations) and compute the mean first passage time directly from this model. While the latter approach has numerous advantages, the MSM estimate might have a small but systematic error. This systematic error will not vanish in the large data limit. It depends on (i) the type of model used, (ii) state space discretization, and (iii) lag time τ.

We therefore suggest validating any kinetic model before using it for analysis. Model validation is considered successful when the model, estimated at lag time τ, is able to predict estimates performed at longer time scales $k\tau$ within statistical error. The canonical approach for MSMs is to test some formulation of the Chapman–Kolmogorov equation

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

The preceding equation holds exactly if there is no systematic or statistical error. When model estimates are made, we test if the systematic error is acceptable by assessing whether the left- and right-hand sides of (3) are equal within their statistical errors.

We have generalized this idea to a class of time-lagged model tests that are applicable to all kinetic models in PyEMMA, including MSMs and HMSMs. This is possible because all of the existing (and also all of the currently planned) kinetic models can be estimated at different lag times and can be used to make predictions at arbitrary longer times $k\tau$, with $k \geq 1$. f denotes some function of interest given for the model $M(\tau)$, and $\tilde{f}^{(k)}[M(\tau)]$ denotes the model prediction of f for a longer time $k\tau$; then we compare whether

$$\underset{\text{estimation}}{f[M(k\tau)]} = \underset{\text{prediction}}{\tilde{f}^{(k)}[M(\tau)]} \tag{4}$$

holds within the error for multiple values of k.

Currently, two such tests are implemented in PyEMMA: (1) The eigenvalue decay test, closely related to the mode correlation test, e.g., described in ref 18, compares the predicted and estimated decay of the model eigenvalues:

$$\lambda_i(k\tau) = \lambda_i^k(\tau)$$

for different model eigenvalues i. This test is very generally applicable to any model of the equilibrium kinetics, including MSMs with transition or rate matrices, HMSMs, Markov transition models,[93] and purely spectral models obtained from the variational approach[95,96] such as TICA.

For models that can propagate probability densities in time, such as MSMs, HMSMs, and Markov transition models, we can perform more specific and sensitive tests. To this end, we have implemented a generalized Chapman–Kolmogorov test that compares prediction and estimation of the probability of being in set J at time $k\tau$ given that we start in a local equilibrium of set I at time 0:

$$p_{IJ}(k\tau) = \tilde{p}_{IJ}^{(k)}(\tau)$$

Although arbitrary sets can be chosen, we use the following as default. For MSMs, we use fuzzy sets representing the metastable states and their initial distribution computed using PCCA++[77] and Bayesian inversion.[78] For HMMs, one usually has a transition matrix between only a few metastable states, and thus the Chapman–Kolmogorov test (3) can be applied directly on the hidden transition matrix.

In PyEMMA the preceding tests are generalized and automatized. Given an estimated MSM or HMSM, one can simply call its function `cktest` and visualize the results via the corresponding plotting function. If the Estimator supports the estimation of error bars (e.g., `BayesianMSM` and `BayesianHMSM`), these will be automatically computed on the predicted quantities, and if desired also on each of the estimates.

Illustrations are shown in Figure 3 for a double-well potential with good (left) and poor (right) discretizations, using either MSMs estimated at a lag time of $\tau = 20$ (panels e and f) or HMMs estimated at a lag time of $\tau = 4$ (panels i and j). In both cases, the model predictions at long time scales are consistent with the corresponding estimates for the good discretization, but the validation fails for the poor discretization. In the first case, the model can be accepted and analyzed. In the second

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX–XXX

case, one has to use a finer discretization and/or a longer lag time.

Kinetic Model Analysis. A large part of PyEMMA's convenience comes through the wealth of analysis functions offered by MSMs, HMSMs, and other model objects. The following quantities can be easily obtained once such a kinetic model has been estimated:

(a) Stationary distribution of the model, π. Calculation of π is extremely fast by using inverse iteration.[123] In PyEMMA, every model that has a stationary distribution can be used to recompute the statistical weight of the MD configurations used as an input. With this feature, the user can easily compute equilibrium expectation of any observables of interest that can be computed from MD frames, for example spectroscopically measurable functions, even if the simulation trajectories were not generated from an equilibrium distribution. Another application is the computation of MSM-reweighed free energy surfaces as shown in Figure 1d.

(b) Eigenvalues $\lambda_i(\tau)$ and left or right eigenvectors. Eigenvalues are related to the relaxation time scales (2) and thus to time scales or rates measurable in kinetic experiments.[85] Eigenvectors indicate the structural changes occurring at these time scales.[62,85] Because eigenvalues and eigenvectors are needed for various other calculations, we have optimized the calculation of eigenvalues and eigenvectors to automatically employ specialized algorithms. For example, the eigenvalue problem for reversible Markov models is formulated as a generalized symmetric eigenproblem which can be solved much faster than a nonsymmetric eigenproblem.

(c) Metastable (long-lived) states and their probability distributions using the PCCA++ algorithm for MSMs[77] or the output probabilities of the HMSMs. Metastable state distributions are shown in Figure 1e. They are employed to generate representative structures of these metastable states, as shown, e.g., in Figure 1f and Figure 4h.

(d) Mean first passage time t_{AB}, or inverse transition rate, to go from any set of states A to any other set of states B.[13] Of particular interest is the mean first passage time between metastable states.

(e) Committor probabilities, also known as probability of folding or splitting probability.[124,125] The forward committor q_i^+ relative to educt states A and product states B is the probability of reaching B before A when starting from state i. The backward committor q_i^- is the probability to have come from A last rather than from B. These properties are ideal reaction coordinates for the $A \rightarrow B$ transitions and are the key to computing transition pathways (see later discussion).

(f) Quantities measurable by kinetic experiments, such as time-relaxation function, that are relevant for temperature-jump, pressure-jump, pH-jump, or rapid mixing experiments.[86,88] Time-correlation functions are relevant for fluorescence correlation and dynamical neutron scattering experiments.[85,89]

(g) Spectral representation of such computational experiments, also called dynamical fingerprint.[85,88,89] Dynamical fingerprints are useful for designing new experiments,[85] e.g., by predicting where to place a spectroscopic probe to maximize the signal/noise ratio when measuring a specific relaxation process.

All of the preceding quantities can be conveniently computed from the MSM or HMSM objects obtained after estimation. This ability relies on using functions of PyEMMA's low-level MSM package `msmtools` (https://github.com/markovmodel/msmtools).

Transition Path Theory. Transition path theory (TPT) is a framework to compute fluxes from a set of source or educt states A to a set of sink or product states B and to thus extract information about the structural mechanism and the kinetics of the $A \rightarrow B$ transition. TPT has been originally introduced for continuous-space Markov processes in[81] and has been formulated for rate matrices in refs 83 and 84 and for transition matrices in ref 20. TPT is useful for studying processes such as protein folding[20,22,126] and protein−ligand binding.[27,31]

In PyEMMA, TPT is represented by a `ReactiveFlux` model that can be generated from an MSMs or HMSMs. A ReactiveFlux contains a set of network fluxes between pairs of microstates. The gross flux, the average number of transitions along an edge $i \rightarrow j$ as part of the transition $A \rightarrow B$, is computed as

$$f_{ij} = q_i^- \pi_i p_{ij} q_j^+$$

where q_i^- and q_i^+ are the backward and forward committor probabilities and $\pi_i p_{ij}$ is the equilibrium flux. The net fluxes are

$$f_{ij}^+ = \max\{0, f_{ij} - f_{ji}\}$$

These net fluxes define a directed graph from A to B states. PyEMMA provides graph-based algorithms to decompose f_{ij}^+ into individual pathways with their corresponding contribution[83] and algorithms to coarse-grain the flux onto sets of interest, such as the metastable sets.[20] An application of TPT is shown in Figure 5 (see Results).

