Markov state models: to optimize or not to optimize

Robert E. Arbon,[†,‡] Yanchen Zhu,[†] and Antonia S.J.S. Mey[*,†]

†*EaStCHEM School of Chemistry, David Brewster Road, Joseph Black Building, The King's Buildings, Edinburgh, EH93FJ, UK*

‡*ReDesign Science, 180 Varick St, New York, NY 10014, USA*

E-mail: antonia.mey@ed.ac.uk

Abstract

Markov state models (MSM) are a popular statistical method for analyzing the conformational dynamics of proteins, including protein folding. With all statistical and machine learning (ML) models choices must be made about the modeling pipeline that cannot be directly learned from the data. These choices, or hyperparameters, are often evaluated by expert judgment or, in the case of MSMs, by maximizing variational scores such as the VAMP-2 score. Modern ML and statistical pipelines often use automatic hyperparameter selection techniques ranging from the simple: choosing the best score from a random selection of hyperparameters to the complex: optimization via e.g., Bayesian optimization. In this work, we ask whether it is possible to automatically select MSM models this way by estimating and analysing over 16 000 000 observations from over 280 000 estimated MSMs. We find that differences in hyperparameters can change the physical interpretation of the optimization objective making automatic selection difficult. In addition, we find that enforcing conditions of equilibrium in the VAMP scores can result in inconsistent model selection. However, other parameters which specify the VAMP-2 score (lag time and number of relaxation processes scored)

have only negligible influence on model selection. We suggest that model observables and variational scores should only be a guide to model selection and that a full investigation of the MSM properties be undertaken when selecting hyperparameters.

Introduction

Markov state models (MSMs) are a popular model for extracting kinetic information from unbiased molecular dynamics simulations. Recent studies include a wide range of applications, such as understanding protein association kinetics,[1,2] enzyme dynamics,[3] ion binding mechanisms,[4,5] hydrogen bond dynamics,[6] drug binding mechanisms for drug discovery,[7–11] mutational effects on conformational dynamics,[12–15] kinetics of intrinsically disordered proteins,[16] protein folding,[17] and understanding allostery.[18–20] Estimating an MSM proceeds[21] by first collecting a data set of unbiased molecular dynamics (MD) simulations, then associating each molecular conformation with discrete states, counting transitions between states separated by the temporal resolution of the model (the lag time, τ), and then deriving transition probabilities between states.[22] The final model is summarized by the transition matrix $\mathbf{T}$, where the elements T_{ij} are the conditional probabilities of being in state i at time t and then transitioning to a state j at time $t{+}\tau$: $T_{ij}(\tau){=}P(j, t{=}t{+}\tau|i, t{=}t)$. The eigenvectors of the transition matrix represent the dynamic modes of the system as they relax to the equilibrium distribution.

The entire process of transforming MD frames into a transition matrix involves making a number of modeling choices called *hyperparameters*. Hyperparameters are differentiated from the *parameters* of the model because the latter are calculated from the data via the optimization of a loss function (e.g., the negative log-likelihood), while the hyperparameters are chosen via expert judgment, or via some summary metric of the model.[23] For MSMs, the important hyperparameters[24–26] are which subset of atoms from the simulation to include (e.g., a protein loop, pocket, or other substructure of interest); the transformation of these coordinates into important features (e.g., residue-residue distances, backbone dihedral

angles); dimensionality reduction onto a set of important collective variables (typically time-lagged independent component analysis, TICA,[27] is used for this purpose); and finally how to define discrete states from these collective variables (via some clustering algorithm such as K-Means). Therefore, the parameters of an MSM are the conditional probabilities in the transition matrix, T_{ij}, whereas the hyperparameters are all the choices (choice of features, clustering algorithm, etc.) that gave us the specific state definitions used in the likelihood maximization step.

Hyperparameter optimization is an important part of modern statistical and machine learning (ML) analysis pipelines[23,28–30] as hyperparameters can have a strong impact of the performance of a model. There are several methods to find the optimal set of hyper-parameters, from exhaustively searching a uniformly spaced grid of choices[31] or randomly selected from a predefined search space,[28] evolutionary and population algorithms[32–35] to active learning approaches such as Bayesian optimization.[30,36–38]

No 'ground truth' data exist for MSMs used for the analysis of protein MD trajectories, so the accuracy of the eigenvectors cannot be judged absolutely. However, a family of variational scores exist which provide a means to compare the relative accuracy of MSMs and thus allow hyperparameter optimization to be performed. The first score to be developed was the cross-validated generalized matrix Rayleigh quotient,[39] GRMQ, which pertains to reversible MSMs; while the variational approach to Markov processes (VAMP) scores[25,40] extended these ideas to both reversible, non-reversible and non-stationary models. These scores measure how well the eigenvectors of the transition matrix (singular vectors in the case of non-reversible models) approximate the 'true' eigenvectors in a variational sense i.e., the higher the score, the better the approximation. Thus, optimization of eigenvectors can proceed without the need for a 'ground truth' to compare to.

To use a variational score, it is necessary to specify the lag time (τ) of the MSM and the number of slow relaxation modes to optimize (k), and then estimate the MSM with different hyperparameters. The 'best' set of hyperparameters is the one with the highest variational

score. In the case of the VAMP-E score, one may also add k to the list of hyperparameters to optimize.

This procedure removes the need for potentially arbitrary hyperparameter selection with the concomitant risk of findings that are not robust to changes in modeling assumptions. This method has been used in a number of different studies.[24,41–49] In addition, it has allowed investigations into the roles of various hyperparameters and for other methods for hyperparameter selection to be developed. In Husic et al.[49] the authors used the GMRQ to show that the Ward and K-means methods are optimal for clustering conformations for MSMs. In Husic et al.[24] the authors performed a sensitivity analysis of the GMRQ in order to determine the sensitivity of hyperparameters in describing protein folding. An extension of the VAMP score by Scherer et al.[25] showed that the optimal set of features could be selected before going through the full MSM creation and scoring pipeline.

It is tempting to think that with a single model metric and state-of-the-art ML optimization software, it should be possible to form an automatic pipeline wherein simulation data are fed in, and a single optimized MSM describing the kinetics and thermodynamics of the system comes out. However, many detailed questions need to be answered before such a pipeline is possible. First, do variational scores refer to the same relaxation mode across all possible combinations of hyperparameters? It is possible that with certain combinations of hyperparameters, the eigenvectors could describe different relaxation modes. It is therefore possible that the variational scores do not compare the same set of processes across different sets of hyperparameters. Second, the MSM lag time and number of scored processes interest will affect the variational scores — does this have any material effect on how we rank different sets of hyperparameters? Third, do we need variational scores to optimize models at all? Will model observables, such as the implied timescales suffice to optimize MSMs? Finally, does hyperparameter optimization work for MSMs compared to randomly sampling hyperparameters? Here, we use a common method (Bayesian optimization with tree Parzen estimators) for optimizing machine learning models to find optimal hyperparameters.

The remainder of this work is structured as follows. In the next section, we cover the necessary theory to understand MSMs and Bayesian optimization of hyperparameters; this is followed by the methods; and results analysis. The paper is concluded with some recommendations based on our findings.

Theory

Markov state models

Overview

What follows is a brief overview of the theory of Markov state models (MSMs), for a more detailed picture see some of the many good references.[22,26,50] MSMs describe the first-order conformational kinetics of a system by specifying the conditional probability of transitioning from a state i at a time t to a state j at a time $t+\tau$ later. This information is summarized in the transition matrix $T_{ij}(\tau) = P(j, t+\tau|i, t)$. Each state, i, is a collection of conformations which have similar kinetic properties. The transition matrix is a finite and discrete representation of the underlying Markovian transfer operator, $\mathcal{T}(\tau)$, which describes the dynamics of the system. The first left eigenvector ϕ_1 (in descending eigenvalue order λ_i, with $\lambda_1 = 1$) corresponds to the stationary or equilibrium distribution, which we also label π; the second left eigenvector, ϕ_2 corresponds to the slowest conformational relaxation process; the third is the next slowest relaxation process and so on. The corresponding right eigenvectors, ψ_i are normalized by π (so $\psi_2 = 1$ for all states). The eigenvalues are related to the timescales of these relaxation processes by: $t_i = -\tau/\log\lambda_i$. The transition matrix is said to be reversible if it obeys detailed balance $\pi_i T_{ij} = \pi_j T_{ji}$.

The transition matrix is specified with respect to a set of p basis states, $\chi_1, \chi_2, ..., \chi_p$ which we denote as a vector $\boldsymbol{\chi}$. In what follows, the basis states are assumed to be discrete and orthonormal and each one corresponds to a small region of conformational space. Each

frame of an MD trajectory can be mapped to one of these basis states and these discretized MD trajectories form the data from which the transition matrix is estimated.

The mapping between the atomic coordinates $\mathbf{x}$ and the basis states we call $f(\mathbf{x};\boldsymbol{\theta}) = \boldsymbol{\chi}$ where $\boldsymbol{\theta}$ is a vector of parameters of that mapping. For example, f may involve projecting coordinates onto the backbone dihedral angles of a protein, followed by clustering into 100 discrete states using k-means clustering. The MSM is then specified with a lag time of 10 ns. The parameters of the MSM are the $100 \times 100 = 10000$ elements of $\mathbf{T}$, while the hyperparameters are $\boldsymbol{\theta}$ = (backbone dihedrals, K-Means, 100) where the elements correspond to the feature, clustering method, number of basis states respectively.

