Blind analysis of molecular dynamics.

Sergei Krivov ∗
Astbury Center for Structural Molecular Biology, University of Leeds, Leeds, United Kingdom
(Dated: May 7, 2020)

We describe a non-parametric approach for accurate determination of the slowest relaxation eigenvectors of molecular dynamics. The approach is blind as it uses no system specific information. In particular, it does not require a functional form with many parameters to closely approximate eigenvectors, e.g., a neural network, and thus no extensive expertise with the system. The power of the approach is illustrated on long atomistic protein folding trajectories. The determined eigenvectors pass a stringent validation test at timescale of 0.2 ns, much shorter than alternative approaches.

Introduction. Molecular dynamics simulations increasingly produce massive trajectories [1, 2]. Accurate analysis and interpretation of such data are widely recognized as fundamental bottlenecks that could limit their applications, especially in the forthcoming era of exascale computing [3–7]. A rigorous way to analyze dynamics in such data is to describe/approximate it by diffusion on the free energy landscape, free energy as a function of reaction coordinates (RCs). For such a description to be quantitatively accurate, the RCs should be chosen in an optimal way [5, 8]. The committor function is an example of such RCs, that can be used to compute some important properties of the dynamics exactly [8]. The eigenvectors (EVs) of the transfer operator are another example [9]. They are often used to decrease the dimensionality of the dynamics during the construction of Markov state models (MSM) [10, 11]. Incidentally, one embarrassingly parallel strategy to exascale simulations consists of running a very large number of short trajectories independently, which are later combined using MSMs in order to obtain a long time behavior [7, 12].

The minimal lag time when a MSM becomes approximately Markovianis a good indicator of the accuracy of the constructed model. State of the art approaches have lag times in the range of tens of nanoseconds [10, 11, 13, 14]. Shorter lag times mean more accurate putative EVs and MSMs, as well as shorter trajectories and higher efficiency for the simple strategy of exascale simulations. Here we present an approach, which determines EVs for protein folding trajectories, which pass a stringent EV validation test at much shorter lag time of trajectory sampling interval of 0.2 ns.

In contrast to alternative approaches, which require a functional form with many parameters to approximate EVs, e.g., linear combinations of molecular descriptors/features [10, 11] or deep neural networks [13, 14], and thus extensive expertise with the system, the suggested approach is non-parametric and can approximate any EV with high accuracy without system specific information. Instead of optimizing the parameters of the approximating function, the approach directly optimizes EVs time-series.

The paper is as follows. First, a non-parametric approach to perform EV optimization while avoiding spurious EVs is introduced and illustrated on a long equilibrium protein folding trajectory. Next, the EV optimality criterion is introduced. It shows that some regions of the putative EV are suboptimal (underfitted), while other are overfitted. The criterion is adopted to perform optimization in a more uniform way. We conclude by discussing the obtained protein folding free energy landscapes.

Non-parametric optimization of eigenvectors. Here we summarize the main ideas of the approach, more details are given in Supplemental Material (SM) [15]. Imagine a hypothetical case of an (infinitely) long trajectory $X(i\Delta t_0)$ that has been clustered in a very fine-grained manner, here $\Delta t_0$ is the trajectory sampling interval. By counting transitions between different clusters, one can compute the transition probability matrix and its left EVs. The dimensionality and complexity of dynamics can be reduced by projecting it on a few EVs describing slowest relaxations, i.e., by computing EVs values as a function of trajectory snapshot or time $u_k(i\Delta t_0)$. In reality, the trajectories are relatively short and accurate fine-grained clustering is impossible due to the curse-of-dimensionality. The proposed approach estimates $u_k(i\Delta t_0)$ without clustering and for trajectories of realistic length. It does so by iteratively improving $u_k(i\Delta t_0)$, using different collective variables time-series $y(i\Delta t_0)$ computed from the trajectory. The exact choice of the family of collective variables, provided it contains all important information about the dynamics, is not important. Here we used distances between randomly chosen pairs of atoms; using backbone dihedral angles instead, for example, leads to the same results.

The EVs time-series are iteratively improved as

$$r(i\Delta t_0) = f(u_k(i\Delta t_0), y(i\Delta t_0)) + \sum_{j\neq k} \alpha_j u_j(i\Delta t_0),$$

$k$ denotes an active EV which contributes via low degree polynomial $f(x, y)$, while other EVs contribute linearly. The optimal values of coefficients of $f(x, y)$ and $\alpha_j$ are obtained by solving a constrained optimization problem: $\min \sum_i [r(i\Delta t + \Delta t) - r(i\Delta t)]^2$ under constraint $\sum_j r^2(i\Delta t_0) = 1$. It leads to the generalized eigenvalue problem, which has a few solutions describing different EVs. $k$-th solution is used to update $k$-th EV time-series.
The problem of determining slowest EVs has the following "instability" [16]. The algorithm seeks EVs with the smallest eigenvalues, which describe the slowest dynamics. However, some of such EVs, which we denote as spurious EVs, are not of interest. For example, in protein folding, such an EV could describe a much slower torsion angle isomerization process [9, 16]. Another, more frequent possibility is due to limited sampling. There are many parts of the configuration space that were visited only once, and EVs describing those transitions have small eigenvalues. Thus, starting with an EV of interest, the algorithm may eventually converge to a spurious EV, with smaller eigenvalue, but of no interest.

Shift to a spurious EV happens usually in an abrupt manner and results in significant changes in the EV time-series. Thus, to avoid spurious EVs the algorithm allows only gradual changes of the putative EV. A main idea is to keep a fraction of trajectory points, selected with probability $p_{\text{fix}}$, fixed during each iteration. It penalizes large changes in the EV time-series, since during optimization the distance between consecutive points is minimized. SM [15] describes other simple checks to avoid spurious EVs.

