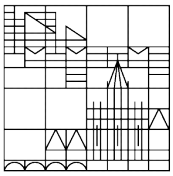
Using machine learning to find simplified representations of molecular systems

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
(Dr.rer.nat.)

vorgelegt von
Tobias Adrian Lemke

an der



Mathematisch-Naturwissenschaftliche Sektion
Fachbereich Chemie

Konstanz, 2020

Tag der mündlichen Prüfung: 24.02.2021

1. Referentin: Prof. Dr. Christine Peter

2. Referent: Prof. Dr. Malte Drescher

3. Referent: Prof. Dr. Andreas Zumbusch

Danksagung

An erster Stelle möchte ich mich bei dir bedanken, **Christine**. Ich glaube dass du, mit einer perfekten Kombination aus enger Betreuung und gleichzeitig maximalem Freiraum, ganz wesentlich zum Erfolg dieser Promotion beigetragen hast. Ich weiß noch genau wie ich dir in einer der ersten Besprechungen am Anfang meiner Promotion irgendetwas Dubioses von kleinen A.I.-Piloten erzählt habe, die in ihren Atom-Raumschiffen sitzen und lernen wie richtige Atome zu fliegen. Du hast mich trotzdem einfach mal machen lassen und standest mir gleichzeitig jederzeit in unseren wöchentlichen Treffen zur Seite. Für dieses, mir immer wieder entgegengebrachte Vertrauen und die Unterstützung, möchte ich mich ganz herzlich bedanken. Auch finanziell hat es mir in meiner Promotion an nichts gefehlt. Ich denke dabei an die generelle Ausstattung und die Computerressourcen aber auch besonders an die zahlreichen Konferenzen, die ich besuchen durfte, und an das Praktikum in Kalifornien, welches du unterstützt hast. Für die ganze Arbeit, die du leistest, um dafür zu sorgen, dass man sich als Doktorand wenig um Geld kümmern muss, möchte ich mich ebenfalls ganz herzlich bedanken.

Malte, dir möchte ich für die Übernahme des Zweitgutachtens danken. Ich bin auch immer noch dankbar für die Zeit, die ich in deiner Arbeitsgruppe verbringen durfte, wo ich das Programmieren gelernt habe und das erste Mal mit komplexerer Datenauswertung in Berührung kam.

Vielen Dank, **Herr Zumbusch**, für die Übernahme des Drittgutachtens.

Vielen Dank, **Herr Mecking**, für die Übernahme des Prüfungsvorsitzes.

Liebe **Arbeitsgruppe**, euch möchte ich ganz besonders für die angenehme und entspannte aber auch produktive Arbeitsatmosphäre danken. Ich finde es toll wie selbstverständlich es in der Arbeitsgruppe ist, dass man sich gegenseitig weiterhilft, egal ob es dabei um die persönliche Expertise oder das Teilen von Simulationsdaten geht. Vielen Dank auch für die vielen spannenden Diskussionen, die wir zu Wissenschaftsthemen oder aber auch ganz anderen Themen hatten.

Sasha, vielen Dank, dass ich viel über dimensionality reduction von dir lernen durfte. Ohne dich wäre EncoderMap wahrscheinlich so nicht entstanden.

Auch wenn ich immer versucht habe, möglichst wenige eigene Simulationen zu machen, um mehr Zeit für die Methodik zu haben, stand ich dabei oft genug vor Problemen. Vielen Dank, **Christoph** und **Micha**, dass ihr mir da in unzähligen Situationen weitergeholfen habt. In dem Zusammenhang möchte ich mich auch bei euch, **Andrej** und **Alok**, dafür bedanken, dass ich mich dann mit euren Simulationsdaten austoben durfte.

Vielen Dank auch euch, **Moritz**, **Jonathan** und **Felix**, dass ihr im Rahmen eurer Bachelorarbeiten so motiviert an den verschiedenen Projekten mitgearbeitet habt.

Ich bin auch dankbar für die tolle Zeit, die ich am Anfang meiner Promotion in Kalifornien verbringen durfte. Thank you, **Geoffrey** and **Baron**, for all the things that you taught me about Monte-Carlo simulations and sampling in general.

