Supporting Information to
Machine Learning of Biomolecular Reaction Coordinates

Simon Brandt*, Florian Sittel*, Matthias Ernst and Gerhard Stock

Biomolecular Dynamics, Institute of Physics, Albert Ludwigs University, 79104 Freiburg, Germany (contributed equally)

(Dated: 9 April 2018)

---

* e-mail:stock@physik.uni-freiburg.de
METHODS

MD data and clustering

The MD data of HP-35 were kindly provided by D.E. Shaw Research Group, who simulated the system using the Amber ff99SB*-ILDN force field and TIP3P explicit water. Here we adopt a 300 µs trajectory with a sampling rate of (200 ps)−1 at 360 K that exhibits 61 folding-unfolding transitions.

Since previous analyses showed that the system is best described by (φ, ψ) backbone dihedral angles, we perform a principal component analysis on these data using our recently developed dPCA+ method. To minimize errors in the computation of the covariance matrix and the projection of the circular coordinates onto the according eigenvectors, dPCA+ shifts the dihedrals periodically in order to form a maximal sampling gap at the periodic boundary. Based on the shape of one-dimensional projections of the free energy landscape showing several distinguishable clusters (Fig. S2), we select principal components 1-5 and 7 for further analysis.

As an alternative dimensionality reduction method for the discussion below, we also perform a time-lagged independent component analysis (TICA) using a lag time of 1 ns, which aims to maximize the timescales exhibited by the first components. To this end, the principal components obtained from dPCA+ are normalized such that the resulting covariance matrix will remain diagonal during the subsequent unitary transformation that may then diagonalize a quantity of choice, in case of TICA the symmetrized time lagged covariance matrix. Figures S2 and S3 show cumulative fluctuations, free energy profiles, autocorrelation functions and eigenvector compositions of the most important components obtained from dPCA+, TICA and machine-learning constructed essential internal coordinates. The TICA results hardly changed, when a much longer lag time (100 ns) was employed.

Employing the chosen six principal components of dPCA+ as a low-dimensional description of coordinate space, we perform density-based clustering to construct microstates representing the local minima of the free energy landscape. Using a density estimator based on a hypersphere of fixed radius, the local free energy is estimated from the number of neighboring points inside the hypersphere. Microstates are found by separating the basins at local free energy barriers. Based on a simple heuristics employing the Boltzmann distribution, the optimal hypersphere radius (here, R = 0.3) is determined from the sampled set of conformers. To reduce the state space from 76 microstates to a set of a few metastable macrostates, we subsequently apply the most-probable-path algorithm, which computes transition probabilities between microstates from the sampled trajectories and assigns all microstates to the basin of highest transition probability and lowest free energy. Using a lag time of τ = 1 ns and a required minimum metastability of Qmin = 0.76, we obtain 12 metastable conformational states (Fig. S1). As the projection on a low-dimensional space may induce spurious transitions in the vicinity of energy barriers, in a final step we identify core regions of the microstates and count transitions only if the core region of the other state is reached.

For the analysis of T4L we adopt a 50 µs MD trajectory at 300 K and a sampling rate of (100 ps)−1 simulated by Ernst et al., using the Amber ff99SB*-ILDN force field and TIP3P explicit water. Testing various types of PCA, the system was found to be best described by a contact PCA, where a contact was considered to be formed if the distance d_{ij} between the closest non-hydrogen atoms of two residues i and j is shorter than 4.5 Å. The metastable states of T4L were characterized using density-based clustering.
XGBoost: Supervised training of decision trees

XGBoost is a well-established machine learning algorithm based on the ansatz of a decision tree ensemble. XGBoost is short for “eXtreme Gradient Boosting”, as the method employs gradient boosting, i.e., it computes first and second derivatives of the loss function which is constructed from a set of trees, in order to rapidly converge to the most promising decision path for fast training. The algorithm is described in detail in the original literature\(^\text{14}\) as well as in the associated website http://xgboost.readthedocs.io. Following closely Ref.\(^\text{14}\), here we briefly describe the basic idea of the model and specify the model parameters employed in our calculations in the main text.