Estimating a reversible MSM

The first step in estimating a reversible MSM is projecting the MD trajectories onto the proposed basis states, $\boldsymbol{\chi}$. Transitions between each basis state at time t and time $t + \tau$ are tabulated in a count matrix, $\mathbf{C}_{0t}$ (the subscript 0 and t refer to the fact that the counted transitions are between t and $t + \tau$). The population of each state is given by the diagonal matrix, $\mathbf{C}_{00}$ calculated as the row-sum of the count matrix $[\mathbf{C}_{00}]_{i,i} = \sum_j [\mathbf{C}_{0t}]_{i,j}$. A *non-reversible* transition matrix is then given by $\mathbf{T}^{\text{irrev}} = \mathbf{C}_{0t}\mathbf{C}_{00}^{-1}$. It is non-reversible because of the finite amount of simulation data will not be in perfect equilibrium. A transition matrix and stationary vector which obey detailed balance, $\mathbf{T}^{\text{rev}}$ and $\boldsymbol{\pi}^{\text{rev}}$, can be estimated from $\mathbf{C}_{0t}$ using maximum likelihood estimation with constraints.[22] The constraints ensure that detailed balance is obeyed by $\mathbf{T}$ and its dynamics are reversible. However, once $\mathbf{T}^{\text{rev}}$ and $\boldsymbol{\pi}^{\text{rev}}$ have been estimated, they are now inconsistent with $\mathbf{C}_{0t}$ and $\mathbf{C}_{00}$, as obtained from the MD trajectory.

Variational scores

The key idea behind variational scores is that approximations to the true eigenvectors of the transition matrix will give rise to eigenvalues which are bounded from above by the true

eigenvalues, specifically:[39,40]

$$\sum_{i=1}^{k} \hat{\lambda}_i^r \leq \sum_{i=1}^{k} \lambda_i^r, \tag{1}$$

where $\hat{\lambda}$ are the eigenvalues estimated from an approximate basis set $\boldsymbol{\chi}$ and λ are the true eigenvalues. The sum runs over the first k eigenvalues, which are typically the dominant slow relaxation processes that one is interested in approximating; while r is some arbitrary positive integer.[40]

When $r = 1$ and the model is assumed to be stationary,[39] the left-hand side of Equation 1 is known as the Generalized Matrix Rayleigh Quotient (GMRQ):

$$\text{GMRQ}(\boldsymbol{\theta}) = \text{Tr}\left[(\mathbf{U}^T\mathbf{C}_{01}\mathbf{U})(\mathbf{U}^T\mathbf{C}_{00}^{\mathbf{U}})^{-1}\right], \tag{2}$$

where $\mathbf{U}$ is the matrix of eigenvectors of $\mathbf{T}$. The functional dependence of the GMRQ on $\boldsymbol{\theta}$ is to emphasize that the eigenvectors and count matrices are dependent on the hyper-parameters.

The variational approach to Markov processes placed reversible and stationary MSMs in a broader context of Koopman models which may or may not be reversible or stationary. In this context there is a family of variational scores, differentiated by a positive integer r:

$$\text{VAMP-r}(k, \boldsymbol{\theta}) = \left\| (\mathbf{U}^T\mathbf{C}_{00}\mathbf{U})^{-\frac{1}{2}}(\mathbf{U}^T\mathbf{C}_{0t}\mathbf{V})(\mathbf{V}^T\mathbf{C}_{tt}\mathbf{V})^{-\frac{1}{2}} \right\|_r^r, \tag{3}$$

where $\mathbf{C}_{tt}$ is the column-sum of the count matrix $[\mathbf{C}_{tt}]_{i,i} = \sum_i [\mathbf{C}_{0t}]_{i,j}$; $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors of the transition matrix. The functional dependence on $\boldsymbol{\theta}$ comes from its influence on the basis states which in turn determines the singular vectors; k is the number of singular vectors being scored and determines the dimensions of $\mathbf{U}$ and $\mathbf{V}$.

The matrix norm denotes takes the r^{th} power of the Schatten-r norm: where,

$$|\boldsymbol{T}|_r^r = \sum_i s_i^r(\boldsymbol{T}) \tag{4}$$

and s_i are the singular values of a matrix, $\boldsymbol{T}$.

If the data are stationary, reversible and r=1 this is equivalent to the GMRQ. With r=2 this expression measures the kinetic variance[51] captured by the basis sets. The VAMP scores have also been adapted to score the models based on the type of feature alone (rather than scoring the full MSM).[25]

As timescales are monotonic functions of the eigenvalues, maximizing the sum of the timescales also maximizes the VAMP scores.

Cross-validation and bootstrapping

Hyperparameters should be chosen to maximize the performance of a model on unseen data. Simply maximizing the variational score on the data used to fit the model (training data) may result in eigenvectors which describe this data well but do not generalize to new data generated by the same system. This is known as over-fitting and is a well-documented phenomenon.[52] To overcome this problem the estimated VAMP scores should be close to those attained on unseen data. One estimation method is to withhold a portion of the data (test set) and calculate the variational scores on this set. While accurate, it requires ignoring a large proportion of the data for training purposes, which may be wasteful when there are only a handful of observed transitions which we are interested in modeling.

Two other popular methods, which make more efficient use of the available data, are cross-validation[53] and bootstrapping.[54] The estimators for the variational scores (Equations 2 and 3) were both adapted to be used with cross-validation:[39,40] data is randomly split into two equally sized subsets. The eigenvectors $\mathbf{U}/\mathbf{V}$ are calculated on one set, while the count matrices $\mathbf{C}_{00/0t/tt}$ are calculated on the other set. This is repeated N_c times (e.g.,[25] $N_c = 50$) and an average of the VAMP scores is taken.

The bootstrap does not require a reformulation of estimators. Instead, a number, N_b, of new data sets are created from the original data set (e.g.,[54] $N_b = 100 - 1000$) and the mean or median of variational scores on each of these data sets used. To create the bootstrapped

data sets, trajectories are split into small independent subtrajectories. The subtrajectories are sampled *with replacement* to create a new bootstrapped data set of the same size as the original.

Hyperparameter optimization

Methods for optimizing hyperparameters

Finding the best set of hyperparameters $\boldsymbol{\theta}$ using either the VAMP scores or implied timescales (we will use the term *response* generally), is a black-box optimization problem. It is black box because we do not have access to the gradients, $\nabla_{\boldsymbol{\theta}}$ VAMP-r$(k, \boldsymbol{\theta})$, which would facilitate a gradient-based optimization. There are three broad classes of optimization techniques in this case: exhaustive searching, model-based searching and population-based algorithms.

Examples of exhaustive searching are grid search where hyperparameters are taken from a uniformly placed grid over the hyperparameter search space, and random search, where hyperparameters are randomly sampled from the search space.

Grid search is an effective strategy when the response is sensitive to all the hyperparameters. However, it has poor scaling with the number of hyperparameters (N^d, where N is the number of grid points per hyperparameter and d is the number of hyperparameters), so when only a small subset of hyperparameters are relevant, random search is more efficient.[28]

Model-based search algorithms construct surrogate models of the mapping between the hyperparameters and the model response which are cheap to evaluate and optimize, and use these models to guide hyperparameters to test. Examples include Bayesian optimization with either a Gaussian process or a tree Parzen estimator (TPE) as the surrogate model.[30] The third class of optimization algorithms is population algorithms, which include evolutionary algorithms,[32,34] particle swarm optimization[33,34] and covariance matrix adaption,[35] these will not be explored here further.

Bayesian optimization with tree-structured Parzen estimators

We chose tree-structured Parzen estimators to perform optimization because they easily handle numerical as well as categorical hyperparameters and can easily model conditional hyperparameter search spaces (i.e., choosing hyperparameters based on the choices of other hyperparameters) - this latter feature is the 'tree-structure' referred to in the name of the method.

Bayesian Optimization with TPE optimization proceeds as follows.

1. Randomly sample a small set of hyperparameters and measure the response of the resulting MSMs. This gives a hyperparameter trial data set $\mathcal{D}_n = \{(y_1, \boldsymbol{\theta}_1), \ldots (y_n, \boldsymbol{\theta}_n)\}$ where y is the model response.

2. Construct a model of the probability of the hyperparameters, given the response, $p(\boldsymbol{\theta}|y)$ as two separate probability density functions:

$$p(\boldsymbol{\theta} \mid y) = \begin{cases} \ell(\boldsymbol{\theta}) & \text{if } y > y^* \\ g(\boldsymbol{\theta}) & \text{if } y \leq y^* \end{cases}, \tag{5}$$

   where y^* is some user specified quantile, γ of the observations. l and g are probability models of the 'good' and 'bad' hyperparameters respectively and are explained more fully below.

3. To find the $n + 1^{\text{th}}$ value of $\boldsymbol{\theta}$ we maximize the *Expected Improvement*: $\text{EI}_{y^*}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}}\left[\max\left(y(\boldsymbol{\theta}) - y^*, 0\right)\right] \propto \left(\gamma + \frac{\ell(x)}{g(x)}(1-\gamma)\right)^{-1}$.