In order to optimize all the timescales uniformly, the optimization starts with a large lag time and iteratively halves it upon convergence, e.g., when eigenvalues estimated with current and much larger lag times are close.

Fig. 1 shows application of the approach to a long equilibrium trajectory of double mutant of HP35 protein consisting of 1509392 snapshots at 380 K [17]. Fig. 1a shows free energy profile (FEP) as a function of the first EV $F(u_1)$. It has a simple shape of one free energy (FE) barrier and two minima. Fig. 1b shows that an accurate estimate of implied timescales is possible with a lag time of the trajectory sampling interval of $\Delta t_0 = 0.2$ ns.

However, eigenvalues and, correspondingly, implied timescales provide a rather crude, cumulative estimate of the accuracy of putative EVs. It is possible that while an eigenvalue is accurate, some parts of EV are overfitted/overoptimized, while other underfitted. To check for that, we apply a more stringent EV optimality/validation criterion $\Theta(x, \Delta t)$ (Fig. 1c), which is a generalization of the criterion for the committor function [8, 25]. Its derivation is provided in [15], here we summarize its properties. Validation: if putative time-series $u(i\Delta t_0)$ closely approximates an EV then $\Theta(x, \Delta t) \approx 0$ for all $\Delta t$ and all $x$ along $u$. Optimality: for a suboptimal $u$, $\Theta(x, \Delta t)$ is generally negative and increases towards zero as $\Delta t$ increases. The bigger the difference between $\Theta(x, \Delta t_1)$ and $\Theta(x, \Delta t_2)$ for $t_1 > t_2$ the less optimal is $u$ around $x$.

Fig. 1c shows that $\Theta(x, \Delta t_0) > \Theta(x, \Delta t)$ around the barrier and $\Theta(x, \Delta t_0) < \Theta(x, \Delta t)$ around minima for large $\Delta t > \Delta t_0$. It means that the putative $u_1$ time-series does not approximate the EV uniformly. It overfits the EV around the barrier region and underfits around the minima.

Thus, our aim is to determine such an EV time-series $u(i\Delta t_0)$ that it passes the validation test, i.e., $\Theta(u, \Delta t) \approx 0$ up to statistical uncertainty. A way to do this is to perform optimization more uniformly, so that all regions of the putative EVs become underfitted to the same degree and stop optimization just before overfitting. Such an adaptive optimization is performed by focusing on less optimized parts of putative EVs. Before every iteration one scans $\Theta(x, \Delta t)$ profiles to find most suboptimal/underfitted regions. Position dependent $p_{\text{fix}}(x)$ is introduced in such a way as to be smaller for more underfitted regions. Smaller $p_{\text{fix}}(x)$ means less constraints and thus faster optimization. The obtained results are robust with respect to specific form of $p_{\text{fix}}(x)$ employed. More details are given in [15].

Fig. 2 shows application of the adaptive approach to determine the first two EVs for the HP35 trajectory. Fig. 2a shows that $\Theta(x, \Delta t)$ is much closer to zero (bounded by $\pm 0.2$) compared to Fig. 1c, indicating that $u_1$ is now better approximates the EV. The FEP $F(u_1)$ also shows more structure in the minima. This additional structure disappeared on Fig. 1a because the regions were not suffi-
We propose to call such approaches blind, in analogy with the blind source separation approaches. How accurately do the FEPs on Fig. 2 describe the kinetics? For example, the FEP along the committor can be used to compute exactly such important properties of kinetics as the equilibrium flux, the mean first passage times, and the mean transition path times between any two regions on the committor [8]. It, thus, can be used to obtain direct accurate estimates of, e.g., free energy barriers and pre-exponential factors [8]. The accuracy is limited only by the accuracy of the determined committor. An EV, while being different, could be quite close to the committor between the boundary minima, especially around the transition state (TS) region [19]. It can be used to compute the properties approximately. The relative error could be roughly estimated by applying the committor validation test $Z_{C,1}$ to the EV time-series and for the first EV the error is around 30%.

FIG. 2. Adaptive non-parametric optimization of first two eigenvectors $u_1$ and $u_2$ of HP35: a) free energy (black) and optimality criterion (red) as functions of $u_1$; b) those as as functions of $u_2$; c) implied timescale $\tau$ for $u_1$ (black) and $u_2$ (red) as functions of lag time $\Delta t$, uncertainties (shaded areas) were computed with bootstrap.

FIG. 3. Free energy landscapes of HP35 double mutant: a) $F(\tilde{u}_1)$, c) $F(\tilde{u}_2)$, and b) $F(\tilde{u}_1, \tilde{u}_2)$. d) shows representative structures for the rectangular regions around the free energy minima on b); colors code the root-mean-square (rms) fluctuations of atomic positions around the average structure from 0.5 (blue) to 13 (red).

Folding landscapes and dynamics. Using $F(u_i)$ (Fig. 2) for the analysis and description of the dynamics is not very convenient as the diffusion coefficient varies significantly along the coordinates [15]. It is more convenient to use a “natural” coordinate, which we denote as $\tilde{u}_i$, where the diffusion coefficient is constant $D(\tilde{u}_i) = 1$. It is related to the committor by the following monotonous transformation $d\tilde{u}_i/du_i = D(u_i)^{-1/2}[20]$. The FEP along the first EV $F(\tilde{u}_1)$ (Fig. 3a) is in agreement with $F(\tilde{q})$, the FEP along the optimal folding coordinate - the committor between the denatured and the native states [8]. $F(\tilde{u}_1)$ and $F(\tilde{q})$, in particular, both efficiently optimized. The second EV similarly has $\Theta(x, \Delta t)$ close to zero (Fig. 2b). The implied timescales are accurate starting from the shortest lag time of 0.2 ns (Fig. 2c).