Coarse-Grained Markov Models. There is a great interest in having a model with few states that contains the essential structural, thermodynamic, and kinetic information on the molecular system under investigation. The user wants to get a comprehensive illustration of important key facts: which metastable states do exist, what are their structural characteristics, what are their equilibrium probabilities or free energies, which of them are kinetically connected, and by which transition rates? In PyEMMA we provide a systematic and rigorous path toward obtaining such a picture.

The number of MSM microstates generated by clustering is usually in the range of hundreds to thousands. While such a fine discretization is important to compute quantitatively accurate information, such as the mean first passage time from one set of states to another, or the time-relaxation behavior in a temperature-jump experiment, it hardly qualifies as a humanly understandable representation of the system.

Suppose one is given a discretization of the state space into n microstates, and a MSM with stationary distribution π_i and transition matrix $p_{ij}(\tau)$. Now one wants to group the microstates into $m \ll n$ sets of microstates, which we shall call macrostates. How does one obtain a coarse-grained model that correctly maintains the structural, thermodynamic, and kinetic information? Simply by grouping structures and adding up equilibrium probabilities:

$$\pi_I = \sum_{i \in I} \pi_i$$

However, coarse-graining the kinetics is not trivial. The direct summation of fluxes,

$$p_{IJ}(\tau) = \frac{1}{\pi_I} \sum_{i \in I} \pi_i \sum_{j \in J} p_{ij}(\tau)$$

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

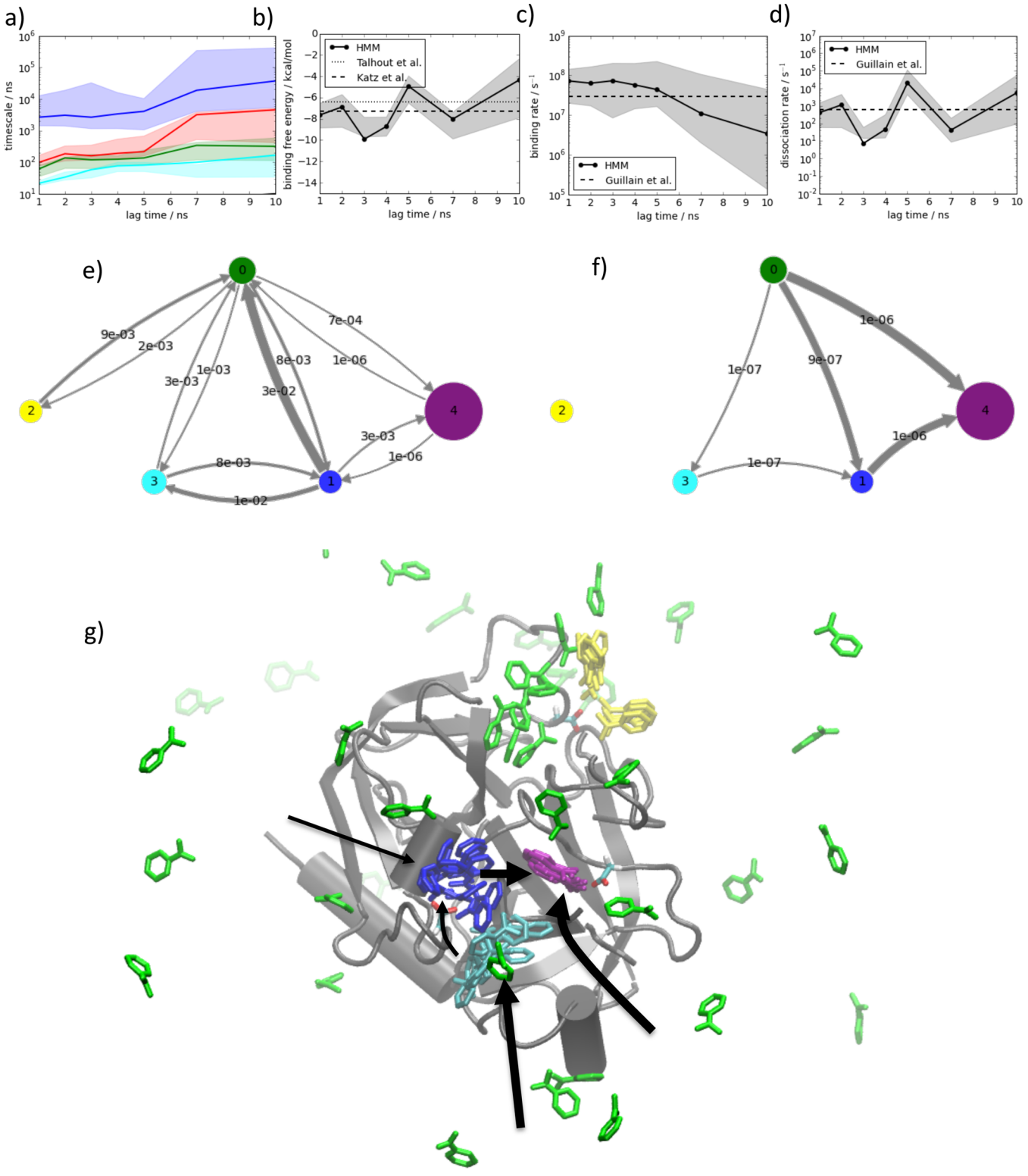


Figure 5. Hidden Markov state model analysis for protein−ligand binding—trypsin−benzamidine. (a−d) Model quantities reported as a function of τ: (a) implied relaxation time scales; (b) binding free energy, measured between bound state 4 and unbound state 2, compared to experimental values; (c, d) dissociation and association rates, compared to experimental values. (e) Rate network (rates in ns^{-1}) between metastable states. Sizes of low-populated states are enhanced for visual clarity. (f) Transition path theory fluxes from unbound to bound state. (g) Structures sampled from the HMM metastable state distributions. Benzamidine positions are shown according to state color (compare to panels e and f). The binding fluxes of panel f are indicated by black arrows.

is correct for lag time τ, but usually fails to be an accurate Markov model as it will no longer make accurate predictions for longer time scales. For example, macroscopic kinetic properties such as the mean first passage time from one set I to another set J will be different when $p_{ij}(\tau)$ or $p_{IJ}(\tau)$ are employed. The fine-grained MSM with an n-state discretization and transition

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

matrix $p_{ij}(\tau)$ has been validated, e.g., using a generalized Chapman−Kolmogorov test (see previous discussion). But by the process of coarse-graining the MSM to m states, we change the discretization, such that the validation does not hold anymore for the coarse-grained model. We could re-estimate and validate a MSM on the coarser state space, but this approach often requires unacceptably long lag times τ to be used.