4. Evaluate $\boldsymbol{\theta}_{n+1}$ on the MSM and measure the response, y_{n+1}, add $(\boldsymbol{\theta}_{n+1}, y_{n+1})$ to the hyperparameter trial data set.

5. Repeat steps 2 to 4 until convergence in the maximum value of y is reached.

The functions l and g are Parzen estimators, otherwise known as kernel density estimators. These model the probability density of $\boldsymbol{\theta}$ by placing truncated Gaussian distributions

over each observation of a continuous hyperparameter, and a categorical distribution proportional to the observed counts of each level for each discrete hyperparameter. More details can be found in Bergstra et al.[30,38]

This method can also be extended to dual objective functions, i.e., when optimizing two (or more) responses, $\mathbf{y}_i = (y_i^1, y_i^2)$, for the same model. In this case the 'best' solutions form a Pareto set. Any member of a Pareto set, $\mathbf{y}_k$, has both responses superior to all other trials ($y_k^1 > y_i^1, y_k^2 > y_i^2$ for all i) but which are only superior to other members of the set in one response ($y_k^1 > y_{k'}^1$ or $y_k^2 > y_{k'}^2$ for k and k' in the Pareto set). For dual-objective optimization the acquisition function was the expected hyper-volume improvement.[55] This function tries to find hyperparameters which expand the Pareto set. The splitting of observations into two sets is complex, see Ozaki et al.[55] for details on the splitting algorithm.

Methods

To answer our research questions we estimated a large number of Markov state models with different hyperparameters, measured their observables and analysed the results. The workflow may be summarized as follows. We used existing molecular dynamics trajectories of Chignolin and BBA and fit MSMs with 140 randomly sampled hyperparameters (*hyperparameter trials*) and recorded implied timescales, eigenvalues, and VAMP-2 scores for a range of different lag times (τ). Each observable was estimated with confidence intervals using bootstrapping. This data constituted our *hyperparameter trial data set* and was analyzed in the first, second and last results subsections. A 'toy' three-state MSM model was constructed to highlight with the VAMP-2 score, for reversible transition matrix estimation. We then performed Bayesian optimization with a TPE surrogate function, with a variety of different objective functions, and using the hyperparameter trial data set to initialize the surrogate function. These results are discussed in the third part of the results.

Molecular dynamics

We use simulation data of the fast-folding proteins Chignolin and BBA, two of the twelve fast-folding proteins which have become the *de facto* benchmark data set for testing molecular kinetics methods. The methods used to create this data are described elsewhere.[56] Important information on the data is shown in Table 1: the average folding time was calculated by the authors;[56] the sub-trajectory length and number of sub-trajectories correspond to the data splitting used in the bootstrapping procedure.

Table 1: **Description of molecular dynamics data**

Name	PDB	Simulation time (µs)	Average folding time (µs)	No. Residues	Sub-trajectory length (µs)	No. sub-trajectories
BBA	1FME	325	18	28	2	164
Chignolin	5AWL	106	0.6	10	2	53

Markov state models

MSMs were estimated using PyEMMA version 2.5.7[57] and used a standard pipeline when focusing on the slow relaxation processes:[21,26]

1. Project molecular dynamics (MD) trajectories onto a set of features.
2. Reduce the dimension of the feature trajectories using TICA with a lag time τ_{TICA} by projecting onto the first m TICA coordinates.
3. The frames of the TICA trajectories were clustered using the k-means algorithm into n discrete microstates.
4. A reversible, maximum likelihood MSM was then estimated.

To save on memory and compute resources the data was subset in parts of the MSM estimation. The MD trajectories were first strided so that the time between each frame was

1 ns in line with previous analysis in the literature.[a reference here] The cluster centers were estimated on frames separated by 10 ns, i.e. only the 0^{th}, 10^{th}, etc. frames were used for estimating the cluster centers.

The uncertainty for model observables was estimated using bootstrap with 100 bootstrap samples. The point estimate and error bars were calculated as the median, 2.5 % and 97.5 % quantiles of the distribution from the bootstrap samples.

Hyperparameters and scoring

140 different hyperparameters were randomly sampled from the search space described by Table 2. Each set of hyperparameters and their corresponding model observables are known as a *hyperparameter trial*. Three different features, f, were used:

1. dihedrals feature ('dihed.'): the sine and cosine of the ϕ, ψ and χ_{1-5} angles of the amino acid residues;

2. contact distance feature ('dist.'): the distance between all pairs of residues separated by three or more residues;

3. logistic distance feature ('logit(dist.)'): the same as feature 2 but with a logistic transform applied to the distance (d): $\text{logit}(d) = [1 + \exp(s(d - c))]^{-1}$, where center, c, and steepness s, have units of Å and Å^{-1} respectively.

The logistic distance feature may be described as a 'soft' or 'fuzzy' contact map: it takes on the value 0 for $d \gg c$ and a value of 1 for $d \ll c$, and varies between these two extremes in the neighborhood of c with a steepness determined by s. The definitions of the contact distances (d) were either the closest heavy-atom distance (X-X) or the distance between the α-Carbons (Cα-Cα). The TICA eigenvectors were scaled by their eigenvalues (λ) so that distances in TICA space correspond to kinetic distances.[51]

The number of trials was approximately proportional to the number of hyperparameters for each feature: 20 trials for the dihedral feature, 40 for the contact distances (20 for each

value of the contact distance scheme: X-X, Cα-Cα), and 80 for the logistic transformation of contact distances (which, in addition to the two distance scheme values, has two other hyperparameters, c and s).

For each trial, $\boldsymbol{\theta} = (f, \tau_{\mathrm{T}}, m, n, c, s)$, an MSM was estimated using the procedure above with a range of Markov lag-times, τ: 1 ns, 11 ns, ..., 101 ns. For each combination of $\boldsymbol{\theta}$ and τ the slowest 2 to 21 eigenvectors were scored using the VAMP-2$(k, \boldsymbol{\theta})$ (Equation 3) and VAMP-2$_{eq}(k, \boldsymbol{\theta})$ score (Equation 6):

$$\text{VAMP-2}_{\text{eq}}(k, \boldsymbol{\theta}) = \sum_{i=1}^{k} \lambda_i^2, \tag{6}$$

where λ are the eigenvalues of the MSM transition matrix which obey detailed balance, along with the implied timescales, t_i. Each of these observations was estimated as the median of $N_b = 100$ bootstrapped samples.

VAMP-2$(k, \boldsymbol{\theta})$ and VAMP-2$_{eq}(k, \boldsymbol{\theta})$ will be abbreviated as VAMP2(k) and VAMP2$_{eq}(k)$ from here on, the dependence on $\boldsymbol{\theta}$ being assumed.

Selected models were validated by:

1. inspection of structures sampled from microstates which had the most extreme values of ψ_2;

2. inspection of both ψ_2 and a two-state coarse grained model in the space of the first two TICA components;

3. a plot of the mean first passage time as a function of the lag time (as suggested in Suarez et al.[58]);

4. implied timescales as a function of the lag time, τ.

This information can be found in the supplementary information.

The hyperparameter trial data set, $\mathcal{D}$, consisted of 100 bootstrap samples of 140 unique sets of hyperparameters, at 10 different lag times, with 20 measurements of the implied

Table 2: **Hyperparameter search space**. X-X and Cα-Cα refer to the closest heavy atom and α-Carbon scheme respectively, for measuring the contact distance (*dist.*). The sine and cosine of dihedral angles were used as features.

Features, (f)				
Dihedral angles	WHICH *dihed.* $= \phi, \psi, \chi_1, \ldots, \chi_5$ (sine and cosine transformation)			
Contact distances	DEFINITION, (d)	TRANSFORM	CENTER (c, Å)	STEEPNESS (s, Å^{-1})
	• X-X • Cα-Cα	• logit(*dist.*) • *dist.*	3 to 15	0.01 to 5
Decomposition	EIGENVECTORS, (m)	LAG-TIME, (τ_T, ns)	SCALING	
TICA	1 to 20	1 to 100	λ	
Clustering	CLUSTERS, (n)			
k-means	10 to 1000			

timescales and 20 measurements of the VAMP2(k) score and 20 measurements of the VAMP2$_{eq}$(k) score. The total number of these observations (t_i, VAMP2(k), VAMP2$_{eq}$(k)) is therefore 8 400 000.

Markov lag time

The Markov lag time, τ, was calculated from the total hyperparameter trial data set. For each trial the following gradient was calculated:

$$g(\tau, \theta) = \frac{\Delta \log\left(t_2(\tau, \theta)\right)}{\Delta \tau}, \tag{7}$$

The selected Markov lag-time, τ^* was chosen as:

$$\tau^* = \arg\min_{\tau,\theta} \left[g(\tau, \theta)\right], \quad 0 < g < \log 1.01. \tag{8}$$

A graphical representation of this process is shown in Figure 1.