The idea of iterative improvement of EVs or committors [8] is somewhat similar to the tree boosting idea in machine learning (ML), the basis of the very successful XGBoost algorithm [18], with a difference that here we employ small degree polynomials instead of regression trees. A more detailed comparison is provided in SM [15]. The treatment of overfitting is another important difference. XGBoost uses standard ML tools, e.g., regularization, cross-validation. Our limited experiments showed that they do not improve the performance of the algorithm. Since the nonuniform approximation during learning of a complex function is likely to be a general phenomenon, the usage of a criterion such as suggested here could improve the performance of ML algorithms.

As one can see the proposed approach uses no system specific information and is formulated in generic, system agnostic terms, e.g., eigenvectors and eigenvalues, committors [8], optimality criteria, free energy landscapes.
have minima 3, 4 and 5 and the folding barrier of $\sim 3.5$ kT, confirming that the approach works. There are however also important differences: $F(\tilde{q})$ does not show minima 1 and 2 and minima 4 and 5 are in the opposite order. This is due to the employed definition of the boundary denatured and native states for the committer, defined as structures that have the $C_\alpha$ root-mean-squared-deviation (rmsd) from the 2f4k pdb structure smaller than 0.5 Å and greater than 10.5 Å, respectively. Using the native minimum (4) with the smallest rmsd as the boundary state forces it to be the rightmost minimum on $F(\tilde{q})$, while $F(\tilde{u}_1)$ reveals that kinetically 5 is the rightmost minimum. Minima 1, 2 and 3 all have very similar projections on the rmsd, and the boundary state with large rmsd is equally connected to all of them, preventing their separation along $\tilde{q}$. This illustrates that proper definition of boundary states for committer is a difficult problem. As even such a natural approach as using the rmsd leads to inaccuracies. The problem is likely to be more severe for more complex cases, e.g., intrinsically disordered proteins, allosteric transitions, etc, which could be treated by the proposed approach.

Once constructed, the landscapes (Fig. 3) can be post-processed to obtain descriptions of minima, TSs, pathways in terms of easy-interpretable coordinates, e.g., dihedral angles, distances [21], or secondary and tertiary structures. Here we do the latter (Fig. 2d). In minimum 3 the protein is almost folded: all three helices are formed with a relatively high propensity and are all at the right positions. The hydrophobic core is not formed and the structure is rather flexible. In the native minimum (4) the folding is completed by forming the hydrophobic core and making the structure stable. Near-native minimum 5 has first and third helices partially unraveled [22]. In minimum 3 residues 18-24 form a turn, connecting second and third helices, whereas in minima 1 and 2, they form a helix with $> 90\%$ propensity. It leads to the possibility of the second and third helices forming a single long helix in minimum 2 and a longer second helix in 1.

The two-dimensional FES $F(\tilde{u}_1, \tilde{u}_2)$ can be used to find the correspondence among the minima on the FEPs and see the evidence of parallel pathways. In particular, the FES for HP35 has an L-like shape and shows no evidence of parallel pathways. The global descriptors of the dynamics, like free energies of TSs and minima, are easier to understand from one-dimensional FEPs, i.e., $F(\tilde{u}_1)$.

We have also applied the approach to the trajectory of FIP35 protein (Fig. 4) [1]. The EV validation test $\Theta(x, \Delta t)$ was bounded by $\pm 0.2$ for both EVs. This trajectory has only 15 folding-unfolding events, which illustrates that the approach can analyze systems with very limited sampling. $F(\tilde{u}_1)$ shows two minima with an intermediate state in agreement with other studies [23, 24]. The two-dimensional FES has an A-like shape and shows the evidence of two parallel pathways, i.e., protein folds from 1 to 4 via 2 or 3. The representative structure of 2 has the first hairpin formed, while that of 3 has the second hairpin formed to a much larger degree. Surprisingly, region 3 is a TS rather then an intermediate state. It might explain why this pathway is much less populated. It might be difficult to detect the pathway using MSMs, as they are not good at identifying TSs.

* s.krivov@leeds.ac.uk

**FIG. 4.** Free energy landscapes of FIP35: notation as in Fig. 3.

In conclusion, we have described a blind approach for the determination of the slowest relaxation eigenvectors from an equilibrium trajectory. The approach determined the first and second slowest eigenvectors for the HP35 and FIP35 proteins with high spatio-temporal accuracy, as validated by a stringent criterion at the shortest lag time of 0.2 ns. In contrast to alternative (parametric) approaches, which require approximating functions with many parameters and extensive expertise with the system, the approach directly determines the eigenvector time-series and uses no system specific information. The optimality criterion is another important ingredient of the approach, which makes the uniform optimization possible. The approach can be used in cases when one does not want to introduce any bias in the analysis, e.g., due to employed approximating functions, or one does not have good approximating functions. It can also be used aposteriory to check if possible bias in the analysis has altered the results. As the HP35 example illustrates, even a seemingly innocent and natural choice of boundary states can hide the inherent complexity of the landscape.