Reference 80 has suggested a scheme to strike a balance between reproducing the fast and the slow relaxation behavior of the model for a given definition of coarse sets. Other studies have suggested finding an optimal choice of coarse states[25,75,76] and then to re-estimating the MSM on the coarse state space. It has been first realized in ref 74 that coarse-graining in a way that exactly preserves the slow kinetics, and in particular the m slowest relaxation time scales, is not possible using any hard assignment of microstates to macrostates, but must rather be made by a fuzzy assignment in which each microstate i has a probability of participating in macrostate I:

$$m_{iI} = \mathbb{P}(\text{macro } I|\text{micro } i) \tag{5}$$

and vice versa, each macrostate I has a probability of being observed in microstate i

$$\chi_{Ii} = \mathbb{P}(\text{micro } i|\text{macro } I)$$

We can write these probabilities into a membership matrix $\mathbf{M} \in \mathbb{R}^{n\times m}$ and an observation probability matrix $\chi \in \mathbb{R}^{m\times n}$ and relate them by

$$\chi = \mathbf{\Pi}^{-1}\mathbf{M}^{\mathrm{T}}\tilde{\mathbf{\Pi}} \tag{6}$$

where $\mathbf{\Pi} = \mathrm{diag}(\pi_i)$ and $\tilde{\mathbf{\Pi}} = \mathrm{diag}(\pi_I)$.[78] The key contribution of ref 74 is the finding that when the membership probabilities (5) are computed by a linear transformation of the first m transition matrix eigenvectors, which is achieved by the PCCA+ and PCCA++ methods,[73,77] then there is a $m \times m$-sized coarse-grained transition matrix (here using the formulation of ref 78):

$$\tilde{\mathbf{P}} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\mathbf{P}\mathbf{M} \tag{7}$$

which exactly preserves the relaxation kinetics of the m slowest processes. Unfortunately the transition matrix (7) is physically not very appealing because it can have a few negative transition probabilities. However, it was noticed in ref 78 that (7) and (6) together define a hidden Markov model and that hidden Markov models are a good approximation to the exact dynamics on discrete state spaces for all long time scales. These insights lead to the following rigorous and unique coarse-graining strategy:

(1) Estimate and validate a microstate MSM with transition matrix $\mathbf{P}(\tau)$ at lag time τ.

(2) Compute the initial HMSM using (7 and 6).

(3) Optimize $\tilde{\mathbf{P}}$ and χ by maximizing the HMSM likelihood using the Baum−Welch algorithm[119,120]

(4) Analyze $\tilde{\mathbf{P}}$ as the coarse-grained Markov model. Representative structures can be sampled from the row vectors of χ.

Error bars on the coarse-grained Markov model can be computed as described earlier for HMSMs. The coarse-graining procedure is completely automatized in PyEMMA. A coarse-grained MSM can be simply obtained by calling `coarse_grain(m)` on an estimated MSM object. This analysis results in the coarse-grained representation shown in Figures 1 and 5.

RESULTS

Feature Selection Using TICA. Here we evaluate the questions of how input features should be selected and how the featurized input data should be transformed in order to prepare it for the construction of kinetic models.

In Figure 4, we have used a 1 ms simulation of bovine pancreatic trypsin inhibitor (BPTI) produced by the D. E. Shaw Anton supercomputer[1] (see there for details of the simulation setup). The analysis was done using a 10 ns stride (100000 frames), which allows all analyses shown in Figure 4 to be run within a few minutes on an ordinary laptop computer. Smaller strides were tried but did not change the qualitative picture.

We have compared three feature sets: (i) the 174 coordinates of the 58 Cα atoms, after rototranslational alignment; (ii) the 1653 distances between pairs of Cα coordinates; and (iii) residue−residue contacts defined to be 1 for Cα distances $\leq$0.75 nm and 0 for larger separations. These feature sets were then fed into a PCA and a TICA transformation, respectively, resulting in six setups that are being compared. In order to communicate a visual impression we show the projections of the simulation trajectory onto the first two main components (left column), as well as the logarithm of the histogram as a simple estimate of the free energy surface (second column). Note that in none of these setups is the variance well-explained by two coordinates, and thus panels a and d should not be quantitatively interpreted. For the quantitative analysis (b, c, and e−h) much higher dimensions are employed.

Looking at the low-dimensional projections (a, d) already reveals very interesting qualitative differences. The three PCA analyses all appear very different. There are two deeper minima in each of them, but the overall structure is different when a different feature set is chosen. In stark contrast, the three TICA projections appear very similar despite extremely different input features. The trajectory projections (d, left) appear almost identical to the eye. Subtle differences show up in the free energy plots (d, right), but the overall structure is conserved: two deeper minima separated by the second-slowest process, both of which are separated from a less deep set of minima by the slowest process. The TICA projection clearly separates the three structures shown in Figure 4h that are later found to be the metastable conformations of BPTI. These qualitative observations confirm that TICA should be preferred over PCA if our aim is to build a kinetic model of the slow molecular transitions.

Within a TICA transformation the quality of feature sets can be compared using the total kinetic variance TKV (see earlier). The best feature set is the one that maximizes the TKV as it best retains the slow processes intrinsic in the unfeaturized data. In Figure 4f, it is seen that, for this BPTI data set, the KTV indicates that Cα coordinates and residue−residue contacts perform similarly, while the Cα distances perform much better in terms of resolving the slow processes.

To validate this indication, we build MSMs from each of these six setups and for a range of lag times τ. In order to make a fair comparison, we always selected a number of dimensions such that 95% of the total (kinetic) variance is retained in each setup. The TICA coordinates were scaled to a kinetic map.[54] Discretization was always done with 100 clusters using k-means with k-means++ initialization,[60] which leads to well-reproducible cluster center distributions and thus to robust conclusions.

The relaxation time scales shown in panels b and e are excellent measures of the model quality, as larger time scales indicate a better discretization,[95] and in this case a better choice of input features and transformation method. Within the PCA setup, the Cα distances perform slightly better than the other feature sets. However, all PCA setups are by far outperformed

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

by each TICA setup that recovers much slower time scales. Within TICA setups the Cα distances are again the best feature set.

Figure 4g summarizes the model quality in a single number for each model at lag time τ: The total kinetic variation TKV of the MSM is computed as the sum of squares of its eigenvalues. This evaluation leads to the same conclusions as those earlier: (i) TICA substantially outperforms PCA in terms of kinetic modeling, which is a general statement because TICA is optimal among linear projection methods in terms of approximating the slow reaction coordinates and maximizing the time scales,[36] and (ii) for BPTI, the Cα distances outperform Cartesian coordinates and contacts.

Since conclusion ii is model-dependent, it should be checked for different systems. Moreover, optimality is not the only relevant aspect. For large macromolecules, it is unaffordable to compute all pairwise residue distances. In contrast, variable residue−residue contacts, i.e., contacts of those residue pairs that form or break a contact at least once in the simulation, are an affordable metric even for rather large macromolecular systems.[127]

Application to Protein−Ligand Binding. Here we illustrate the construction and analysis of a kinetic model for protein−ligand binding. A set of 50 × 200 ns simulations (10 μs cumulated simulation time) was run for the serine protease trypsin with its reversible competitive inhibitor benzamidine in explicit solvent. The simulation setup is described in ref 29. All simulations started with trypsin and benzamidine separated, i.e., in the dissociated state. This simulation set is much shorter than data sets used in earlier studies,[29,31] and in addition data were only saved once every 1 ns, leading to larger statistical errors than in previous analyses. However, we have deliberately limited the amount of data as these data are provided for download as part of the PyEMMA tutorials. Despite the smaller amount of data, we shall see that a very interesting kinetic analysis can be made.