Auch wenn ich damit deutlich vor den Anfang der Promotion zurückgehe, möchte ich auch noch dir, **Martin**, danken. Ich glaube jeder, der irgendwann Programmieren lernt und sich nicht mehr vorstellen kann, wie manche ohne diese Fähigkeit zurechtkommen, wird nie vergessen, wer ihm das beigebracht hat. Vielen Dank, dass du dir so viel Zeit dafür genommen hast.

Zu guter Letzt möchte ich noch meinen **Freunden** und meiner **Familie**, besonders meinen **Eltern** danken. Dazu muss man vielleicht wissen, dass ich in meiner Schulzeit in jedem einzelnen Diktat eine Sechs geschrieben habe – nicht gerade Ergebnisse, mit denen man sich im deutschen Bildungssystem auf direktem Weg zum höchsten Bildungsabschluss befindet. Daran, dass ich trotzdem so weit gekommen bin, habt ihr einen ganz großen Anteil. Vielen Dank!

Nele, praktisch unsere gesamte Studien- und Promotionszeit haben wir zusammen bestritten und trotz aller Herausforderungen ist dabei die Freude nie zu kurz gekommen. Danke, dass du diese Zeit so lebenswert gemacht hast.

Contents

Introduction

In recent years, machine learning techniques have become very popular, and they surround us in our daily life already. Machine learning helps us to rank Internet search results [Burges et al., 2005], enables software to read hand writing [LeCun et al., 1990], recognize voice commands [Hinton et al., 2012], and recognize spam emails [Rothwell et al., 2004]. All these examples and most other examples of particularly successful machine learning applications have one aspect in common: they involve the processing of large amounts of data. Another aspect of tasks that can be successfully accomplished by machine learning is their complexity, which prevents us from solving it by defining few simple rules. A conventional programmer would struggle to come up with rules that accurately separate spam emails from normal emails. In an attempt to get rid of all spam emails designed to make us believe we won something, a simple algorithm might filter mails containing the three-letter combination "win". However, emails from a person with the surname "Winter" might be important to us, and maybe sometimes we are even lucky and truly win something. It is quite obvious in the spam email example that a single descriptor, like the occurrences of some word, is not enough to solve the problem. A successful spam filter, instead, has to take numerous descriptors into account. It has to deal with a high-dimensional descriptor space.

Molecular simulation also requires processing of large amounts of high-dimensional data. A fundamental outcome of molecular simulations is trajectories that contain the positions of all atoms for thousands or even millions of frames in time. A simulation of, for example, a small protein already contains hundreds of atoms, and the complexity of the investigated systems keeps growing. The task

of handling such large amounts of data is becoming more and more a demanding challenge in its own right. Two problems, virtually all simulation studies are facing, arise from the complexity of the investigated systems: the computational cost of the simulation grows with the complexity of the simulated system, and the resulting data becomes harder to interpret. Finding simplified representations in the context of molecular simulations is, therefore, important to reduce computational cost and to condense the massive amounts of data into a human understandable form. The complexity of these challenges and the large amounts of data available, render molecular simulation an ideal area to take advantage of the recent developments in machine learning.

In the context of this thesis, we explored how machine learning can be used to find simplified representations of molecular systems. Section 2.1 describes the basic methodology of molecular simulations and machine learning. Interestingly, we will see that the methodology of both fields is quite intimately linked. Equipped with this basic methodology, we take a deeper look into the two above mentioned problems arising from the increasing complexity of the molecular systems under investigation: the problem of computational cost and the problem of overwhelming amounts of data. These problems are described in Section 2.2 along with possible solution strategies. This section covers previously known solution strategies and also points to the solution strategies developed in the context of this theses, which are described, in detail, in the publications included in Section 3.