I am grateful to David Shaw and his coworkers for making the folding trajectories available.
[1] D. E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R. O. Dror, M. P. Eastwood, J. A. Bank, J. M. Jumper, J. K. Salmon, Y. Shan, and W. Wriggers, Science \textbf{330}, 341 (2010).
[2] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw, Science \textbf{334}, 517 (2011).
[3] P. L. Freddolino, C. B. Harrison, Y. Liu, and K. Schulten, Nat Phys \textbf{6}, 751 (2010).
[4] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. \textbf{11}, 600 (2015).
[5] P. V. Banushkina and S. V. Krivov, WIREs Comput Mol Sci \textbf{6}, 748 (2016).
[6] F. Noé and C. Clementi, Curr. Opin. Struct. Biol. \textbf{43}, 141 (2017).
[7] H. Jung, R. Covino, and G. Hummer, (2019), arXiv:1901.04595.
[8] S. V. Krivov, J. Chem. Theory Comput. \textbf{14}, 3418 (2018).
[9] R. T. McGibbon, B. E. Husic, and V. S. Pande, J Chem Phys \textbf{146}, 044109 (2017).
[10] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. \textbf{9}, 2000 (2013).
[11] G. Prez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. No, J. Chem. Phys. \textbf{139}, 015102 (2013).
[12] H. Wan and V. A. Voelz, J. Chem. Phys. \textbf{152}, 024103 (2020).
[13] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, Phys. Rev. E \textbf{97}, 062412 (2018).
[14] A. Mardt, L. Pasquali, H. Wu, and F. No, Nature Communications \textbf{9}, 5 (2018).
[15] See Supplemental Material at http://... for a detailed description of the approach, its comparison with the tree boosting and derivation of the optimality criterion.
[16] P. V. Banushkina and S. V. Krivov, J. Chem. Phys. \textbf{143}, 184108 (2015).
[17] S. Piana, K. Lindorff-Larsen, and D. E. Shaw, PNAS \textbf{109}, 17845 (2012).
[18] T. Chen and C. Guestrin, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 16 (Association for Computing Machinery, New York, NY, USA, 2016) p. 785794.
[19] A. Berezhkovskii and A. Szabo, J. Chem. Phys. \textbf{121}, 9186 (2004).
[20] S. V. Krivov and M. Karplus, PNAS \textbf{105}, 13841 (2008).
[21] S. Brandt, F. Sittel, M. Ernst, and G. Stock, J. Phys. Chem. Lett. \textbf{9}, 2144 (2018).
[22] K. A. Beauchamp, R. McGibbon, Y.-S. Lin, and V. S. Pande, PNAS \textbf{109}, 17807 (2012).
[23] S. V. Krivov, J. Phys. Chem. B \textbf{115}, 12315 (2011).
[24] L. Boninsegna, G. Gobbo, F. Noe, and C. Clementi, J. Chem. Theory Comput. \textbf{11}, 5947 (2015).
[25] S. V. Krivov, J. Chem. Theory Comput. \textbf{9}, 135 (2013).
[26] A. M. Berezhkovskii and A. Szabo, J. Phys. Chem. B \textbf{117}, 13115 (2013).
[27] J. Lu and E. Vanden-Eijnden, J. Chem. Phys. \textbf{141}, 044109 (2014).
Supplemental Material for: Blind analysis of molecular dynamics.

In this Supplemental Material we present technical details related to the proposed non-parametric approach of eigenvector optimization, the comparison of the approach with the tree boosting, namely, XGBoost approach, and derivation of the eigenvector optimality criterion.

Variational optimization of eigenvectors

Assume that the system dynamics is described approximated by a Markov chain with transition probability matrix $P(i|j, \Delta t)$ for transition from state $j$ to state $i$ after time interval $\Delta t$. Note that this assumption is used only for the derivation of equations. One does not need to know the actual Markov chain, meaning that this assumption does not restrict the applicability of the algorithm.

Given an infinitely long equilibrium trajectory $X(k \Delta t_0)$, where $\Delta t_0$ is the trajectory sampling interval, and using a very fine-grained clustering of the configuration space of the system, one can, in principle, estimate the transition matrix $P(i|j, \Delta t_0) = n(i|j, \Delta t_0)/n(j)$, where $n(i|j, \Delta t_0)$ is the number of transitions from cluster $j$ to cluster $i$ observed in the trajectory and $n(j) = \sum_i n(i|j, \Delta t_0)$. Knowing $P(i|j, \Delta t_0)$ one can estimate the left eigenvectors

$$\sum_i u(i)P(i|j, \Delta t_0) = e^{-\mu \Delta t_0}u(j). \quad \text{(S1)}$$

Knowing the cluster index $i$ as a function of (trajectory) time $i(k \Delta t_0)$ one can compute eigenvector value as a function of time $u(k \Delta t_0)$ or project the trajectory on the eigenvector.

In practice very long trajectories are rarely available, which makes this approach with accurate very fine-grained clustering non viable. Variational approaches are promising alternative to the clustering approach [S1–S3]. A functional form (FF) with many parameters $R(X, \alpha_i)$ (usually a weighted sum) is suggested as an approximation to EVs. One numerically optimizes the parameters $\alpha_i$ by e.g., maximizing the auto-correlation function or minimizing the total squared displacement [S4].

Namely, given a long equilibrium multidimensional trajectory $X(k \Delta t_0)$, where $\Delta t_0$ is the trajectory sampling interval, one computes the reaction coordinate time-series $r(k \Delta t_0) = R(X(k \Delta t_0), \alpha_i)$. Here and below $r$ denotes any reaction coordinate, while $u$ is reserved for putative eigenvectors. The functional form $R$ approximates the first left eigenvector, if it provides the minimum to the total squared displacement $\Delta r^2(\Delta t_0) = \sum_k [r(k \Delta t_0 + \Delta t_0) - r(k \Delta t_0)]^2$, under the constraint $\sum_k r(k \Delta t_0)^2 = 1$. Note that, due to the constraint, the minimization of $\Delta r^2$ is equivalent to the maximization of the auto-correlation function $C(r, \Delta t_0) = \sum_k r(k \Delta t_0 + \Delta t_0)r(k \Delta t_0)$. The functional form $R$ approximates the $i$-th left eigenvector if it provides the minimum to the $\Delta r^2$ under constraint $\sum_k r(k \Delta t_0)^2 = 1$ and is orthogonal to the previous $i-1$ eigenvectors $\sum_k r(k \Delta t_0)u_j(k \Delta t_0) = 0$, $j = 1, ..., i-1$.