Consider a given data set \(\{x_i\}, x_i \in \mathbb{R}^m (i = 1, \ldots, n)\) with \(n\) data points (e.g., the frames of the MD trajectory) and \(m\) features (e.g., the coordinates of the MD trajectory). The task is to “learn” some function \(y = \Phi(x)\) which allows us to predict values \(\hat{y}_i\), given some data point \(x_i\). To this end, a tree ensemble model employs a sum of functions \(f_k\)

\[
\hat{y}_i = \Phi(x_i) = \sum_k f_k(x_i),
\]

where a tree function \(f(x) = w_q(x)\) is defined by its structure \(q : \mathbb{R}^m \rightarrow \{1, \ldots, T\}\), that maps the input values to the corresponding leaf index, with \(T\) being the number of leaves of the tree. Hence each function \(f_k\) is associated with an independent tree structure \(q\) with leaf weights \(w\).

To learn these functions, we minimize a regularized objective

\[
\mathcal{L}(\Phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k),
\]

\[
\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2,
\]

where \(l\) is a differentiable convex loss function that measures the difference between the prediction \(\hat{y}_i\) and the true value \(y_i\) (using, e.g., least squares). In the second term, \(T\) reflects the tree size while \(\|w\|^2\) corresponds to the L2-norm of the weights of the tree. Both quantities therefore describe the complexity of the model, which is penalized using parameters \(\gamma\) and \(\lambda\). Making the model strictly convex, this regularization term helps to smooth the weights and to avoid over-fitting.

Since the model built from Eq. (2) contains functions as parameters, it cannot be optimized using standard methods in Euclidean space, but needs to be trained in an additive manner. Denoting the prediction of \(\hat{y}_i\) in the \(t\)-th iteration by \(\hat{y}_i^{(t-1)}\), this means that we iteratively add functions \(f_t\) to \(\hat{y}_i^{(t-1)}\) in order to minimize the objective

\[
\mathcal{L}^{(t)} = \sum_i l(y_i, \hat{y}_i^{(t-1)} + f_i(x_i)) + \Omega(f_t),
\]

which is often referred to as “tree boosting.” Employing second-order Taylor approximation and dropping constant terms, we obtain the simplified expression

\[
\tilde{\mathcal{L}}^{(t)} = \sum_i [g_i f_i(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega(f_t),
\]

S3
where
\[ g_i = \frac{\partial l(y_i, \hat{y}^{(t-1)})}{\partial (\hat{y})^{(t-1)}} \quad \text{and} \quad h_i = \frac{\partial^2 l(y_i, \hat{y}^{(t-1)})}{\partial ((\hat{y})^{(t-1)})^2} \]

(5)
denote first- and second-order derivatives of the loss function, respectively.

Introducing \( I_j = \{ i | q(x_i) = j \} \) as the set of indices of data points assigned to the \( j \)-th leaf and inserting the definition of \( \Omega \), the objective in the \( t \)-th iteration can be expressed as
\[
\hat{L}^{(t)} = \sum_{j=1}^{T} \left[ w_j \sum_{i \in I_j} g_i + \frac{1}{2} w_j^2 \sum_{i \in I_j} (h_i + \lambda) \right] + \gamma T. 
\]

(6)

For a fixed structure \( q(x) \), this leads to explicit expressions for the optimal weight \( w_j^* \) of leaf \( j \)
\[ w_j^* = \frac{-\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}, \]

(7)

and the optimal objective
\[
\hat{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \left( \sum_{i \in I_j} g_i \right)^2 \sum_{i \in I_j} h_i + \lambda + \gamma T, 
\]

(8)

which represents a scoring function that measures the quality of tree structure \( q \).