2.1 Basic Methodology

This section aims to provide an introduction to the basic methodology of molecular simulation and machine learning, which is necessary to understand the rest of the thesis. A special emphasis is placed on the deep connection between the

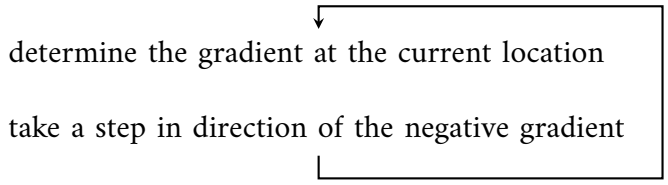
methodologies of both fields which becomes evident if we look at them in the light of optimization.

2.1.1 Optimization

Optimization is probably the concept most central to this thesis. Optimization is the minimization or maximization of some function with respect to its variables. [Nocedal and Wright, 2006] In machine learning, this applies, for example, to the training of an artificial neural network, where the parameters (variables) of the network are varied to minimize a cost-function, which is a measure of how well the network performs a given task: e.g. classify spam emails. It also applies to protein folding, which is a prominent application of molecular simulation, where finding low-energy conformations requires the minimization of an energy function with respect to its variables. The variables in this case are all the protein degrees of freedom: the bond lengths, bond angles, dihedral angles etc.. A popular strategy to finding minima for such high-dimensional functions is gradient descent.

Gradient Descent

Gradient descent is an iterative process to minimize some function. The steps involved in a basic gradient descent algorithm are:



The route that such a simple gradient descent algorithm would take in a two-

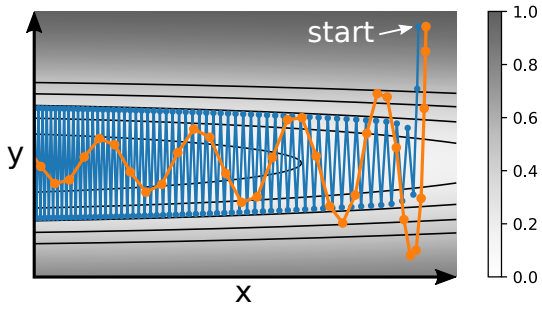


Figure 2.1: Gradient descent in a two-dimensional example landscape. The black lines represent contour lines of the landscape. The blue trace represents the path, a gradient descent algorithm with a constant learning rate (constant factor multiplied with the gradient) would take. The orange trace represents the path, a gradient descent algorithm with damped momentum would take.

dimensional example landscape is shown in Figure 2.1 with a blue line. In this example it becomes evident that the route, plain gradient descent takes toward the minimum, is not necessarily the direct one. Especially in cases with an inhomogeneous gradient (steep in one direction and shallow in an other direction), the gradient descent algorithm is known to be inefficient. In such cases, the step size tends to be too small in the shallow direction, which prevents quick progress in this direction; and too large for the steep direction, which leads to constant overshooting.

A common way to tackle this problem is the introduction of a damped momentum term. [Qian, 1999] The orange line in Figure 2.1 shows a path, gradient descent with damped momentum would take. We can imagine this as a marble rolling down on a landscape representing the function to minimize. In the steep y-direction the sign of the gradient frequently changes and, due to the damping (friction), the velocity in the y-direction decreases. In the less steep x-direction, the gradient points to the same direction for multiple steps, and because of the momentum, the marble will pick up speed in this direction. The minimum is approached faster compared to plain gradient descent.

An additional advantage of adding momentum to the algorithm is the chance to leave local minima. Imagine a marble rolling down a slightly rough surface. Once the marble has picked up some momentum downhill, it will not get trapped in small dents in the surface. A plain gradient descent algorithm, instead, has no chance to leave any local minima.