This codifies and extends the generally accepted process by which the implied timescales

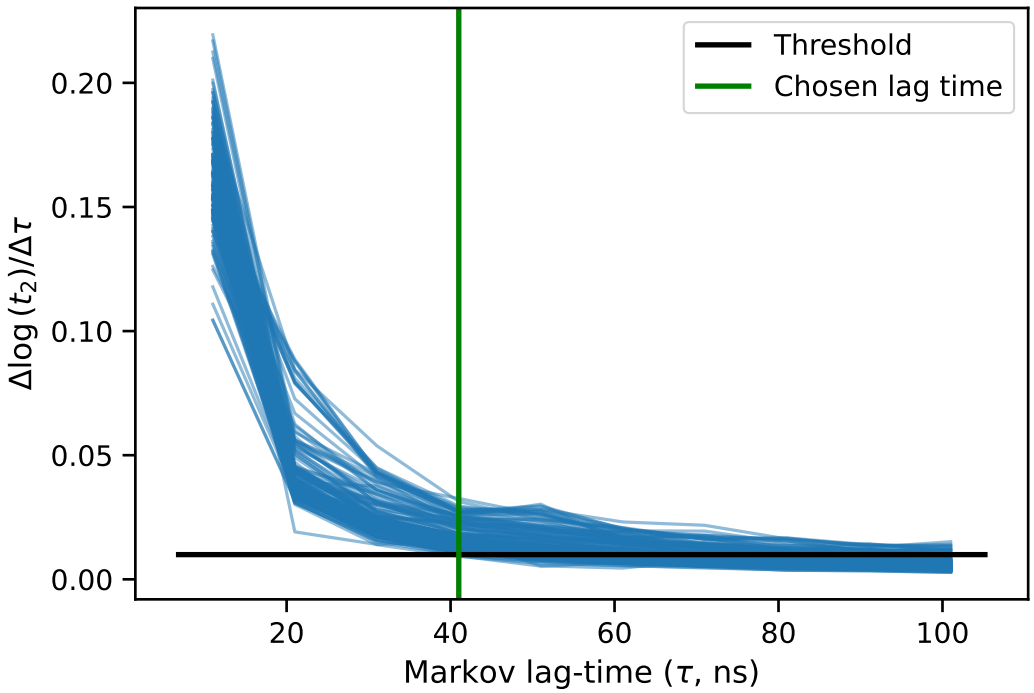

Figure 1: MARKOV LAG TIME SELECTION. Each blue line is the median of the gradient defined in Equation 7 taken over the bootstrapped samples. The black horizontal line is the threshold for convergence. The green vertical line is the selected lag time. The data represented here is from BBA, the same method applies for Chignolin.

t_i as a function of τ are plotted on a log scale and the smallest τ for which t_2 is constant is chosen. Our extension is that we consider a range of different values of θ.

Optimization

We used Bayesian optimization to optimize the full hyperparameter feature space described in Table 2. We used the tree-structured Parzen estimator as the surrogate function, as implemented in the Python package *Optuna* version 3.0.3[59] which was also used to perform the optimizations. The optimization runs were initialized with data from the randomly sampled hyperparameter trial data set. The objective functions were estimated as the median from 20 bootstrap samples. Four different objective functions were used for each protein, two single objective and two dual-objective functions, these were

1. t_2: the timescale dominant process,
2. $\mathrm{VAMP2}_{eq}(2) = 1 + \lambda_2^2$: the 'equilibrium' VAMP2 score of the 2$^{\mathrm{nd}}$ (dominant) process
3. t_2 and t_2/t_3: a multi-objective function of the timescale of the dominant process and the gap between the 2$^{\mathrm{nd}}$ and 3$^{\mathrm{rd}}$ timescale.
4. $\mathrm{VAMP2}_{eq}(2)$ and $\mathrm{VAMP2}_{eq}(2)/\mathrm{VAMP}_{eq}(3)$: a multi-objective function of the equilibrium $\mathrm{VAMP2}_{eq}(k)$ score of the 2$^{\mathrm{nd}}$ process and the gap between the 2$^{\mathrm{nd}}$ and 3$^{\mathrm{rd}}$ process.

In the case of single objective optimization the acquisition function was the *expected improvement*; in the case of multi-objective optimization the acquisition function was the *expected hyper-volume improvement*.[55] The quantile for splitting observations into 'good' and 'bad' trials was set at 25 %. This information is summarized in Table 3. The number of initial observations is less than the full 140 hyperparameter trials because a) some trials failed to converge an MSM, and b) in the case of Chignolin, some MSMs did not have a resolvable value of t_3.

Table 3: Hyperparameter optimization tasks.

Protein	**Objective Functions**	**Initial Data**	**No. Trials**
Chignolin	t_2	131	95
Chignolin	t_2, t_2/t_3	55	141
Chignolin	VAMP2$_{eq}$(2)	131	100
Chignolin	VAMP2$_{eq}$(2), VAMP2$_{eq}$(2)/VAMP2$_{eq}$(3)	55	150
BBA	t_2	136	100
BBA	t_2, t_2/t_3	136	100
BBA	VAMP2$_{eq}$(2)	136	100
BBA	VAMP2$_{eq}$(2), VAMP2$_{eq}$(2)/VAMP2$_{eq}$(3)	136	100

The code used to create the hyperparameter trial data set, $\mathcal{D}$ can be found at `https://github.com/RobertArbon/msm_sensitivity` and the code used to perform all other analyses can be found at `https://github.com/RobertArbon/msm_sensitivity_analysis`.

Results and discussion

Having created the hyperparameter trial data set we first highlight some inconsistencies in the VAMP2 scores; then we show results for optimization using random selection and Bayesian optimization; and finally we determine what effect the lag time and number of scored eigenvectors makes on model selection.

VAMP2(k) scores of reversible MSMs give inconsistent results

The VAMP2(k) score[40] provides a principled metric for optimizing MSM hyperparameters. The benefits are that it can be used for stationary, non-stationary, reversible and non-reversible MSMs. It is linked directly to the kinetic variance captured by the basis states such that maximizing the VAMP2(k) score will maximize the timescales of pertaining to the first k eigenvectors of the model. In addition, it can be used with bootstrapping and cross-validation techniques for assessing generalizability.

Inspection of the VAMP2(k) and t_2 values in the hyperparameter trial data set for BBA revealed that for some subsets of the trials, VAMP2(2) was inversely proportional to t_2. An example of this is shown in Figure 2. In panel (a) the VAMP2(2) score is shown for the trials ranked first, third, and fourth. In panel (c) the first five timescales are shown for each model. Timescales for the third to sixth eigenvectors are similar for each trial, however t_2 clearly *increases* with *decreasing* VAMP2(2) score. The second-ranked model is omitted for clarity because it does not follow this pattern. We suggest the reason for this behavior is due to the fact by enforcing reversibility in the estimation of the transition matrix it is difficult to get numerical consistency between the three count matrices ($\mathbf{C}_{00/0t/tt}$) and the eigenvectors ($\mathbf{U}/\mathbf{V}$) in Equation 3.

To ensure that this phenomenon was not an artifact of the processing pipeline the effect was replicated with a three-state toy model (example 1 in Trendelkamp et al.[22]). $10\,000 \times 20$-step trajectories were sampled from the same 3×3 transition matrix and for each trajectory

count matrices ($\mathbf{C}_{00/0t/tt}$) were calculated. We assert that the differences in the count matrices arising from the finite sampling in this toy model are similar to the differences from different discretization schemes in the example of BBA. From each set of count matrices t_2 and VAMP2(2) scores were estimated and these are shown in Figure 3 panel (a). While t_2 is clearly rank-correlated with VAMP2(2), the rank correlation is not perfect. Many subsets of these results form sets which are anti-correlated, three examples of this inverse relationship are shown as black lines labeled 'Inverse'. These subsets mirror the effect seen in the BBA models in Figure 2. As a comparison, in panel (b) we plot the sum of the squares of the first two eigenvalues, VAMP2$_{eq}$(2), which shows perfect rank correlation (as they must).

The reason for writing the VAMP2(k) score as the product of count matrices and eigenvectors/singular vector matrices is to facilitate data-splitting in cross-validation. While we used bootstrapping for this work and thus mitigated this, the effect of data splitting would be to worsen the discrepancy between the count and transition matrices. This is because the count matrices are now estimated on different data compared to the eigenvector matrices.

Due to the problem of consistency between the matrices in Equation 6 arising from a) enforcing reversibility and b) data splitting for cross-validation, we recommend that VAMP2(k) scores, either cross-validated or bootstrapped, should not used for reversible and stationary MSMs. Instead, we recommend bootstrapping the sum of the squared eigenvalues (VAMP2$_{eq}$(k)) directly from the reversible transition matrix. This has the same theoretical properties of the VAMP2(k) score (i.e., represents captured kinetic variance, and link to variational theorem) while not a) wasting data due to data splitting and b) perfect correlation with the implied timescales.