It is straightforward to prove this principle. Consider the Markov chain, describing the dynamics. Let indexes $i$ and $j$ denote the states of the chain and $r(i)$ their position on an RC. The total squared displacement equals $\Delta r^2(\Delta t) = N\sum_{ij}[r(i) - r(j)]^2P(i|j, \Delta t_0)P(j)$, while the constraint is $N\sum_r r^2(j)P(j) = 1$, where $N$ is the total number of snapshots in the trajectory. Using $2\lambda$ as the Lagrange multiplier, differentiating with respect to $r(j)$ and assuming the detailed balance one obtains Eq. S1 with $\lambda = e^{-\mu \Delta t}$.

Estimation of implied timescales

The minimal value of the functional, attained when $r$ approximates an eigenvector, equals $\Delta u^2(\Delta t) = 2(1 - e^{-\mu \Delta t})$, which, for small $\Delta t$, gives $\Delta u^2(\Delta t) \approx 2\mu \Delta t$. Correspondingly, the maximum value of the autocorrelation term equals $C(u, \Delta t) = e^{-\mu \Delta t}$. They can be used to estimate the eigenvalues $\mu$, or the so called implied timescales $\hat{\tau} = 1/\mu$ as

$$\hat{\tau}_i = -\Delta t/\ln[1 - \Delta u^2(\Delta t)/2] \quad \text{(S2a)}$$

$$\hat{\tau}_i = -\Delta t/\ln[C(u_i, \Delta t)] \quad \text{(S2b)}$$

as functions of lag time $\Delta t$. Large lag times mask suboptimality of the putative eigenvector and lead to a more accurate estimate of $\tau$. However at very large lag times it becomes difficult to accurately estimate an exponentially decreasing value of $C(u_i, \Delta t) = e^{-\mu \Delta t}$, since its statistical accuracy is limited by the number of transitions between different free energy minima. An accurate and robust estimate should have statistical errors much smaller than the estimated value. A characteristic lag time $\Delta t^*$, where the two are comparable could be roughly estimated as $(\mu T)^{-1/2} = e^{-\mu \Delta t^*}$, where $T$ is the total duration of the trajectory. We denote $\Delta t_{inf}$ as the lag time chosen to accurately estimate the eigenvalues and the implied timescales. For the HP35 protein considered here $\Delta t^* \sim 10^4 \Delta t_0$ and we took $\Delta t_{inf} = 1024 \Delta t_0$.

Non-parametric optimization of eigenvectors

A major weakness of the variational approaches is that it is difficult to suggest a good FF approximating EVs. The difficulty becomes apparent if one remembers that such a FF should be able to accurately project a few
million snapshots of a very high-dimensional trajectory. In particular, it implies an extensive knowledge of the system, and that such a FF is likely to be system specific.

Recently we have suggested a non-parametric approach for the determination of the committor function, which bypasses the difficult problem of finding an appropriate FF [S4, S5]. Since \( \Delta r^2 \) depends explicitly only on the RC time-series \( r(k\Delta t_0) \), one may directly optimize the values of \( r(k\Delta t_0) \) rather than the parameters \( \alpha_i \) of the FF \( R(X, \alpha_i) \).

The power of the approach was demonstrated by applying it to the equilibrium folding trajectory of the HP35 double mutant. The determined RC closely approximates the committor as was validated by the optimality criterion \( Z_{C,1} \) is constant up to the expected statistical noise. The approach performs optimization of the RC in a uniform manner by focusing optimization on the time scales and the regions of the putative RC which are most suboptimal.

Here we extend the approach to non-parametric determination of eigenvectors. Namely, instead of determining the function approximating an eigenvector \( R(X, \alpha_j) \) we determine the values of the function for the snapshots of the trajectories \( u(i\Delta t_0) = R(X(i\Delta t_0), \alpha_j) \). More precisely, given a multidimensional trajectory \( X(i\Delta t_0) \) and the number of the slowest eigenvectors required \( n_{ev} \), the approach determines time-series of the required eigenvectors \( u_k(i\Delta t_0) \) and corresponding eigenvalues \( \mu_k \), where \( i = [1, N] \) and \( N \) is the trajectory length and \( k = [1, n_{ev}] \).

The eigenvectors time-series are improved by the following iterations

\[
r(i\Delta t_0) = f(u_k(i\Delta t_0), y(i\Delta t_0)) + \sum_{j\neq k} \alpha_j u_j(i\Delta t_0),
\]

where \( u_k(i\Delta t_0) \) are the time-series of eigenvectors, \( y(i\Delta t_0) \) is the time-series of a randomly chosen collective variable of the original multidimensional space \( X \), and \( f(x, y) = \sum \alpha_j x^i y^j \) is a low degree polynomial. \( k \) denotes the index of an active eigenvector, whose contribution to the time-series is higher then linear.

The optimal values of the parameters \( \alpha_j \) and coefficients of the polynomial \( a_{ij} \) are determined by numerically solving the constrained optimization problem: \( \min \Delta r^2(\Delta t) \) or max \( C(r, \Delta t) \) under constraint \( \sum_k r^2(k\Delta t_0) \), \( \Delta t \) = 1. The problem has a few solutions describing different EVs. Every EV time-series is updated using the corresponding solution: \( k \)-th EV is updated using \( k \)-th solution. Inclusion of the linear combination of all eigenvectors in Eq. S3 insures that each solution has the corresponding eigenvector as a baseline and that the updated eigenvectors are orthogonal. Active eigenvectors can be selected randomly or, one may select the least optimal eigenvector, i.e., the one having the largest ratio \( \mu(\Delta t)/\mu(\Delta t_{inf}) \).