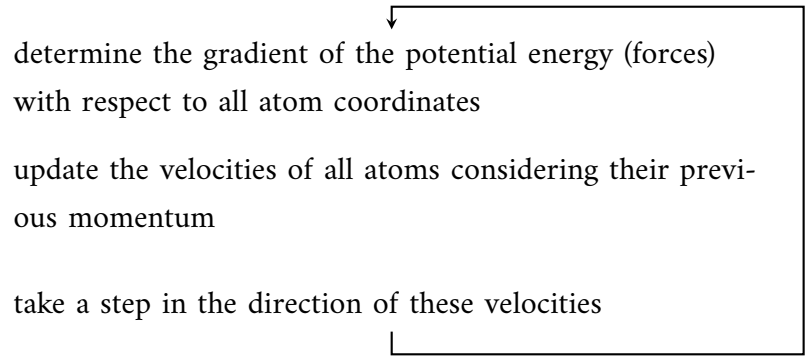
Modern optimizers for neural networks, such as the Adam optimizer [Kingma and Ba, 2014], which was used in this work, include momentum because of the above mentioned advantages. Momentum is also used in molecular dynamics algorithms, as we will see in the next section, though mainly for a different reason.

2.1.2 Molecular Simulation

Molecular Dynamics Simulations

The following is a somewhat unconventional introduction to molecular dynamics simulations, highlighting the close relation to gradient descent. For a more conventional introduction see [Zuckerman, 2011].

A molecular dynamics (MD) algorithm is basically a gradient descent algorithm with a momentum term:



MD simulation differs from gradient descent optimization algorithms in that we are usually not (only) interested in the minimum. We are also interested in the path the algorithm takes, and we want this path to resemble the path a true molecular system would take. For protein folding simulations, this would mean, for example, that we do not only want to find the most stable folded state of the protein, instead, we also want to see how the protein reaches that state. We include momentum, therefore, not because it might speed up the search for the optimum, as described in section 2.1.1, but because of the underlying physics of Newton's laws of motion. The masses of all atoms and consequently their momenta need to be considered in a simulation to obey these laws.

An additional reason why we are usually not interested in the minimum (lowest energy structure), is that it is actually quite unimportant in real molecular systems. The energetic minimum only plays an important role when a system is close to 0 Kelvin. For any conditions other than 0 Kelvin, instead, molecular systems visit a variety of different structures. It is this ensemble of structures, accessible under certain conditions, that we are interested in. A common choice of conditions is for example: constant number of molecules/atoms, constant volume, and constant temperature. An ensemble of structures accessible under such conditions is called a canonical ensemble or NVT-ensemble.

In Section 2.1.1 we compared momentum-gradient-descent with a marble rolling

down on some landscape. With a damping term (friction) the marble will roll down on the landscape and will ultimately come to rest in some minimum. If we remove all the friction, the marble would instead continue to roll around on the landscape, visiting (sampling) different states accessible with its initially given energy. This represents a simulation at constant energy and results in an ensemble of states at constant energy.

To sample an ensemble at constant temperature, we have to extend the algorithm with a thermostat. In MD simulations, a thermostat adjusts the temperature by manipulating the atom velocities. [Frenkel and Smit, 2001] In the marble example you can imagine this with a marble rolling on some landscape while it occasionally receives kicks from some invisible agent trying to adjust the velocity distribution of the marble. The distribution of atom velocities at a given temperature is described by the Maxwell-Boltzmann distribution. It is this distribution that a proper thermostat aims to establish in the simulation. When, in addition to the temperature, also the volume and the number of atoms is kept constant, an MD simulation results in an ensemble of states weighted according to their Boltzmann weights:

$$p(x) \propto \exp\left(-\frac{U(x)}{k_B T}\right) \tag{2.1}$$

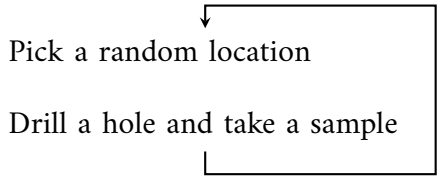
where x are the coordinates of all atoms, $p(x)$ is the probability density, $U(x)$ is potential energy, k_B is the Boltzmann constant, and T is the temperature. [Zuckerman, 2011]

In conclusion, MD simulation is a tool to sample an ensemble of structures accessible under certain conditions, and, while sampling, the MD simulation takes a physically reasonable path, which allows to get insights into the dynamics of the system. An MD simulation, however, crucially relies on the availability of the gradients (forces). When these gradients are not available, a different simulation technique has to be used.