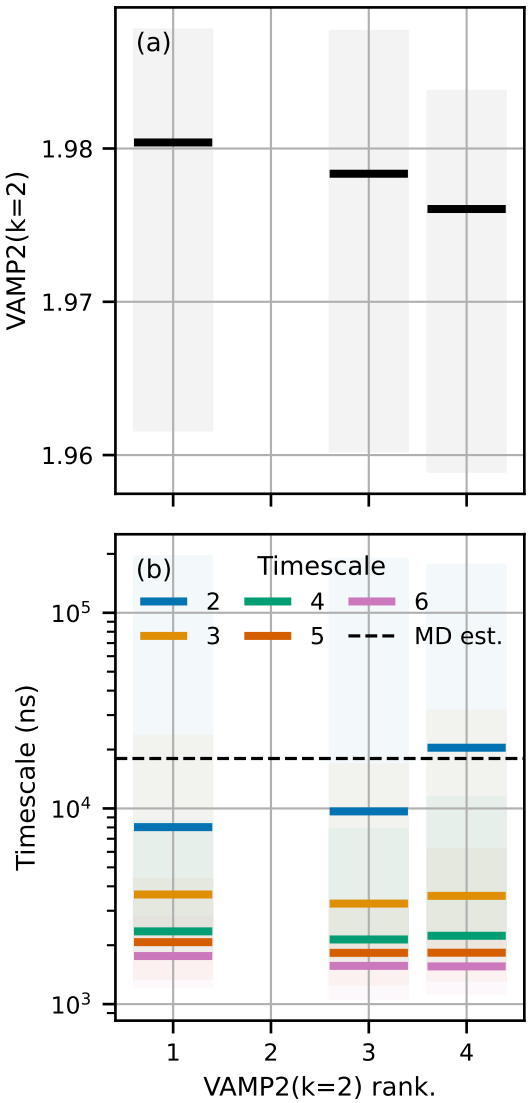

Figure 2: **Models with VAMP2(k) scores inversely proportional to timescales**. (a) shows the VAMP2(2) scores and (b) shows the first five dominant timescales, for a selection of models of BBA. The horizontal axis in both panels is the model rank as judged by the VAMP2(2) score. The selection shows models where the slowest timescale is inversely proportional to the VAMP2(2) score. Models which do not show this correlation are not shown.

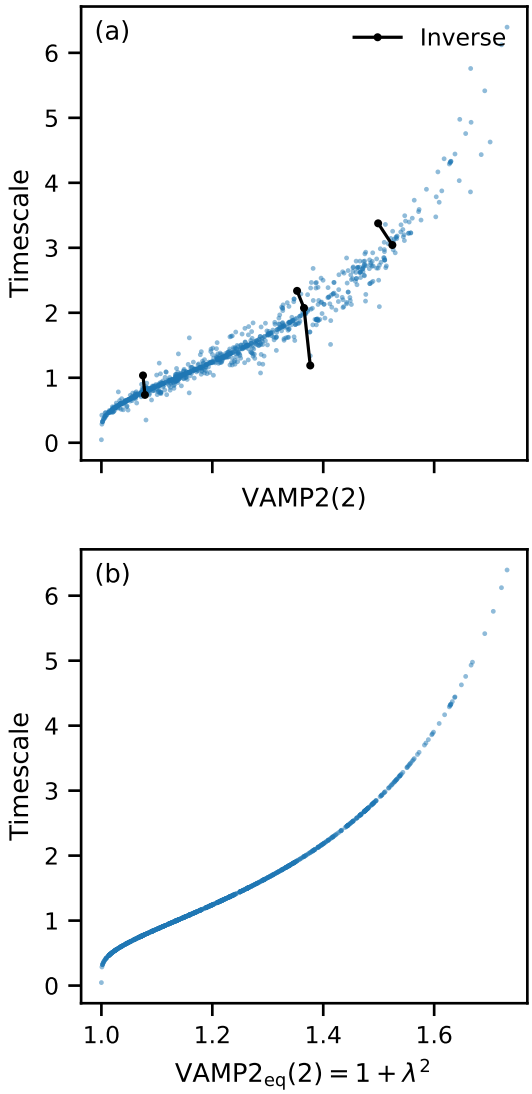

Figure 3: Relationship between implied timescales, t_2, VAMP2(2), VAMP2$_{eq}$(2) scores. Each of the 1000 blue points is calculated from an MSM estimated from a distinct simulated trajectory of 20 time steps. The trajectories were generated from the same three-state reference transition matrix (taken from Trendelkamp-Schroer et al.[22]). The estimated transition matrices were all estimated ensuring reversibility. (a) shows t_2 as a function of VAMP2(2) scores while (b) shows t_2 as a function of VAMP2$_{eq}$(2). The black points labeled 'Inverse' are example subsets of MSMs where the relationship between the implied timescale and VAMP2(2) score are inverted.

Eigenvectors may change definition with change in hyperparameters

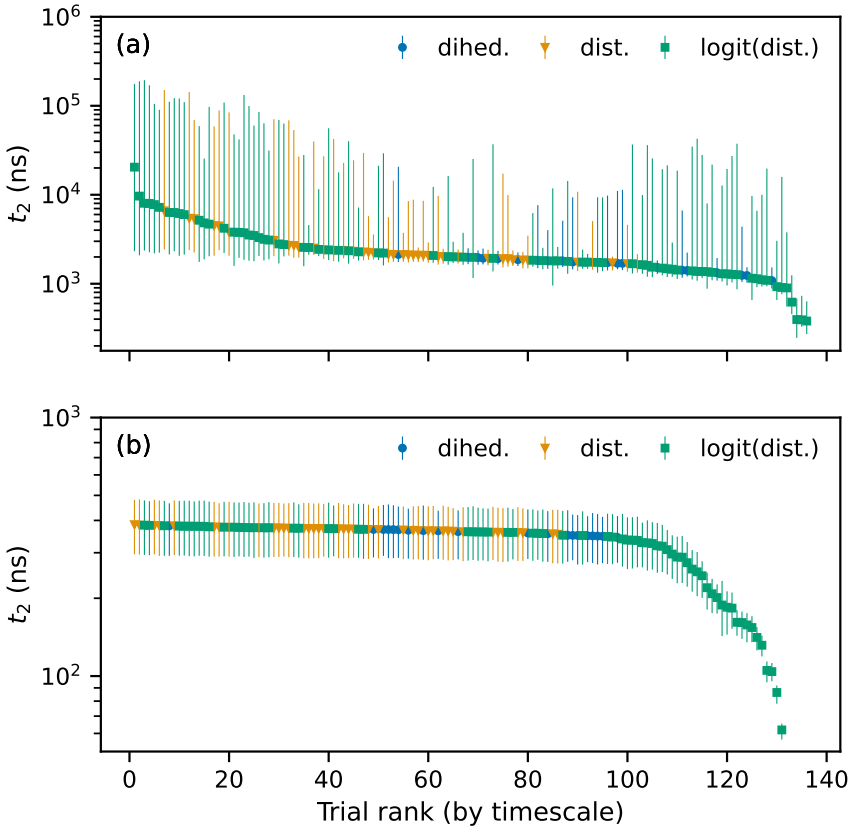

Figure 4: TIMESCALES OF RANDOMLY SAMPLED HYPERPARAMETER TRIALS. Panel (a) refers to BBA, panel (b) refers to Chignolin. The vertical axis is the dominant timescale (t_2), the horizontal axis is the trial rank. The solid disc and error bars are the median and 95 % bootstrapped confidence intervals.

Figure 4 shows the results of the optimization through random selection: the distribution of the timescale (t_2) of the dominant process (corresponding to the 2$^{\text{nd}}$ right-eigenvector of the MSM transition matrix, ψ_2) for each MSM of BBA (panel (a)) and Chignolin (panel (b)), ordered left to right with highest value of t_2 on the left. Each point is colored according to the feature used. According to previous research[25] we expect in both cases that the dominant relaxation process to correlate with the folded-to-unfolded transition. We expect that the best model would be the model with the highest value of t_2. However, implicit in this decision is that the implied timescale represents the same underlying relaxation process.

For Chignolin, both t_2 and the corresponding relaxation process for different hyperparameters trials are similar, as can be seen by inspection of the models 1 and 2 (ranked 1 and 4 respectively in Figure 4, see Table S1 and sections S2.1 and S2.2 for more detailed information on each model). In model 1, Figure 5(a), shows ψ_2 as an unfolded-folded transition. In model 2, Figure S6 (c) shows ψ_2 as transitioning between a structure which is almost completely unfolded with only two non-native contacts and the folded state.

In the case of BBA, the situation is different. The top two best performing models, models 1 and 2 (ranked 1 and 2 respectively in Figure 4, see Table S1 and sections S2.4 and S2.5) show evidence of optimizing a similar relaxation mode, ψ_2. Comparing models 1 and 2 we see a similar folded-unfolded transition (Figure 5 panel (a) and (b)). Model 1 is more accurate, as the values of t_2 show: 20.4 µs, 95 %C.I. [2.3—176.2 µs] cf. 9.7 µs, 95 %C.I. [2.1—188.7 µs]. The accords with differences in the hyperparameters: model 1 and 2 use the logistic distances feature, but model 1 has a logistic transform which is more sensitive to changes in contact distances between 0.1 Å—10 Å (see Figure S1) and more discrete basis functions (471 cf. 289, see Table S1).

However, the best-performing models with the other features have markedly different ψ_2 which do not correspond to the same transition as models 1 and 2. Model 3 is the best model with the distance feature (ranked 7 in Figure 4, see Table S1 and section S2.7) and model 4 is the best model with the dihedrals feature (ranked 54 in Figure 4, see Table S1 and section S2.8). Both of these show markedly different transitions for ψ_2, see Figures S21(c) and S24(c) respectively.