Previous analyses were able to characterize the binding kinetics and binding affinity of trypsin−benzamidine.[29,128] In other studies,[31,129,130] it has been found that trypsin−benzamidine binding and dissociation are accompanied by conformational changes in the trypsin binding loops. It has been shown that these trypsin conformation changes need to be taken into account in order to arrive at a complete description of the binding kinetics.[31] However, none of these studies could correctly compute the dissociation rate of trypsin and benzamidine—these rates were always overestimated.[29,31]

Here we demonstrate the capabilities of hidden MSMs by using the 10 μs trypsin−benzamidine data set and selecting the nearest neighbor heavy-atom contacts between benzamidine and all trypsin residues as input features. These features were transformed by TICA to a kinetic map preserving 95% of the kinetic variance, resulting in a 59-dimensional transformed space. As described in Methods, 200 clusters were generated by k-means clustering. Bayesian hidden Markov models using five metastable states were estimated at lag times of 1−10 ns. These lags are short compared to those previously used MSM analyses;[29,31] however, it has been found (see ref 78 and Figure 3) that HMSMs can often be estimated accurately with much smaller lag times than those of MSMs.

Figure 5a shows the implied relaxation time scales as a function of the lag time, which are indeed constant within statistical error for short lag times around 2 ns, while the estimates start to diverge for lag times of 5 ns and larger. We therefore select τ = 2 ns and further analyze this model. Figure 5b shows the binding free energy, computed as

$$\Delta G = -kT \ln\left(\frac{\pi_{\mathrm{bound}}}{\pi_{\mathrm{unbound}}}\frac{V_{\mathrm{unbound}}}{V_0}\right)$$

where π_{bound} is the stationary probability of the bound state, π_{unbound} is the stationary probability of the state including the cluster with zero trypsin−benzamidine contacts, V_{unbound} is the simulated solvent volume, and V_0 = 1.663 nm^3 is the standard volume (see refs 31 and 128). The maximum-likelihood HMSM binding free energy estimate at τ = 1 is slightly higher but within statistical uncertainty of the experimental values given in refs 131 and132.

Parts c and d of Figure 5 show the binding and dissociation rates of trypsin−benzamidine, computed as

$$k_{\mathrm{on}} = \frac{1}{t_{\mathrm{unbound}\to\mathrm{bound}}}\frac{V_{\mathrm{unbound}}}{V_0}$$

$$k_{\mathrm{off}} = \frac{1}{t_{\mathrm{bound}\to\mathrm{unbound}}}$$

where $t_{\mathrm{unbound}\to\mathrm{bound}}$ and $t_{\mathrm{bound}\to\mathrm{unbound}}$ are the mean first passage times computed from the HMSM from the unbound to the bound state and vice versa. For the first time we can match not only the binding rate but also the dissociation rate with the experimental value of ref 133 within statistical error.

Figure 5e shows the rate matrix obtained from the five-state MSM, represented by a PyEMMA network plot. It is seen that the network has a star structure, where the unbound (green) state can communicate with all of the other states. While the yellow state is misbound, i.e., a dead end or trap state, the other states are arranged linearly; i.e., they can transition into one another toward binding or dissociate. The individual transition rates are given in units of (2 ns)$^{-1}$, and are in the range of 1μs to 1/(10 ms).

Figure 5f shows the ensemble of binding pathways analyzed by TPT. The main binding pathways are direct binding from the dissociated state or binding through the blue intermediate. Other pathways, in which the benzamidine goes through low-populated intermediates are possible but much less frequent.

Figure 5g shows the positions of the benzamidine drawn on the simulation starting structure of trypsin. The five different metastable states of the HMSM are shown in different colors, corresponding to the color code in Figure 5e,f. Note that the structures shown here (and all structures shown in the other figures) have been automatically selected by PyEMMA based on the metastable state probabilities in the HMSM model and were not manually chosen. The bound state (purple) is very well defined by the Trp in the binding cavity and forming a salt bridge to Asp 189. The unbound state (green) can be clearly seen as the only state which has most ligands in the solvent, or weakly attached to trypsin. Two metastable misbound states are found (yellow, blue), where benzamidine binds to off-target aspartic acids. Out of these, the yellow state is far from the binding site and forms an off-pathway intermediate, that benzamidine has to dissociate from in order to get on a binding pathway. Another weakly stable intermediate that is not stabilized by specific salt bridge contacts (cyan) is close to the binding pocket. The arrows in Figure 5g qualitatively show the binding fluxes from Figure 5f. Benzamidine can bind from the dissociated state to any intermediate or directly to the bound state. When starting in the cyan state, benzamidine

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

jumps to the blue prebound state and finally binds into the purple state.

CONCLUSIONS

We have described the software package PyEMMA for the construction, validation, and analysis of Markov models, hidden Markov models, and other kinetic models from MD simulation data. The features, the software architecture, and the usage of the software were discussed, and new methodological approaches and results were described.

PyEMMA is a very actively developed code, and we have a long-term interest in keeping the development up. Near-future additions will include packages for the variational approach of conformation dynamics (VAC),[95,96] Markov transition models (MTMs),[93] and kinetic models using data from multiple thermodynamic states, e.g., using the TRAM estimator.[97,98] In addition, various extensions to make kinetic model construction more convenient, automatic, and efficient will be made.

AUTHOR INFORMATION

Corresponding Author

*E-mail: frank.noe@fu-berlin.de.

Author Contributions

†M.K.S. and B.T.-S. contributed equally to this work.

Funding

We acknowledge funding from ERC starting grant pcCell (to M.K.S., G.P.-H., B.T.-S., F.P., and F.N), FU Berlin startup funds (to F.N., F.P.-H., and M.H.), Deutsche Forschungsgemeinschaft Grant Nos. 825/3-1 (to B.T.-S. and G.P.-H.) and SFB 1114 (to F.P.), and Einstein Foundation Berlin, SoOPiC (to N.P.).

Notes

PyEMMA is an open-source code hosted at https://github.com/markovmodel/pyemma, and contributions from the community are very welcome. Bug reports or feature requests are also welcome as they help to improve the software. We encourage users to post issues on our Github page if bugs are found or new feature requests are made. PyEMMA's license is permissive; i.e., the code or parts of it can be used in other software frameworks, including commercial ones, provided that a copyright note is kept with the used code. If improvements to or extensions of the code are made, we ask them to be contributed to the PyEMMA repository such that they become available to the community. In order to avoid duplicate work, please contact the authors before making time-consuming developments. We prefer code contributions to be made in the form of pull requests. Please refer to the developer guide at pyemma.org for details.

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We are grateful to the following scientists and organizations: Gianni De Fabritiis and Stefan Doerr (UPF Barcelona) for setup and parameters for the trypsin-benzamidine system; D. E. Shaw Research for providing access to the 1 ms BPTI simulation trajectory; Robert T. McGibbon (Stanford University), Kyle A. Beauchamp, and John D. Chodera (MSKCC) for setting up the Omnia infrastructure; Susanna Röblitz (ZIB) for her help with PCCA++; Antonia S. J. S. Mey, Christoph Wehmeyer, and Edina Rosta for their help with organizing PyEMMA workshops; Cecilia Clementi (Rice University), Gianni De Fabritiis (UPF Barcelona), Feliks Nüske, Francesca Vitalini, Bettina G. Keller, Antonia S. J. S. Mey, Christoph Wehmeyer, Hao Wu (FU Berlin), and the participants of the previous PyEMMA workshops for inspiring discussions.