The EV corresponding to \( \mu = 0 \), the equilibrium EV, which we denote as \( u_0 \) equals to a constant. It is not considered explicitly in the sum in Eq. S3, as the polynomial contains a constant term already.

If the system obeys some symmetry (e.g., the rotational and translational symmetries for biomolecules), then the eigenvectors should obey the same symmetry. A simple way to ensure this is to use as \( y(i\Delta t) \) collective variables that respect the symmetry. For analysis of protein dynamics, one can use distance time series between randomly chosen pairs of atoms.

The problem of determining slowest EVs has the following inherent "instability" [S4]. The algorithm seeks EVs with the smallest eigenvalues, which describe the slowest dynamics. However, some of such EVs, which we denote as spurious EVs, are not of interest. For example, in protein folding, such an EV could describe a much slower torsion angle isomerization process [S4, S6]. Another, more frequent possibility, is due to limited sampling. There are many parts of the configuration space that were visited only once, and EVs describing those transitions have small eigenvalues. Thus, starting with an EV of interest, the algorithm may eventually converge to a spurious EV, with smaller eigenvalue, but of no interest. In general terms, this peculiarity of EV optimization is due to its unsupervised nature: we seek any EV with smallest eigenvalue. Optimization of the committor function, which is a variant of supervised learning, as the function interpolates between two given boundary states of interest, is free of such a problem [S4, S5].

To keep the optimization from converging to spurious eigenvectors we employ the following empirical fact. Change to spurious eigenvectors happens abruptly. One of the eigenvectors and its corresponding eigenvalue change significantly. The idea is to enforce only gradual changes. First, one may want to penalize large changes in eigenvector times series, e.g., by adding to optimization functional the squared difference \( \sum_i r^2(i\Delta t_0) - u_k(i\Delta t_0)]^2 \). An equivalent, while computationally more efficient way is to fix positions of some of the points. Allowing, an overall shift and change of scale, it means that fixed points are transformed according to Eq. S3, where polynomial is \( f(x, y) = a + bx \), i.e., all eigenvectors contribute linearly. Transformation of other points is described by a polynomial of a higher degree (fourth here). The fixed points are selected randomly before every iteration with probability \( p_{fix} \) (0.5 here). Increasing \( p_{fix} \) leads to a more gradual change of eigenvectors during optimization.

The eigenvalue of an eigenvector, estimated at large lag time \( \Delta t_{inf} \), changes rather little after an initial settling phase. Hence, a relatively large change (5% here), is an indication that an eigenvector has changed significantly. Iterations with such changes are not accepted.

Collective variable \( y \) is first transformed to the first eigenvector as its function \( y \rightarrow u_1(y) \), by using transformation Eq. S3 without eigenvectors, e.g., \( \alpha_j = 0 \) and \( f(x, y) = p(y) \) a polynomial of 10-th degree. If the corre-
sponding eigenvalue is very small ($\mu < 10^{-4}$ here), which indicates that $y$ is likely to describe a spurious EV, the CV is rejected.

In the infrequent cases, when, in spite of the above, the algorithm switches to a spurious eigenvector, the optimization is restarted. Such events are detected by the following heuristics: one monitors the amplitude of an eigenvector $A(u) = \max_i u(i\Delta t_0) - \min_i u(i\Delta t_0)$. When the amplitude reaches a relatively large value (30 here), it indicates of a spurious eigenvector.

To perform optimization uniformly over different time scales we proceed as follows. Optimization starts with a relatively large lag time $\Delta t$ (256$\Delta t_0$ here). It continues until $\mu(\Delta t)/\mu(\Delta t_{\text{init}}) > 1.25$ if $\Delta t > \Delta t_0$ or until $\mu(\Delta t)/\mu(\Delta t_{\text{init}}) > 1$ for $\Delta t = \Delta t_0$. If $\Delta t > \Delta t_0$ then $\Delta t$ is halved and optimization continued with new $\Delta t$, otherwise optimization stops.

**Eigenvector optimality criterion**

Here we introduce an eigenvector optimality criterion for the committer reaction coordinate. $Z_{C,1}$ can be straightforwardly computed from time-series $r(k\Delta t_0)$: $Z_{C,1}(x, \Delta t)$ equals half the total length the trajectory, sampled with interval $\Delta t$, makes, when it transits through point $x$ [55, 57]. Alternatively, each transition of trajectory from $x_1 = x(i\Delta t)$ to $x_2 = x(i\Delta t + \Delta t)$ adds $1/2|x_1 - x_2|$ to $Z_{C,1}(x, \Delta t)$ for all $x$ such that $\min[|x_1, x_2| < x < \max(x_1, x_2)$, i.e., all the points between $x_1$ and $x_2$.

It has a number of useful properties [55, 57]. If reaction coordinate $q$ closely approximates the committer function, then $Z_{C,1}(q, \Delta t) \approx N_{AB}$, where $N_{AB}$ is the number of transitions between boundary states $A$ and $B$. For a suboptimal reaction coordinate $r$, $Z_{C,1}(r, \Delta t)$ values generally decrease to the limiting value of $N_{AB}$, as $\Delta t$ increases. The larger the difference between $Z_{C,1}(r, \Delta t_1)$ and $Z_{C,1}(r, \Delta t_2)$ the less optimal the reaction coordinate around $r$. This property is used to find suboptimal spatio-temporal regions and focus optimization on them to make it more uniform.