Thus, when optimizing MSMs the objective function (t_2 in this case), ψ_2 may change definition across the search space and one is not comparing like-with-like when looking at *just* the objective function.

To mitigate this problem, we advocate checking the character of the eigenvectors when selecting appropriate hyperparameters to ensure one is optimizing at least a consistent set of relaxation processes.

Bayesian optimization may optimize different processes

We tested whether Bayesian optimization could increase t_2 by selecting better hyperparameters. We optimized the search space in Table 2 using both single-objective and dual-objective optimization, with objectives based on the timescales, $t_{2/3}$ and the VAMP2$_{eq}$(2/3) scores, see Table 3 for a list of optimization experiments. The optimization using dual objectives of t_2 with t_2/t_3 (and the VAMP2$_{eq}$ equivalent) was prompted by the observation that from the randomly sampled hyperparameter trial data set, there were many models with similar values of t_2 but with a wide range of timescale gaps t_2/t_3. A large timescale gap gives rise to models which are more accurate when truncated and coarse-grained into a two-state model. Our hope was therefore to bias the optimization results in favor of both large t_2 and a large separation of timescales. The optimization trajectories, which show the largest value of t_2 (vertical axis) in all trials up to the current trial number (horizontal axis) are shown in Figure S32.

Single objective optimization of both t_2 and VAMP2$_{eq}$(2) increased t_2 for Chignolin and BBA. For Chignolin the increase was modest, between 2.4 %—5.5 % for all four objective functions. The single objective optimization of t_2 had the smallest increase (Figure S32, panel (a) red squares) while the multi-objective optimization of the VAMP2$_{eq}$(2) and VAMP2$_{eq}$(2)/VAMP2$_{eq}$(3) gave the largest increase in t_2 (Figure S32, panel (c) blue squares), although the increase in the gap was modest (see Figure S33 (c)).

The small t_2 increase for Chignolin is unsurprising given the consistency of t_2 across the randomly sampled hyperparameter trials. However, the t_2 optimized MSM, model 3 (see Table S1 and section S2.3) shows a partially folded to folded transition in Figure 5(b) rather than the fully unfolded to folded transition in model 1 (Figure 5(b)). In terms of the values of the hyperparameters, the optimization has changed the TICA lag-time significantly (from 71 ns in model 1 to 3 ns in model 3 - the other hyperparameters have remained largely unchanged). The VAMP2$_{eq}$(2) and VAMP2$_{eq}$(2)/VAMP2$_{eq}$(3) optimized MSM, model 4 (see Table S1 and section S2.4) shows the same unfolded—folded transition as model 1 (see

Figure 5(c). Both the dual-objective optimizations (t_2 with t_2/t_3 and the VAMP equivalent) increased both the t_2 and the separation of timescales (see Figure S33).

For BBA the single objective optimization of t_2 and VAMP2$_{eq}$(2) increased t_2 by 128 % and 135 % respectively. However, these models have not optimized the same relaxation process as the incumbent from the randomly sampled hyperparameters, model 1. The t_2 optimized MSM, model 5 (see Table S1 and section S2.9), denotes a transition between two misfolded structures (see Figure 5(f). This is perhaps surprising given that the main difference between the two model specifications is that change in the logistic transform (see Figure S1 for the difference between model 1 and model 5's logistic transform). In contrast the VAMP2$_{eq}$(2) optimized model shows a similar transition to models 1 and 2 (see Figure S30(c)). Both the dual objective optimization runs did not improve t_2 significantly, although, in the case of optimization of t_2 with t_2/t_3, the gap increased significantly (see Figure S33(b)).

The implications for Bayesian optimization are similar to the lessons learned from random optimization: changing hyperparameters can change the optimized process, meaning one must analyze the character of the eigenvectors to ensure one is optimizing the same processes.

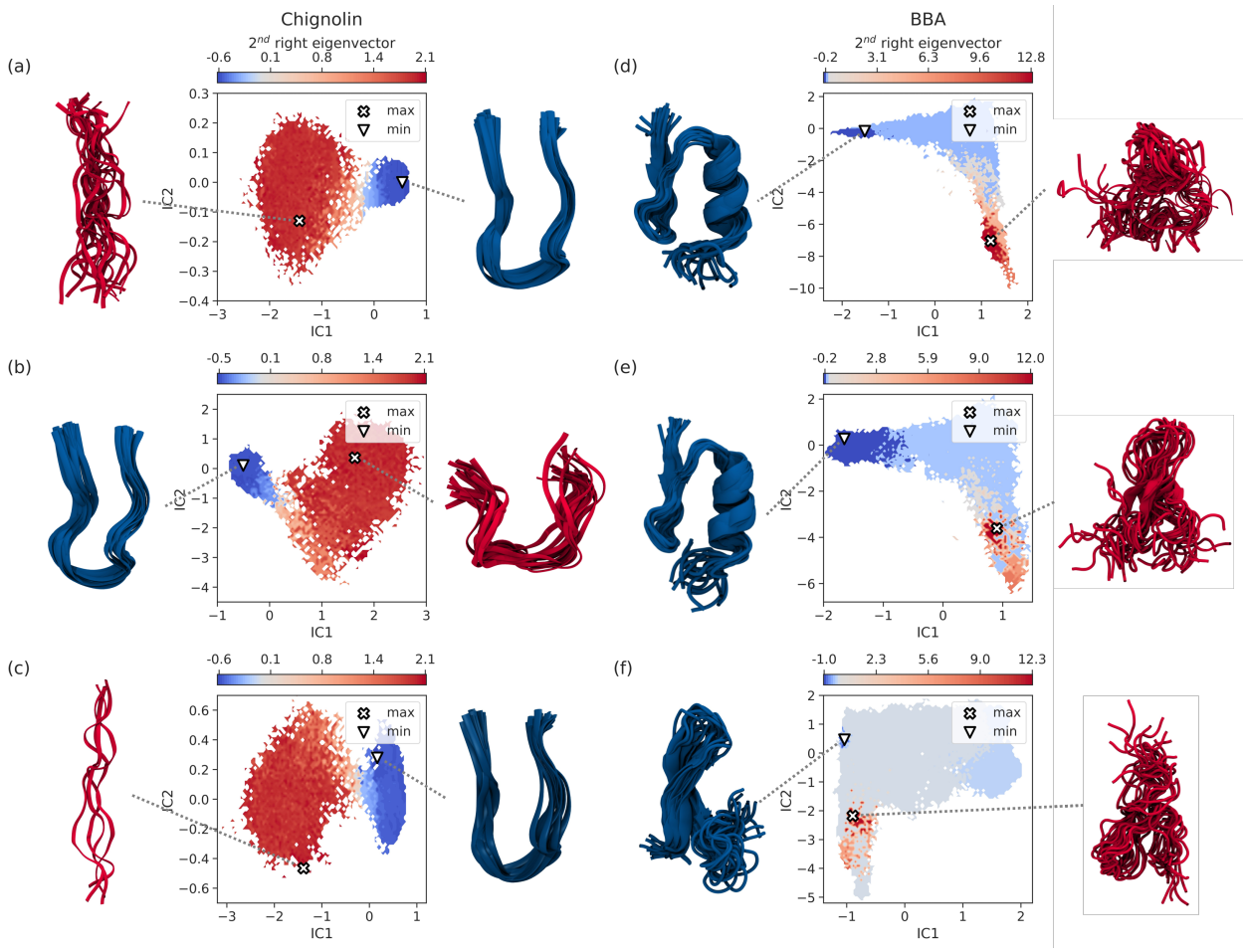

Figure 5: MSMs WITH DIFFERENT RELAXATION PROCESSES. Each panel shows ψ_2 in the space of the first two TICA coordinates. Also shown are an ensemble of structures sampled from the microstates with the extreme values of ψ_2. Panel (a): Chignolin model 1, (the largest median t_2 from random sampling); panel (b): the Chignolin model 3 (the largest median t_2 from Bayesian optimization of t_2); panel (c): the Chignolin model 4 (the largest t_2 from Bayesian optimization of VAMP2$_{eq}$(2) and VAMP2$_{eq}$(2)/VAMP2$_{eq}$(3)); panel (d): BBA model 1 (the largest median t_2 from random sampling); panel (e): BBA model 2 (the second largest median t_2 from random sampling); panel (f): BBA model 5 (the largest median t_2 from Bayesian optimization of t_2).

The lag time and number of scored eigenvectors do not affect model selection.

When evaluating MSMs using a variational score one must specify both the Markov lag time (τ) and the number of eigenvectors to score (k). However, both these choices affect the VAMP score although it is not clear whether these choices affect the model ranking. To test how these choices affect model selection we measured the consistency in model rank for BBA, as measured by the VAMP2$_{eq}$(k), using the Spearman's rank correlation coefficient at: a) different lag times for given values of k and b) at different values of k at a given lag time.

Figure 6 shows the consistency between BBA model rankings at different lag times ($1\,\text{ns} \leq \tau \leq 101\,\text{ns}$) with k=2 (panel (a)) and with k=10 (panel (b)). In addition, scatter plots of the data used to calculate these coefficients for k=2, 3, 5, & 10 are shown in Figures S33 - 37. Across all lags and for both small (k=2) and large (k=10) numbers of scored eigenvectors, the consistency in the model ranking is high (greater than 85 %). The consistency between models with lag times $\tau > 1\,\text{ns}$ is much greater, with rank correlations up to 100 %. This effect is most pronounced for k=10 scored eigenvectors. In particular, good consistency is achieved at lag times smaller than those required for the model to be Markovian (τ=41 ns).