REFERENCES

(1) Shaw, D. E.; Maragakis, P.; Lindorff-Larsen, K.; Piana, S.; Dror, R.; Eastwood, M.; Bank, J.; Jumper, J.; Salmon, J.; Shan, Y.; Wriggers, W. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science* **2010**, *330*, 341−346.

(2) Buch, I.; Harvey, M. J.; Giorgino, T.; Anderson, D. P.; De Fabritiis, G. High-throughput all-atom molecular dynamics simulations using distributed computing. *J. Chem. Inf. Model.* **2010**, *50*, 397.

(3) Shirts, M.; Pande, V. S. Screen Savers of the World Unite! *Science* **2000**, *290*, 1903−1904.

(4) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* **2005**, *26*, 1781−1802.

(5) Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; Hess, B.; Lindahl, E. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* **2013**, *29*, 845−854.

(6) Pronk, S.; Pouya, I.; Lundborg, M.; Rotskoff, G.; Wesén, B.; Kasson, P. M.; Lindahl, E. Molecular Simulation Workflows as Parallel Algorithms: The Execution Engine of Copernicus, a Distributed High-Performance Computing Platform. *J. Chem. Theory Comput.* **2015**, *11*, 2600−2608.

(7) Harvey, M.; Giupponi, G.; Fabritiis, G. D. ACEMD: Accelerated molecular dynamics simulations in the microseconds timescale. *J. Chem. Theory Comput.* **2009**, *5*, 1632.

(8) Eastman, P.; et al. OpenMM 4: A Reusable, Extensible, Hardware Independent Library for High Performance Molecular Simulation. *J. Chem. Theory Comput.* **2013**, *9*, 461−469.

(9) Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine microsecond molecular dynamics simulations with AMBER - Part II: Particle Mesh Ewald. *J. Chem. Theory Comput.* **2013**, *9*, 3878−3888.

(10) Schütte, C.; Fischer, A.; Huisinga, W.; Deuflhard, P. A Direct Approach to Conformational Dynamics based on Hybrid Monte Carlo. *J. Comput. Phys.* **1999**, *151*, 146−168.

(11) Swope, W. C.; Pitera, J. W.; Suits, F. Describing protein folding kinetics by molecular dynamics simulations: 1. Theory. *J. Phys. Chem. B* **2004**, *108*, 6571−6581.

(12) Chekmarev, D. S.; Ishida, T.; Levy, R. M. Long-Time Conformational Transitions of Alanine Dipeptide in Aqueous Solution: Continuous and Discrete-State Kinetic Models. *J. Phys. Chem. B* **2004**, *108*, 19487−19495.

(13) Singhal, N.; Pande, V. S. Error analysis and efficient sampling in Markovian state models for molecular dynamics. *J. Chem. Phys.* **2005**, *123*, 204909.

(14) Sriraman, S.; Kevrekidis, I. G.; Hummer, G. Coarse Master Equation from Bayesian Analysis of Replica Molecular Dynamics Simulations. *J. Phys. Chem. B* **2005**, *109*, 6479−6484.

(15) Noé, F.; Horenko, I.; Schütte, C.; Smith, J. C. Hierarchical Analysis of Conformational Dynamics in Biomolecules: Transition Networks of Metastable States. *J. Chem. Phys.* **2007**, *126*, 155102.

(16) Chodera, J. D.; Singhal, N.; Pande, V. S.; Dill, K. A.; Swope, W. C. Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics. *J. Chem. Phys.* **2007**, *126*, 155101.

(17) Noé, F. Probability Distributions of Molecular Observables computed from Markov Models. *J. Chem. Phys.* **2008**, *128*, 244103.

(18) Buchete, N. V.; Hummer, G. Coarse Master Equations for Peptide Folding Dynamics. *J. Phys. Chem. B* **2008**, *112*, 6057−6069.

(19) Pan, A. C.; Roux, B. Building Markov state models along pathways to determine free energies and rates of transitions. *J. Chem. Phys.* **2008**, *129*, 064107.