The constancy of $Z_{C,1}(q, \Delta t)$ along the committer coordinate $q$ follows from the following. Consider the Markov chain, describing the dynamics. Let indexes $i$ and $j$ denote the states of the chain and $x(i)$ their position on an RC. Value of $Z_{C,1}(x, \Delta t)$ can change, in a step-wise fashion, only when position $x$ goes through a particular state $j$, i.e., $x$ goes from $x(j-1)$ to $x(j+1)$ and equals [57]

$$\Delta Z_{C,1}(x(j), \Delta t) = \sum_i [x(i) - x(j)]n(i|j, \Delta t).$$

$$\text{(S4)}$$

It is zero for the committer function (if $j$ is not a boundary state) since committer is defined by the following equation

$$\sum_i [q(i) - q(j)]P(i|j, \Delta t) = 0 \quad \text{(S5a)}$$

$$q(A) = 0, \quad q(B) = 1$$

and $n(i|j, \Delta t) = P(i|j, \Delta t)P(j)$. Eq. S1 is different from S5a which means that $Z_{C,1}$ along an eigenvector is not constant. However Eq. S1 can be rewritten as

$$\sum_i [u(i) - u(j)]n(i|j, \Delta t) = (1 - e^{-\mu\Delta t})[0 - u(j)]n(j),$$

and interpreted in the following way. On the left hand side we have change in $Z_{C,1}$ around $u(j)$ computed in the standard way. It is proportional to the change of $Z_{C,1}$ computed for a virtual trajectory consisting of collection of transitions $0$ to $u(j)$ and back to $0$ made $n(j)$ times for every $j$. We denote the second profile as $Z_{C,1}^0$. Since both profiles are $0$ at large negative $x$ and have proportional changes, they are proportional themselves $Z_{C,1}(x, \Delta t) = (1 - e^{-\mu\Delta t})Z_{C,1}^0(x)$. Consider the following variable

$$\Theta(x, \Delta t) = -\ln \frac{Z_{C,1}(x, \Delta t)}{(1 - e^{-\mu\Delta t})Z_{C,1}^0(x)}. \quad \text{(S7)}$$

If $u$ and $\tilde{u}$ closely approximate an eigenvector and corresponding eigenvalue, then $\Theta(x, \Delta t) \approx 0$ for any $x$ along $u$ and any $\Delta t$. An accurate estimate of $\tilde{u}$ is obtained from the EV time-series $u$ using the autocorrelation function at large lag times $\tilde{\mu} = -\ln C(\tilde{u}, \Delta t_{\text{inf}})/\Delta t_{\text{inf}}$.

One can give following interpretation of the result. $Z_{C,1}(x, \Delta t)$ can be interpreted as a local density of the total squared displacement $\Delta x^2(\Delta t)/2$, since $\int Z_{C,1}(x, \Delta t)dx = \Delta x^2(\Delta t)/2$. Analogously, $Z_{C,1}^0(x)$ can be considered as a local density of $\sum_x x^2(k\Delta t_0)$. The constraint optimization problem is equivalent to finding minimum of an integral of $Z_{C,1}(x, \Delta t)$ under constraint that an integral of $Z_{C,1}^0(x)$ is $1$. And when a putative coordinate closely approximates an eigenvector, the local densities are proportional. For a suboptimal coordinate $Z_{C,1}(x, \Delta t)$ is larger than that for the optimal coordinate, meaning $\Theta$ is smaller and negative. And the larger the difference between $\Theta(x, \Delta t_1)$ and $\Theta(x, \Delta t_2)$ the less optimal the EV is around $x$.

**Transformation to locally optimal time-series**

We found that keeping the putative EV locally optimal improves the performance of the approach. By locally optimal, we mean that one can not improve the putative EV time-series $r$, using only the EV time-series itself, i.e., the transformations $r \rightarrow f(r)$. The only way to improve the EV time-series further, i.e., to decrease the optimization functional, is to use information provided by a collective time series $y(i\Delta t_0)$.
If time-series is locally optimal for a particular lag time \( \Delta t \), then the optimality criterion for this lag time is satisfied \( \theta(x, \Delta t) \approx 0 \). Thus by locally reoptimizing an EV time-series one can satisfy the optimality criterion for any chosen lag time. To satisfy the criterion for all lag times simultaneously, the putative time-series should closely approximate an EV.

One way to locally optimize a putative time-series \( r \) is to iteratively apply Eq. S3 without collective variables \( (y) \) using a high degree polynomial of the active EV only \( f(x, y) = p(x) \). Another possibility is to determine the free energy \( F(r) \) and the diffusion coefficient \( D(r) \) as functions of \( r \) and use a finite-difference method to numerically find the first eigenvector \( u_1(r) \) for the diffusive dynamics.