Figure 7 shows the consistency between model rankings at different number of scored eigenvectors ($2 \leq k \leq 21$) at a lag time of 41 ns (the value used in all previous analysis for BBA). Again, the consistency is generally high with a rank correlation between all pairs of k of at least 80 %. The ranking is most consistent between values of k larger than 4. From these two analyses taken together, we see that for long lag-times and a large number of scored eigenvectors model ranking is significantly affected by the choice of τ and k.

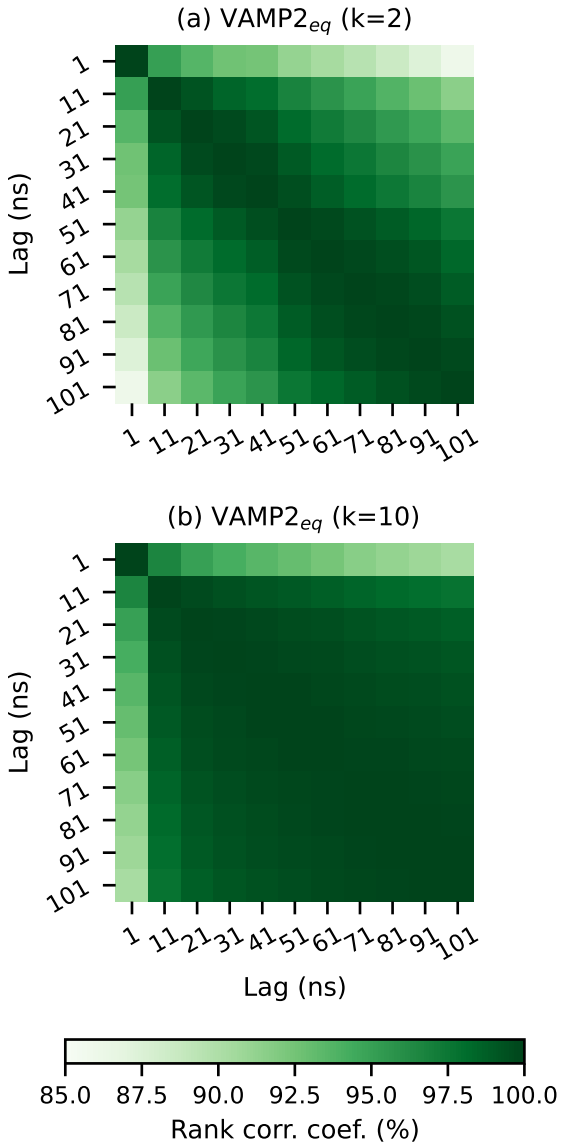

Figure 6: Consistency of VAMP2$_{\mathrm{eq}}(k)$ rank with Markov lag time, τ for BBA. The i, j^{th} cell in panel (a) shows the Spearman's rank correlation coefficient of VAMP2$_{\mathrm{eq}}(2)$ for each trial measured at the i^{th} lag time, with VAMP2$_{eq}(2)$ measured at the j^{th} lag time. Panel (b) show the same measurements with VAMP2$_{\mathrm{eq}}(10)$ score respectively.

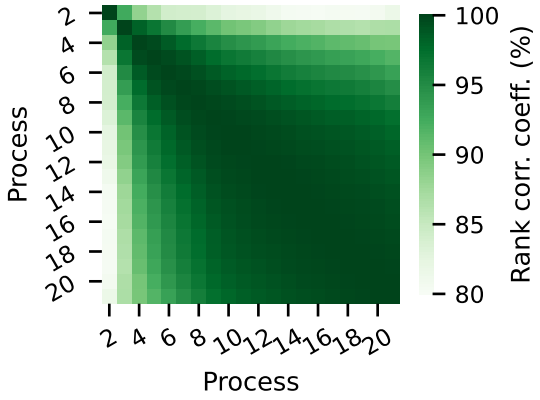

Figure 7: Consistency of VAMP2$_{eq}(k)$ rank with number of scored eigenvectors for BBA. The ranks of trials in the row k are compared to their rank at the column k using the Spearman's rank correlation coefficient at a lag time of 41 ns

Conclusions

This work has drawn a complex picture of MSM optimization which suggests that model selection of MSM hyperparameters by inspection of a single objective measure (e.g., t_2) is not advisable as changes in hyperparameters can change the physical meaning of the MSM eigenvectors. The commonly used VAMP2 metric when used with the assumption of reversibility can give rise to rankings of MSMs which are inconsistent with the implied timescales. We suggest that this is due to a numerical, rather than theoretical problem. In its place, we suggest using the sum of the square eigenvalues, or other model observables, e.g., t_2, with bootstrapping in order to estimate uncertainty.

Bayesian optimization of MSM hyperparameters is possible using tree Parzen estimators for the surrogate function. TPEs are able to easily model the different types of hyperparameters (continuous, integer and discrete) and improve the implied timescales. Multi-objective optimization can be used but does not give a clear advantage over optimizing a single objective. Caution must again be exercised as this can give rise to models in which the meaning of the eigenvectors can change.

We also showed that selecting lag-times and a number of scored eigenvectors in the objective function does not drastically change the the ranking of the hyperparameters as long as both are sufficiently large.

Taken together these observations suggest a number of recommendations:

1. Randomly sample a range of hyper-parameters and use the VAMP2$_{eq}$ or t_2 (or the timescale of interest) to rank hyperparameters.

2. Use a small subset of models with different lag times (τ) and score with a range of eigenvectors (k) and choose τ and k such that VAMP2$_{eq}$ is independent of both.

3. Inspect eigenvectors to check for consistency across different hyperparameters.

4. Bayesian optimization of t_2 or VAMP2$_{eq}$ can be used to optimize hyperparameters but

the eigenvectors must be inspected for consistency.

Data Availability

The molecular dynamics trajectories used in this work were used with permission from D. E. Shaw Research.

Acknowledgement

The authors would like to thanks Andreas Mardt for useful discussions and Redesign Science for partial support of this work.

Supporting Information Available

All processing and analysis for the experiments carried out and instructions on how to reproduce this work can be found at `www.github.com/RobertArbon/msm_sensitivity` and `www.github.com/RobertArbon/msm_sensitivity_analysis`.

References

(1) Cannariato, M.; Miceli, M.; Cavaglià, M.; Deriu, M. A. Prediction of Protein–Protein Interactions Between Alsin DH/PH and Rac1 and Resulting Protein Dynamics. *14*.

(2) Chakrabarti, K. S.; Olsson, S.; Pratihar, S.; Giller, K.; Overkamp, K.; Lee, K. O.; Gapsys, V.; Ryu, K.-S.; de Groot, B. L.; Noé, F.; Becker, S.; Lee, D.; Weikl, T. R.; Griesinger, C. A litmus test for classifying recognition mechanisms of transiently binding proteins. *13*, 3792.

(3) Koulgi, S.; Jani, V.; V N, M. U.; Sonavane, U.; Joshi, R. Structural insight into the binding interactions of NTPs and nucleotide analogues to RNA dependent

RNA polymerase of SARS-CoV-2. *0*, 1–15, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07391102.2021.1894985.

(4) Dutta, S.; Selvam, B.; Shukla, D. Distinct Binding Mechanisms for Allosteric Sodium Ion in Cannabinoid Receptors. *13*, 379–389, Publisher: American Chemical Society.

(5) McKiernan, K. A.; Koster, A. K.; Maduke, M.; Pande, V. S. Dynamical model of the CLC-2 ion channel reveals conformational changes associated with selectivity-filter gating. *16*, e1007530, Publisher: Public Library of Science.

(6) Ibrahim, M. T.; Trozzi, F.; Tao, P. Dynamics of hydrogen bonds in the secondary structures of allosteric protein Avena Sativa phototropin 1. *20*, 50–64.

(7) Hu, X.; Pang, J.; Zhang, J.; Shen, C.; Chai, X.; Wang, E.; Chen, H.; Wang, X.; Duan, M.; Fu, W.; Xu, L.; Kang, Y.; Li, D.; Xia, H.; Hou, T. Discovery of Novel GR Ligands toward Druggable GR Antagonist Conformations Identified by MD Simulations and Markov State Model Analysis. *9*, 2102435, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.202102435.

(8) Pantsar, T.; Kaiser, P. D.; Kudolo, M.; Forster, M.; Rothbauer, U.; Laufer, S. A. Decisive role of water and protein dynamics in residence time of p38-alpha MAP kinase inhibitors. *13*, 569, Number: 1 Publisher: Nature Publishing Group.

(9) Hempel, T.; Raich, L.; Olsson, S.; Azouz, N. P.; Klingler, A. M.; Hoffmann, M.; Pöhlmann, S.; Rothenberg, M. E.; Noé, F. Molecular mechanism of inhibiting the SARS-CoV-2 cell entry facilitator TMPRSS2 with camostat and nafamostat. *12*, 983–992, Publisher: The Royal Society of Chemistry.