Since in practice it is impossible to enumerate all possible tree structures, a greedy algorithm is used that starts with a single leaf and iteratively splits a leaf into two leaves. Let \( I_L \) and \( I_R \) be the set of indices of left and right leaves after the split, and \( I = I_L \cup I_R \) the set of indices before. Then the loss reduction after the split is given by
\[
L_{\text{split}} = \frac{1}{2} \left[ \frac{\left( \sum_{i \in I_L} g_i \right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left( \sum_{i \in I_R} g_i \right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left( \sum_{i \in I} g_i \right)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma, 
\]

(9)

which is used in practice to evaluate the split candidates. By maximizing the loss reduction with respect to the features evaluated as split candidates and the corresponding split values, the best feature for the split is chosen from the data set. Therefore, the difference of the loss functions before the split and after the split is calculated and compared for all split candidates. In this way, the feature that best separates the data at the corresponding split note is chosen. While the first split is applied to the whole data set, for each following split a smaller data subset is considered. Thus, features that are taken for the splits on the first level of the decision tree will be accredited a higher importance as they consider a larger data set.

For our training of XGBoost to identify the essential internal coordinates of proteins HP-35 and T4L, we typically used 20 training rounds, i.e., an ensemble of 20 trees per classifier. For most parameters of the model, their suggested default values proved to be sufficient. For example, we used a maximum tree depth of 6, a learning rate \( \eta = 0.3 \) (to scale down the influence of updated weights for a more conservative optimization), the regularization parameter \( \lambda = 1 \), and the minimum loss reduction \( \gamma = 0 \) (required to make a further partition on a leaf node of the tree). We note that the exact choice of parameters is not critical, because we do not train the model for highest accuracy but only need enough split nodes such that sufficiently many features are screened for their respective importance.
FIG. S1. (a) Ramacolor plot obtained from dPCA+ based clustering of HP-35, describing the secondary structure of the 12 metastable states, where green, red and blue indicate $\alpha$-helical, $\beta$-extended and left-handed conformations, respectively. (b) Comparison of states obtained from clustering on dPCA+ data (bottom) and states obtained from clustering on six essential dihedral angles (top), demonstrating that both approaches yield quite similar states, with exception perhaps of some unfolded structures that are hard to assign to metastable states and may be reclassified during the process. (c) Ramacolor plot obtained for a reclustering of the MD data using only the six essential coordinates. To generate these states, we first shifted the data according to the maximum-gap approach in order to minimize the projection error. Then a combination of density-based clustering (using a hypersphere radius of 6 degrees) and most-probable-path dynamical clustering was performed in order to obtain 12 metastable states.
FIG. S2. Comparison of the most important components of HP-35 obtained from (left) machine-learning constructed essential internal coordinates, (middle) dPCA+, and (right) TICA. (a) Cumulative normalized variance obtained for the three methods. As expected, the essential internal coordinates exhibit smaller variances than the principal components, which are optimized to achieve maximum variance. Due to the inherent normalization of the components in TICA, the variances of all components are equal and therefore rise slowly. (b) Free energy curve along the first six components obtained for the three methods. The normalization in TICA is seen to significantly distort the true one-dimensional projections of the free energy. The essential dihedral angles reveal several metastable states, separated by significant energy barriers. (c) Autocorrelation functions of the first six components. While the decay time of the first component is similar for all three methods, we find that the autocorrelations of the essential internal coordinates decay on average slower than the autocorrelations of the dPCA+ and TICA components, although the latter are optimized to achieve long timescales.
FIG. S3. Composition of the first 20 eigenvectors of HP-35, as obtained from (a) dPCA+ and (b) TICA. While the first component is delocalized over all coordinates in both methods, higher components of dPCA+ and TICA are seen to differ. Interestingly, principal components 2, 3, 4, 5 and 7 show high contributions from the essential coordinates at the N-terminus, $(\phi_3, \psi_2$ and $\phi_2)$. The TICA eigenvectors evidently can not be easily interpreted in terms of the essential coordinates.
FIG. S4. Comparison of (top) metastable states obtained in the original four-state model of T4L by Ernst et al.\textsuperscript{12} and (bottom) states obtained from clustering on the three most important contact distances identified by machine-learning from a two-state model (see main text), demonstrating that both approaches yield quite similar states.
REFERENCES