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(20) Noé, F.; Schütte, C.; Vanden-Eijnden, E.; Reich, L.; Weikl, T. R. Constructing the Full Ensemble of Folding Pathways from Short Off-Equilibrium Simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 19011−19016.
(21) Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. Progress and challenges in the automated construction of Markov state models for full protein systems. *J. Chem. Phys.* **2009**, *131*, 124101.
(22) Voelz, V. A.; Bowman, G. R.; Beauchamp, K. A.; Pande, V. S. Molecular Simulation of ab Initio Protein Folding for a Millisecond Folder NTL9. *J. Am. Chem. Soc.* **2010**, *132*, 1526−1528.
(23) Bowman, G. R.; Pande, V. S. Protein folded states are kinetic hubs. *Proc. Natl. Acad. Sci. U. S. A.* **2010**, *107*, 10890−10895.
(24) Voelz, V. A.; Jäger, M.; Zhu, L.; Yao, S.; Bakajin, O.; Weiss, S.; Lapidus, L. J.; Pande, V. S. Markov State Models of Millisecond Folder ACBP Reveals New Views of the Folding Reaction. *Biophys. J.* **2011**, *100*, 515a.
(25) Beauchamp, K. A.; McGibbon, R.; Lin, Y. S.; Pande, V. S. Simple few-state models reveal hidden complexity in protein folding. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 17807−17813.
(26) Sborgi, L.; Verma, A.; Piana, S.; Lindorff-Larsen, K.; Cerminara, M.; Santiveri, C. M.; Shaw, D. E.; de Alba, E.; Muñoz, V. Interaction Networks in Protein Folding via Atomic-Resolution Experiments and Long-Time-Scale Molecular Dynamics Simulations. *J. Am. Chem. Soc.* **2015**, *137*, 6506−6516.
(27) Held, M.; Metzner, P.; Prinz, J.-H.; Noé, F. Mechanisms of Protein-Ligand Association and Its Modulation by Protein Mutations. *Biophys. J.* **2011**, *100*, 701−710.
(28) Silva, D.-A.; Bowman, G. R.; Sosa-Peinado, A.; Huang, X. A Role for Both Conformational Selection and Induced Fit in Ligand Binding by the LAO Protein. *PLoS Comput. Biol.* **2011**, *7*, e1002054.
(29) Buch, I.; Giorgino, T.; De Fabritiis, G. Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 10184−10189.
(30) Gu, S.; Silva, D.-A.; Meng, L.; Yue, A.; Huang, X. Quantitatively Characterizing the Ligand Binding Mechanisms of Choline Binding Protein Using Markov State Model Analysis. *PLoS Comput. Biol.* **2014**, *10*, e1003767.
(31) Plattner, N.; Noé, F. Protein conformational plasticity and complex ligand binding kinetics explored by atomistic simulations and Markov models. *Nat. Commun.* **2015**, *6*, 7653.
(32) Jiang, H.; Sheong, F. K.; Zhu, L.; Gao, X.; Bernauer, J.; Huang, X. Markov State Models Reveal a Two-Step Mechanism of miRNA Loading into the Human Argonaute Protein: Selective Binding followed by Structural Re-arrangement. *PLoS Comput. Biol.* **2015**, *11*, e1004404.
(33) de Groot, B.; Daura, X.; Mark, A.; Grubmüller, H. Essential Dynamics of Reversible Peptide Folding: Memory-Free Conformational Dynamics Governed by Internal Hydrogen Bonds. *J. Mol. Biol.* **2001**, *309*, 299−313.
(34) Singhal, N.; Snow, C.; Pande, V. S. Path sampling to build better roadmaps: predicting the folding rate and mechanism of a Trp Zipper beta hairpin. *J. Chem. Phys.* **2004**, *121*, 415−425.
(35) Chodera, J. D.; Swope, W. C.; Pitera, J. W.; Dill, K. A. Long-time protein folding dynamics from short-time molecular dynamics simulations. *Multiscale Model. Simul.* **2006**, *5*, 1214−1226.
(36) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, 015102.
(37) Stanley, N.; Esteban-Martin, S.; De Fabritiis, G. Kinetic modulation of a disordered protein domain by phosphorylation. *Nat. Commun.* **2014**, *5*, 5272.
(38) Schor, M.; Mey, A. S. J. S.; Noé, F.; MacPhee, C. Shedding Light on the Dock-Lock Mechanism in Amyloid Fibril Growth Using Markov State Models. *J. Phys. Chem. Lett.* **2015**, *6*, 1076−1081.
(39) Morcos, F.; Chatterjee, S.; McClendon, C. L.; Brenner, P. R.; López-Rendón, R.; Zintsmaster, J.; Ercsey-Ravasz, M.; Sweet, C. R.; Jacobson, M. P.; Peng, W. J.; Izaguirre, J. A. Modeling Conformational Ensembles of Slow Functional Motions in Pin1-WW. *PLoS Comput. Biol.* **2010**, *6*, e1001015.
(40) Faelber, K.; Posor, Y.; Gao, S.; Held, M.; Roske, Y.; Schulze, D.; Haucke, V.; Noé, F.; Daumke, O. Crystal structure of nucleotide-free dynamin. *Nature* **2011**, *477*, 556−560.
(41) Sadiq, S. K.; Noé, F.; De Fabritiis, G. Kinetic characterization of the critical step in HIV-1 protease maturation. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 20449−20454.
(42) Xia, J.; Deng, N.; Levy, R. M. NMR Relaxation in Proteins with Fast Internal Motions and Slow Conformational Exchange: Model-Free Framework and Markov State Simulations. *J. Phys. Chem. B* **2013**, *117*, 6625−6634.
(43) Kohlhoff, K. J.; Shukla, D.; Lawrenz, M.; Bowman, G. R.; Konerding, D. E.; Belov, D.; Altman, R. B.; Pande, V. S. Cloud-based simulations on Google Exacycle reveal ligand modulation of GPCR activation pathways. *Nat. Chem.* **2014**, *6*, 15−21.
(44) Shukla, D.; Meng, Y.; Roux, B.; Pande, V. S. Activation pathway of Src kinase reveals intermediate states as targets for drug design. *Nat. Commun.* **2014**, *5*, 3397.
(45) Pontiggia, F.; Pachov, D. V.; Clarkson, M. W.; Villali, J.; Hagan, M. F.; Pande, V. S.; Kern, D. Free energy landscape of activation in a signalling protein at atomic resolution. *Nat. Commun.* **2015**, *6*, 7284.
(46) Banerjee, R.; Yan, H.; Cukier, R. I. Conformational Transition in Signal Transduction: Metastable States and Transition Pathways in the Activation of a Signaling Protein. *J. Phys. Chem. B* **2015**, *119*, 6591−6602.
(47) Malmstrom, R. D.; Kornev, A. P.; Taylor, S. S.; Amaro, R. E. Allostery through the computational microscope: cAMP activation of a canonical signalling domain. *Nat. Commun.* **2015**, *6*, 7588.
(48) Reubold, T. F.; Faelber, K.; Plattner, N.; Posor, Y.; Ketel, K.; Curth, U.; Schlegel, J.; Anand, R.; Manstein, D. J.; Noé, F.; Haucke, V.; Daumke, O.; Eschenburg, S. Crystal structure of the dynamin tetramer. *Nature* **2015**, *525*, 404−408.
(49) Perkett, M. R.; Hagan, M. F. Using Markov state models to study self-assembly. *J. Chem. Phys.* **2014**, *140*, 214101.
(50) Peters, B.; Trout, B. L. Obtaining reaction coordinates by likelihood maximization. *J. Chem. Phys.* **2006**, *125*, 054108.
(51) Rohrdanz, M. A.; Zheng, W.; Maggioni, M.; Clementi, C. Determination of reaction coordinates via locally scaled diffusion map. *J. Chem. Phys.* **2011**, *134*, 124116.
(52) Schwantes, C. R.; Pande, V. S. Improvements in Markov State Model Construction Reveal Many Non-Native Interactions in the Folding of NTL9. *J. Chem. Theory Comput.* **2013**, *9*, 2000−2009.
(53) Schwantes, C. R.; Pande, V. S. Modeling Molecular Kinetics with tICA and the Kernel Trick. *J. Chem. Theory Comput.* **2015**, *11*, 600−608.
(54) Noé, F.; Clementi, C. Kinetic distance and kinetic maps from molecular dynamics simulation. *J. Chem. Theory Comput.* **2015**, DOI: 10.1021/acs.jctc.5b00553.
(55) Altis, A.; Nguyen, P. H.; Hegger, R.; Stock, G. Dihedral angle principal component analysis of molecular dynamics simulations. *J. Chem. Phys.* **2007**, *126*, 244111.
(56) Nadler, B.; Lafon, S.; Coifman, R. R.; Kevrekidis, I. G. Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. *Adv. Neural Inf. Process Syst.* **2005**, *18*, 955−962.
(57) Yao, Y.; Sun, J.; Huang, X.; Bowman, G. R.; Singh, G.; Lesnick, M.; Guibas, L. J.; Pande, V. S.; Carlsson, G. Topological methods for exploring low-density states in biomolecular folding pathways. *J. Chem. Phys.* **2009**, *130*, 144115.
(58) Keller, B. G.; Daura, X.; van Gunsteren, W. F. Comparing geometric and kinetic cluster algorithms for molecular simulation data. *J. Chem. Phys.* **2010**, *132*, 074110.
(59) Dasgupta, S.; Long, P. Performance guarantees for hierarchical clustering. *J. Comput. Syst. Sci.* **2005**, *70*, 555−569.
(60) Arthur, D.; Vassilvitskii, S. k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*; Society for Industrial and Applied Mathematics: Philadelphia, PA,USA, 2007; pp 1027−1035.
(61) Beauchamp, K. A.; Bowman, G. R.; Lane, T. J.; Maibaum, L.; Haque, I. S.; Pande, V. S. MSMBuilder2: Modeling Conformational