(10) Tosstorff, A.; Peters, G. H. J.; Winter, G. Study of the interaction between a novel, protein-stabilizing dipeptide and Interferon-alpha-2a by construction of a Markov state model from molecular dynamics simulations. *149*, 105–112.

(11) Liu, Q.; Wang, Y.; Lai-Han Leung, E.; Yao, X. In silico Study of Intrinsic Dynamics of Full-length apo-ACE2 and RBD-ACE2 complex.

(12) Fernández-Quintero, M. L.; Kroell, K. B.; Hofer, F.; Riccabona, J. R.; Liedl, K. R. Mutation of Framework Residue H71 Results in Different Antibody Paratope States in Solution. *12*.

(13) Sharma, N.; Sonavane, U.; Joshi, R. Comparative MD simulations and advanced analytics based studies on wild-type and hot-spot mutant A59G HRas. *15*, e0234836, Publisher: Public Library of Science.

(14) Juárez-Jiménez, J.; A. Gupta, A.; Karunanithy, G.; S. Mey, A. S. J.; Georgiou, C.; Ioannidis, H.; Simone, A. D.; N. Barlow, P.; N. Hulme, A.; D. Walkinshaw, M.; J. Baldwin, A.; Michel, J. Dynamic Design: Manipulation of Millisecond Timescale Motions on the Energy Landscape of Cyclophilin A. *Chemical Science* **2020**, *11*, 2670–2680.

(15) Wapeesittipan, P.; Mey, A. S. J. S.; Walkinshaw, M. D.; Michel, J. Allosteric Effects in Cyclophilin Mutants May Be Explained by Changes in Nano-Microsecond Time Scale Motions. *Communications Chemistry* **2019**, *2*, 1–9.

(16) Paul, F.; Thomas, T.; Roux, B. Diversity of Long-Lived Intermediates along the Binding Pathway of Imatinib to Abl Kinase Revealed by MD Simulations. *16*, 7852–7865, Publisher: American Chemical Society.

(17) Zhou, M.; Wen, H.; Lei, H.; Zhang, T. Molecular dynamics study of conformation transition from helix to sheet of A-beta-42 peptide. *109*, 108027.

(18) Tian, H.; Trozzi, F.; Zoltowski, B. D.; Tao, P. Deciphering the Allosteric Process of the Phaeodactylum tricornutum Aureochrome 1a LOV Domain. *124*, 8960–8972, Publisher: American Chemical Society.

(19) Bowman, G. R.; Bolin, E. R.; Hart, K. M.; Maguire, B. C.; Marqusee, S. Discovery of multiple hidden allosteric sites by combining Markov state models and experiments. *Proceedings of the National Academy of Sciences* **2015**, *112*, 2734–2739.

(20) Pontiggia, F.; Pachov, D.; Clarkson, M.; Villali, J.; Hagan, M.; Pande, V.; Kern, D. Free energy landscape of activation in a signalling protein at atomic resolution. *Nature Communications* **2015**, *6*, 7284.

(21) Noé, F.; Rosta, E. Markov Models of Molecular Kinetics. *MMMK*, 190401.

(22) Trendelkamp-Schroer, B.; Wu, H.; Paul, F.; Noé, F. Estimation and uncertainty of reversible Markov models. *143*, 174101.

(23) Feurer, M.; Hutter, F. *Automated machine learning*; Springer, Cham, pp 3–33.

(24) Husic, B. E.; McGibbon, R. T.; Sultan, M. M.; Pande, V. S. Optimized parameter selection reveals trends in Markov state models for protein folding. *145*, 194103.

(25) Scherer, M. K.; Husic, B. E.; Hoffmann, M.; Paul, F.; Wu, H.; Noé, F. Variational selection of features for molecular kinetics. *150*, 194108, Publisher: American Institute of Physics.

(26) Husic, B. E.; Pande, V. S. Markov State Models: From an Art to a Science. *140*, 2386–2396.

(27) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *139*, 015102, Publisher: American Institute of Physics.

(28) Bergstra JAMESBERGSTRA, J.; Yoshua Bengio YOSHUABENGIO, U. Random Search for Hyper-Parameter Optimization. *13*, 281–305.

(29) Bergstra, J.; Yamins, D.; Cox, D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Proceedings of the 30th International Conference on Machine Learning. pp 115–123, Issue: 1.

(30) Bergstra, J. S.; Bardenet, R.; Bengio, Y.; Kégl, B. In *Advances in neural information processing systems 24*; Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., Weinberger, K. Q., Eds.; Curran Associates, Inc., pp 2546–2554.

(31) C Montgomery, D. *Montgomery design and analysis of experiments*; John Wiley.

(32) Simon, D. *Evolutionary optimization algorithms*; John Wiley & Sons.

(33) Kennedy, J.; Eberhart, R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks. pp 1942–1948.

(34) Eberhart, R. C.; Shi, Y. Comparison between genetic algorithms and particle swarm optimization. Evolutionary programming VII. pp 611–616.

(35) Hansen, N. The CMA evolution strategy: A tutorial.

(36) Hutter, F.; Hoos, H. H.; Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. Learning and intelligent optimization. pp 507–523.

(37) Snoek, J.; Larochelle, H.; Adams, R. P. In *Advances in neural information processing systems 25*; Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Curran Associates, Inc., pp 2951–2959.

(38) Bergstra, J.; Yamins, D.; Cox, D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Proceedings of the 30th International Conference on Machine Learning. pp 115–123.

(39) McGibbon, R. T.; Pande, V. S. Variational cross-validation of slow dynamical modes in molecular kinetics. *142*.

(40) Wu, H.; Noé, F. Variational Approach for Learning Markov Processes from Time Series Data. *30*, 23–66.

(41) Husic, B. E.; McKiernan, K. A.; Wayment-Steele, H. K.; Sultan, M. M.; Pande, V. S. A Minimum Variance Clustering Approach Produces Robust and Interpretable Coarse-Grained Models. *Journal of Chemical Theory and Computation* **2018**, *14*, 1071–1082.

(42) Wan, H.; Voelz, V. A. Adaptive Markov State Model Estimation Using Short Reseeding Trajectories. *The Journal of Chemical Physics* **2020**, *152*, 024103.

(43) Sidky, H.; Chen, W.; Ferguson, A. L. High-Resolution Markov State Models for the Dynamics of Trp-Cage Miniprotein Constructed Over Slow Folding Modes Identified by State-Free Reversible VAMPnets. *The Journal of Physical Chemistry B* **2019**, *123*, 7999–8009.

(44) Mittal, S.; Shukla, D. Maximizing Kinetic Information Gain of Markov State Models for Optimal Design of Spectroscopy Experiments. *The Journal of Physical Chemistry B* **2018**, *122*, 10793–10805.

(45) McKiernan, K. A.; Husic, B. E.; Pande, V. S. Modeling the Mechanism of CLN025 Beta-Hairpin Formation. *The Journal of Chemical Physics* **2017**, *147*, 104107.

(46) Hruska, E.; Abella, J. R.; Nüske, F.; Kavraki, L. E.; Clementi, C. Quantitative Comparison of Adaptive Sampling Methods for Protein Dynamics. *The Journal of Chemical Physics* **2018**, *149*, 244119.

(47) Mardt, A.; Pasquali, L.; Wu, H.; Noé, F. VAMPnets for Deep Learning of Molecular Kinetics. *Nature Communications* **2018**, *9*, 5.

(48) Hernández, C. X.; Wayment-Steele, H. K.; Sultan, M. M.; Husic, B. E.; Pande, V. S. Variational Encoding of Complex Dynamics. *Physical Review E* **2018**, *97*, 062412.

(49) Husic, B. E.; Pande, V. S. Ward Clustering Improves Cross-Validated Markov State Models of Protein Folding. *Journal of Chemical Theory and Computation* **2017**, *13*, 963–967.

(50) Prinz, J. H.; Wu, H.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schtte, C.; Noé, F. Markov models of molecular kinetics: Generation and validation. *134*.

(51) Noé, F.; Clementi, C. Kinetic Distance and Kinetic Maps from Molecular Dynamics Simulation. *11*, 5002–5011.

(52) Friedman, J.; Hastie, T.; Tibshirani, R. *The elements of statistical learning*, second edition ed.; Springer-Verlag New York.

(53) Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *4*, 40–79.

(54) Efron, B.; Tibshirani, R. J. *An Introduction to the Bootstrap*; Springer US.

(55) Ozaki, Y.; Tanigaki, Y.; Watanabe, S.; Nomura, M.; Onishi, M. Multiobjective Tree-Structured Parzen Estimator. *Journal of Artificial Intelligence Research* **2022**, *73*, 1209–1250.

(56) Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; Shaw, D. E. How Fast-Folding Proteins Fold. *334*, 517–520.

(57) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pérez-Hernández, G.; Hoffmann, M.; Plattner, N.; Wehmeyer, C.; Prinz, J.-H.; Noé, F. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *11*, 5525–5542.

(58) Suárez, E.; Wiewiora, R. P.; Wehmeyer, C.; Noé, F.; Chodera, J. D.; Zuckerman, D. M. What Markov State Models Can and Cannot Do: Correlation versus Path-Based Observables in Protein-Folding Models. *Journal of Chemical Theory and Computation* **2021**, *17*, 3119–3133.

(59) Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery I& Data Mining. pp 2623–2631.