Monte-Carlo Simulations

Like a molecular dynamics simulation, a Monte-Carlo simulation can be used to sample an ensemble of molecule conformations. The name "Monte-Carlo" refers to the city in Monaco, known for its casinos and gambling. "Monte-Carlo" is basically used as a synonym for randomness and "Monte-Carlo sampling" just means random sampling.

Imagine you owned a stretch of land and were looking for the best spot to dig for gold. The Monte-Carlo approach to finding the best spot would be:



Irrespective of how much gold you found in the previous hole, you would just continue to pick a completely random location for your next hole. Probably, you would not be a very successful gold digger with this strategy.

Let us assume that gold is not completely randomly distributed but occurs in veins of gold and that we are limited in the number of holes we can drill. Once you found some gold, it is probably more effective to pick a location for your next hole in the proximity of your previous hole where you already found gold. If you found more gold in the next hole, you would probably continue to search near this new hole. If you found less gold in the new hole, you would rather go back to the previous hole and continue your search from there.

These ideas can be directly transferred to molecular systems. In principle we could do a completely random sampling of conformations. However, we would end up sampling a lot of conformations which are not at all important because they are energetically very unfavorable, for example, because of overlapping

atoms. Instead, we could do the "advanced gold digger strategy" and rather sample molecular conformations in the proximity of already found, energetically-favoured conformations. A way to implement such an "importance sampling", is the Metropolis-Monte-Carlo algorithm:

create a trial conformation by slightly changing your current conformation in some random direction

according to the acceptance criterion in Equation 2.2, either accept this trial conformation and make it your new current conformation, or reject the trial conformation and continue with your old conformation

The acceptance criterion in a Metropolis-Monte-Carlo simulation is:

$$\text{uniform random number between 0 and 1} < \exp\left(\frac{E_c - E_t}{k_B T}\right) \tag{2.2}$$

where E_c is the energy of the current conformation, and E_t is the energy of the trial conformation [Metropolis et al., 1953]. With this criterion, a trial conformation with lower energy compared to the current conformation is always accepted. A trial conformation with a higher energy compared to the current conformation is only accepted with a probability according to the Boltzmann term given in Equation 2.2. This criterion ensures that the algorithm rarely samples "unimportant" (very high energy) regions of the conformational space, because trial conformations with such high energy are rarely accepted. The algorithm will rather focus on exploring the "important" (low energy) regions. Monte-Carlo can be used as an optimization method for example to find the lowest energy conformation (or the best spot to dig for gold). However, as already mentioned in the context of molecular dynamics simulations, we are usually not particularly interested in the minimum of the energy landscape of a molec-

ular system. Instead, we are interested in the ensemble of conformations occurring under certain conditions. The true power of the Metropolis-Monte-Carlo acceptance criterion is that it allows to sample an ensemble of conformations where the probability of visiting a conformation is proportional to the Boltzmann factor $\exp\left(-\frac{E_c}{k_B T}\right)$. [Frenkel and Smit, 2001]

In conclusion, like a molecular dynamics simulation with a thermostat, a Metropolis-Monte-Carlo simulation can be used to sample an ensemble of conformations for a given temperature. However, in contrast to a molecular dynamics simulation, no gradients are required, which can be very useful whenever gradients are hard to obtain. On the downside, the path a Monte-Carlo simulation takes through the conformational space is typically not following any physical rules and gives no information about the dynamics of a system.

2.1.3 Machine Learning

Like molecular simulation, machine learning is also closely related to optimization. A machine or program is learning when it improves its performance at some task by optimizing some performance measure. [Mitchell, 1997] Let us consider a spam filter as an example. The task of the program is to classify emails into spam and non-spam mails. The performance measure could be the percentage of misclassified mails. When the program changes in some way to minimize this performance measure, it learns how to correctly classify spam mails.

There are many different approaches to machine learning. The following section discusses only the approach used in this work: artificial neural networks.

Artificial Neural Networks

An artificial neural network is built from artificial neurons also called perceptrons. An artificial neuron is a unit which takes a weighted sum of all its inputs, applies an activation function to this sum, and returns one number as output, the result of this calculation. A scheme of such a neuron is shown in Figure 2.2.