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

Dynamics at the Picosecond to Millisecond Scale. *J. Chem. Theory Comput.* **2011**, *7*, 3412−3419.
(62) Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B. G.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. Markov models of molecular kinetics: Generation and Validation. *J. Chem. Phys.* **2011**, *134*, 174105.
(63) Hinrichs, N. S.; Pande, V. S. Calculation of the distribution of eigenvalues and eigenvectors in Markovian state models for molecular dynamics. *J. Chem. Phys.* **2007**, *126*, 244101.
(64) Chodera, J. D.; Noé, F. Probability distributions of molecular observables computed from Markov models II: Uncertainties in observables and their time-evolution. *J. Chem. Phys.* **2010**, *133*, 105102.
(65) Bacallado, S.; Chodera, J. D.; Pande, V. S. Bayesian comparison of Markov models of molecular dynamics with detailed balance constraint. *J. Chem. Phys.* **2009**, *131*, 045106.
(66) Metzner, P.; Noé, F.; Schütte, C. Estimating the sampling error: Distribution of transition matrices and functions of transition matrices for given trajectory data. *Phys. Rev. E: Stat., Nonlinear, Soft Matter Phys.* **2009**, *80*, 021106.
(67) Schütte, C.; Noé, F.; Lu, J.; Sarich, M.; Vanden-Eijnden, E. Markov State Models Based on Milestoning. *J. Chem. Phys.* **2011**, *134*, 204105.
(68) Trendelkamp-Schroer, B.; Noé, F. Efficient Bayesian estimation of Markov model transition matrices with given stationary distribution. *J. Chem. Phys.* **2013**, *138*, 164113.
(69) Trendelkamp-Schroer, B.; Wu, H.; Paul, F.; Noé, F. Estimation and uncertainty of reversible Markov models. *J. Chem. Phys.* **2015**, manuscript in revision (Preprint at http://arxiv.org/abs/1507.05990).
(70) Sarich, M.; Noé, F.; Schütte, C. On the approximation quality of Markov state models. *Multiscale Model. Simul.* **2010**, *8*, 1154−1177.
(71) Djurdjevac, N.; Sarich, M.; Schütte, C. Estimating the eigenvalue error of Markov State Models. *Multiscale Model. Simul.* **2012**, *10*, 61−81.
(72) McGibbon, R. T.; Pande, V. S. Variational cross-validation of slow dynamical modes in molecular kinetics. *J. Chem. Phys.* **2015**, *142*, 124105.
(73) Deuflhard, P.; Weber, M.; Dellnitz, M.; Kirkland, S.; Neumann, M.; Schütte, C. Robust Perron Cluster Analysis in Conformation Dynamics. *Linear Algebra Its Appl.* **2005**, *398*, 161−184 (Dellnitz, , M., Kirkland, , S., Newmann, , M., Schutte, , C., Eds.).
(74) Kube, S.; Weber, M. A coarse graining method for the identification of transition rates between molecular conformations. *J. Chem. Phys.* **2007**, *126*, 024103.
(75) Bowman, G. R. Improved coarse-graining of Markov state models via explicit consideration of statistical uncertainty. *J. Chem. Phys.* **2012**, *137*, 134111.
(76) Yao, Y.; Cui, R. Z.; Bowman, G. R.; Silva, D.-A.; Sun, J.; Huang, X. Hierarchical Nyström methods for constructing Markov state models for conformational dynamics. *J. Chem. Phys.* **2013**, *138*, 174106.
(77) Röblitz, S.; Weber, M. Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification. *Adv. Data Anal. Classif.* **2013**, *7*, 147−179.
(78) Noé, F.; Wu, H.; Prinz, J.-H.; Plattner, N. Projected and Hidden Markov Models for calculating kinetics and metastable states of complex molecules. *J. Chem. Phys.* **2013**, *139*, 184114.
(79) Sheong, F. K.; Silva, D.-A.; Meng, L.; Zhao, Y.; Huang, X. Automatic State Partitioning for Multibody Systems (APM): An Efficient Algorithm for Constructing Markov State Models To Elucidate Conformational Dynamics of Multibody Systems. *J. Chem. Theory Comput.* **2015**, *11*, 17−27.
(80) Hummer, G.; Szabo, A. Optimal Dimensionality Reduction of Multistate Kinetic and Markov-State Models. *J. Phys. Chem. B* **2015**, *119*, 9029−9037.
(81) E, W.; Vanden-Eijnden, E. Towards a Theory of Transition Paths. *J. Stat. Phys.* **2006**, *123*, 503−523.
(82) Jain, A.; Stock, G. Hierarchical Folding Free Energy Landscape of HP35 Revealed by Most Probable Path Clustering. *J. Phys. Chem. B* **2014**, 7750−7760.
(83) Metzner, P.; Schütte, C.; Vanden-Eijnden, E. Transition Path Theory for Markov Jump Processes. *Multiscale Model. Simul.* **2009**, *7*, 1192−1219.
(84) Berezhkovskii, A.; Hummer, G.; Szabo, A. Reactive flux and folding pathways in network models of coarse-grained protein dynamics. *J. Chem. Phys.* **2009**, *130*, 205102.
(85) Noé, F.; Doose, S.; Daidone, I.; Löllmann, M.; Chodera, J. D.; Sauer, M.; Smith, J. C. Dynamical fingerprints for probing individual relaxation processes in biomolecular dynamics with simulations and kinetic experiments. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 4822−4827.
(86) Zhuang, W.; Cui, R. Z.; Silva, D.-A.; Huang, X. Simulating the T-Jump-Triggered Unfolding Dynamics of trpzip2 Peptide and Its Time-Resolved IR and Two-Dimensional IR Signals Using the Markov State Model Approach. *J. Phys. Chem. B* **2011**, *115*, 5415−5424.
(87) Bowman, G. R.; Voelz, V. A.; Pande, V. S. Atomistic Folding Simulations of the Five-Helix Bundle Protein Lambda 6−85. *J. Am. Chem. Soc.* **2011**, *133*, 664−667.
(88) Keller, B. G.; Prinz, J.-H.; Noé, F. Markov models and dynamical fingerprints: Unraveling the complexity of molecular kinetics. *Chem. Phys.* **2012**, *396*, 92−107.
(89) Lindner, B.; Yi, Z.; Prinz, J.-H.; Smith, J. C.; Noé, F. Dynamic Neutron Scattering from Conformational Dynamics I: Theory and Markov models. *J. Chem. Phys.* **2013**, *139*, 175101.
(90) Yi, Z.; Lindner, B.; Prinz, J.-H.; Noé, F.; Smith, J. C. Dynamic Neutron Scattering from Conformational Dynamics II: Application Using Molecular Dynamics Simulation and Markov Modeling. *J. Chem. Phys.* **2013**, *139*, 175102.
(91) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B.; Marqusee, S. Discovery of multiple hidden allosteric sites by combining Markov state models and experiments. *Proc. Natl. Acad. Sci. U. S. A.* **2015**, *112*, 2734−2739.
(92) Bowman, G. R., Pande, V. S., Noé, F., Eds. An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation *Advances in Experimental Medicine and Biology*; Springer: Heidelberg, Germany, 2014; Vol. *797*.
(93) Wu, H.; Noé, F. Gaussian Markov transition models of molecular kinetics. *J. Chem. Phys.* **2015**, *142*, 084104.
(94) McGibbon, R. T.; Ramsundar, B.; Sultan, M. M.; Kiss, G.; Pande, V. S. Understanding Protein Dynamics with L1-Regularized Reversible Hidden Markov Models. *Proc. Int. Conf. Mach. Learn.* **2014**, 1197−1205.
(95) Noé, F.; Nüske, F. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Model. Simul.* **2013**, *11*, 635−655.
(96) Nüske, F.; Keller, B. G.; Pérez-Hernández, G.; Mey, A. S. J. S.; Noé, F. Variational Approach to Molecular Kinetics. *J. Chem. Theory Comput.* **2014**, *10*, 1739−1752.
(97) Wu, H.; Noé, F. Optimal estimation of free energies and stationary densities from multiple biased simulations. *Multiscale Model. Simul.* **2014**, *12*, 25−54.
(98) Mey, A. S. J. S.; Wu, H.; Noé, F. xTRAM: Estimating equilibrium expectations from time-correlated simulation data at multiple thermodynamic states. *Phys. Rev. X* **2014**, *4*, 041018.
(99) Rosta, E.; Hummer, G. Free energies from dynamic weighted histogram analysis using unbiased Markov state model. *J. Chem. Theory Comput.* **2015**, *11*, 276−285.
(100) Wu, H.; Mey, A. S. J. S.; Rosta, E.; Noé, F. Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. *J. Chem. Phys.* **2014**, *141*, 214106.
(101) Senne, M.; Trendelkamp-Schroer, B.; Mey, A. S. J. S.; Schütte, C.; Noé, F. EMMA - A software package for Markov model building and analysis. *J. Chem. Theory Comput.* **2012**, *8*, 2223−2238.
(102) *GNU Operating System*, http://www.gnu.org/licenses/gpl-3.0.html.
(103) Molgedey, L.; Schuster, H. G. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.* **1994**, *72*, 3634−3637.