Adaptive non-parametric optimization of eigenvectors

The simple algorithm, described above, optimizes eigenvectors in a non-uniform way analogous to the committor case. It is easier to optimize free energy barriers than minima. To perform optimization in a uniform way one needs first to detect sub-optimal regions of eigenvectors and focus optimization on them. To detect the most suboptimal regions for current lag time \( \Delta t \), we first find a longer lag time \( \Delta t_1 \), which exhibits the most nonuniformness in the distance between \( \Theta(x, \Delta t_1) \) and \( \Theta(x, \Delta t) \):

\[
\Delta t_1 = \arg \sup \left[ \max_x \Delta \Theta(x, \Delta t_1, \Delta t) - \min_x \Delta \Theta(x, \Delta t_1, \Delta t) \right]
\]

(S8)

here \( \Delta \Theta(x, \Delta t_1, \Delta t) = \Theta(x, \Delta t_1) - \Theta(x, \Delta t) \). Then, the relative degree of suboptimality of region around \( x \) is defined as

\[
s(x) = \exp[\Delta \Theta(x, \Delta t_1, \Delta t) - \max_x \Delta \Theta(x, \Delta t_1, \Delta t)]
\]

(S9)

It takes maximal value of 1 for the most suboptimal part where the difference between \( \Theta(x, \Delta t_1) \) and \( \Theta(x, \Delta t) \) is maximal. To focus optimization on such suboptimal regions we make \( p_{\text{fix}} \) position dependent, large for optimal regions and small for suboptimal regions. Consequently, the optimization is more focused on less optimized regions, because they have a smaller number of fixed points and are less constraint. For example, an extremely over-optimized region might have \( p_{\text{fix}} = 1 \), i.e., all the points fixed and thus it will not be optimized at all. Here we used

\[
p_{\text{fix}}(x) = \min[1, p_{\text{fix}} \times s(x)^{-10}]
\]

(S10)

Before every iteration, the \( p_{\text{fix}}(x) \) values are computed for active (k-th) eigenvector, and are used to select fixed points. Namely, a point at time moment \( i \Delta t \), that has eigenvector coordinate \( u_k(i \Delta t) \) is selected to be fixed with probability \( p_{\text{fix}}(u_k(i \Delta t)) \).

Transformation to natural coordinate

Using \( F(u) \) for the analysis and description of the dynamics is not very convenient as the diffusion coefficient varies significantly along the EVs. For example, using the similarity between an EV and committor function [S8] one has \( D(x) \sim e^{F(x)/kT} \) [S7, S9, S10]. It is more convenient to use a “natural” coordinate, which we define as \( \tilde{u}_i \), where the diffusion coefficient is constant \( D(\tilde{u}_i) = 1 \) and that is related to the committor by the following monotonous transformation \( d\tilde{u}_i/du_i = D(u_i)^{-1/2} \) [S11]. Since the transformation is monotonous the new coordinate is as good as the original for the description of the dynamics. \( D(\tilde{u}_i) = 1 \) in units where time is measured in timesteps of 0.2 ns. Note, that \( D(\tilde{u}_1) = \text{const} \) and \( D(\tilde{u}_2) = \text{const} \) does not mean that \( D(\tilde{u}_1, \tilde{u}_2) = \text{const} \). Thus quantitative analysis of kinetics using \( F(\tilde{u}_1, \tilde{u}_2) \) should be done with caution.

Comparison with tree boosting.

The proposed non-parametric approach for EV and committor optimization [S5] share some similarities with the tree boosting algorithms, e.g., the very successful XGBoost [S12]. Here we discuss their differences and similarities.

Both approaches are non-parametric and they both improve the approximation of a complex function in an iterative manner, by combining simple functions. The proposed approach improves the approximation by transforming the putative function via a low degree polynomial \( f \) symbolically represented as \( u' \leftarrow f(u, y) \), here \( u \) is putative function and \( y \) is a randomly chosen collective variable of the configuration space \( X \). Thus \( u \) is transformed in a nonlinear way at each iteration. The XGBoost improves the approximation by iteratively adding a regression tree \( u' \leftarrow u + f(X) \), here \( u \) is putative function and \( f \) is a regression tree as a function of configuration space \( X \) constructed in a greedy way. The greedy approach of selecting collective variables could be useful if one wants to find the best approximation to eigenvectors using a small fixed number of collective variables.

Both approaches minimize quadratic functionals. The XGBoost minimizes the squared error difference between the putative and target functions. The proposed approach for EVs is unsupervised. It minimizes the squared displacement computed with some lag time under quadratic constraint.

The approaches fight with overfitting in different ways. XGBoost uses standard machine learning tools as \( L_1 \) and \( L_2 \) regularizations, cross-validation, etc. The current approach uses the developed optimality criterion to detect the overfitted and underfitted spatio-temporal regions. The regularizations, cross-validation have generic applicability, while the optimality criterion can be applied.
only for the analysis of (Markovian) dynamics. Our limited experiments with regularization and cross-validation showed that they are much less efficient compared to the optimality criterion in optimizing the EVs without overfitting. A possible reason could be that they don’t allow for explicit determination of overfitted and underfitted regions, to focus optimization on the latter.

[S1] G. Prez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. No, J. Chem. Phys. 139, 015102 (2013).
[S2] C. R. Schwantes and V. S. Pande, J. Chem. Theory Comput. 9, 2000 (2013).
[S3] P. V. Banushkina and S. V. Krivov, WIREs Comput Mol Sci 6, 748 (2016).
[S4] P. V. Banushkina and S. V. Krivov, J. Chem. Phys. 143, 184108 (2015).
[S5] S. V. Krivov, J. Chem. Theory Comput. 14, 3418 (2018).
[S6] R. T. McGibbon, B. E. Husic, and V. S. Pande, J Chem Phys 146, 044109 (2017).
[S7] S. V. Krivov, J. Chem. Theory Comput. 9, 135 (2013).
[S8] A. Berezhkovskii and A. Szabo, J. Chem. Phys. 121, 9186 (2004).
[S9] A. M. Berezhkovskii and A. Szabo, J. Phys. Chem. B 117, 13115 (2013).
[S10] J. Lu and E. Vanden-Eijnden, J. Chem. Phys. 141, 044109 (2014).
[S11] S. V. Krivov and M. Karplus, PNAS 105, 13841 (2008).
[S12] T. Chen and C. Guestrin, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 16 (Association for Computing Machinery, New York, NY, USA, 2016) p. 785794.