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX

(104) Ferrenberg, A. M.; Swendsen, R. H. Optimized Monte Carlo data analysis. *Phys. Rev. Lett.* **1989**, *63*, 1195−1198.
(105) Vitalini, F.; Noé, F.; Keller, B. G. A basis set for peptides for the variational approach to conformational kinetics. *J. Chem. Theory Comput.* **2015**, *11*, 3992−4004.
(106) Nüske, F.; Schneider, R.; Vitalini, F.; Noé, F. Variational Tensor Approach for Approximating the Rare-Event Kinetics of Macromolecular Systems. *J. Chem. Phys.* **2015**, manuscript in revision.
(107) McGibbon, R. T.; Beauchamp, K. A.; Schwantes, C. R.; Wang, L.-P.; Hernández, C. X.; Harrigan, M. P.; Lane, T. J.; Swails, J. M.; Pande, V. S. MDTraj: a modern, open library for the analysis of molecular dynamics trajectories. BioRxiv, 2014, DOI: 10.1101/008896.
(108) ten Wolde, P. R.; Chandler, D. Drying-induced hydrophobic polymer collapse. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99*, 6539−6543.
(109) Prada-Gracia, D.; Shevchuk, R.; Hamm, P.; Rao, F. Towards a microscopic description of the free-energy landscape of water. *J. Chem. Phys.* **2012**, *137*, 144504.
(110) Harrigan, M. P.; Shukla, D.; Pande, V. S. Conserve Water: A Method for the Analysis of Solvent in Molecular Dynamics. *J. Chem. Theory Comput.* **2015**, *11*, 1094−1101.
(111) Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philos. Mag.* **1901**, *2*, 559−572.
(112) Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Edu. Psych* **1933**, *24*, 417−441.
(113) Voronoi, M. G. Nouvelles applications des parametres continus a la theorie des formes quadratiques. *J. Reine Angew. Math.* **1908**, *134*, 198−287.
(114) Lloyd, S. P. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129−137.
(115) Sculley, D. Web-scale k-means clustering. *Proceedings of the 19th international conference on World wide web*, Raleigh, NC, USA, Apr. 26−30, 2010; ACM: New York, NY, USA, 2010; pp 1177−1178, DOI: 10.1145/1772690.1772862.
(116) Noé, F. Statistical inefficiency of Markov model count matrices. Preprint, http://publications.mi.fu-berlin.de/1699/ 2015.
(117) Trendelkamp-Schroer, B.; Noé, F. Efficient estimation of rare-event kinetics. *Phys. Rev. X* **2015**, manuscript in revision (preprint at arXiv:1409.6439).
(118) Prinz, J.-H.; Chodera, J. D.; Noé, F. Spectral rate theory for two-state kinetics. *Phys. Rev. X* **2014**, *4*, 011020.
(119) Baum, L. E.; 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*, 164−171.
(120) Welch, L. R. Hidden Markov Models and the Baum-Welch Algorithm. *IEEE Inf. Theory Soc. Newsl.* **2003**, *53*, 1−13.
(121) Rabiner, L. R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257−286.
(122) Chodera, J. D.; Elms, P.; Noé, F.; Keller, B. G.; Kaiser, C. M.; Ewall-Wice, A.; Marqusee, S.; Bustamante, C.; Hinrichs, N. S. Bayesian hidden Markov model analysis of single-molecule force spectroscopy: Characterizing kinetics under measurement uncertainty, 2011; http://arxiv.org/abs/1108.1430.
(123) Pohlhausen, E. Berechnung der Eigenschwingungen statisch-bestimmter Fachwerke. *Z. Angew. Math. Mech.* **1921**, *1*, 28−42.
(124) Du, R.; Pande, V. S.; Grosberg, A. Y.; Tanaka, T.; Shakhnovich, E. S. On the transition coordinate for protein folding. *J. Chem. Phys.* **1998**, *108*, 334−350.
(125) Bolhuis, P. G.; Dellago, C.; Chandler, D. Reaction coordinates of biomolecular isomerization. *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 5877−5882.
(126) Pirchi, M.; Ziv, G.; Riven, I.; Cohen, S. S.; Zohar, N.; Barak, Y.; Haran, G. Single-molecule fluorescence spectroscopy maps the folding landscape of a large protein. *Nat. Commun.* **2011**, *2*, 493.
(127) Blau, C.; Grubmüller, H. g_contacts: Fast contact search in bio-molecular ensemble data. *Comput. Phys. Commun.* **2013**, *184*, 2856−2859.
(128) Doerr, S.; De Fabritiis, G. D. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *J. Chem. Theory Comput.* **2014**, *10*, 2064−2069.
(129) Limongelli, V.; Bonomi, M.; Parrinello, M. Funnel metadynamics as accurate binding free-energy method. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110*, 6358−6363.
(130) Tiwary, P.; Limongelli, V.; Salvalaglio, M.; Parrinello, M. Kinetics of protein-ligand unbinding: Predicting pathways, rates, and rate-limiting steps. *Proc. Natl. Acad. Sci. U. S. A.* **2015**, *112*, E386−E391.
(131) Katz, B. A.; et al. A novel serine protease inhibition motif involving a multi-centered short hydrogen bonding network at the active site. *J. Mol. Biol.* **2001**, *307*, 1451−1486.
(132) Talhout, R.; Engberts, J. B. F. N. Thermodynamic analysis of binding of p-substituted benzamidines to trypsin. *Eur. J. Biochem.* **2001**, *268*, 1554−1560.
(133) Guillain, F.; Thusius, D. Use of proflavine as an indicator in temperature-jump studies of the binding of a competitive inhibitor to trypsin. *J. Am. Chem. Soc.* **1970**, *92*, 5534−5536.
(134) Humphrey, W.; Dalke, A.; Schulten, K. VMD: visual molecular dynamics. *J. Mol. Graphics* **1996**, *14*, 33−38.

NOTE ADDED AFTER ASAP PUBLICATION

This paper was published on the Web on October 14, 2015. An additional reference was added to the paper, and the paper was reposted on October 16, 2015.

DOI: 10.1021/acs.jctc.5b00743
J. Chem. Theory Comput. XXXX, XXX, XXX−XXX