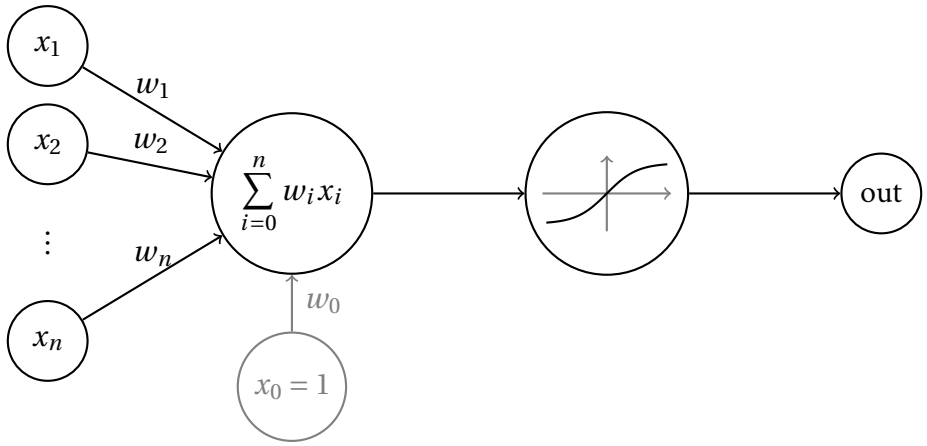


Figure 2.2: Scheme of an artificial neuron. All inputs x_i are added in a weighted sum. An activation function like tanh is applied to this sum, and results in some output value which can be one of the inputs of a following neuron.

Inputs to a neuron can either be external inputs to the neural network or outputs of other neurons. Typically there is one extra input set to 1. Changing the weight of this input, also called bias, allows to shift the values relative to the activation function as needed. The activation function is applied to add non-linearity to the neuron. We can understand the advantage of such a non-linearity, as e.g. a step-like or sigmoidal shaped activation function adds the possibility to represent a decision boundary. In the case of a tanh activation function for example, weighted sum values below the turning point result in an output close to -1, and weighted sum values above the turning point result in an output close to $+1$. This essentially allows to implement an "if query" with a neuron, but also

linear relations can be represented using the almost linear part of the tanh function around its turning point. The scaling and shifting of the activation function necessary to implement such different behaviours can be achieved by altering the weights w_i of the neuron.

The representational capabilities of a single neuron are very limited. However, when multiple neurons are combined in a network, very complex relations can be represented. A common way to arrange neurons in a network is to place them in layers where the outputs of the neurons in the previous layer are the inputs of the neurons in the current layer. Such a network architecture, which is called feed-forward neural network, is illustrated in Figure 2.3. It was shown

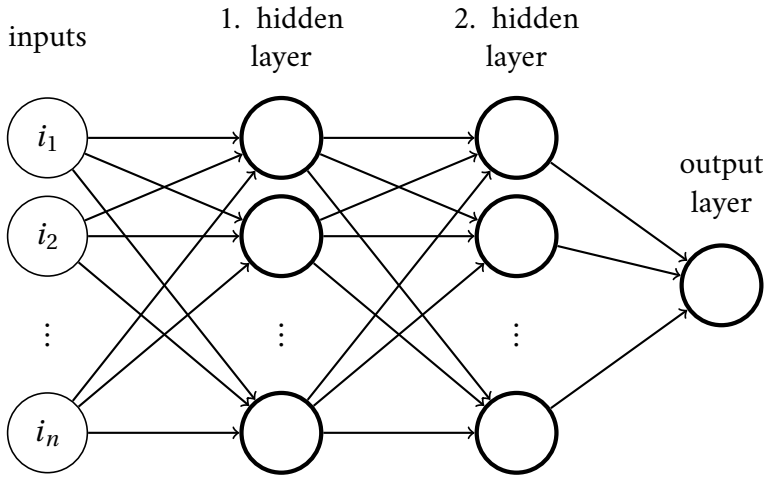


Figure 2.3: Example feed-forward neural network with two hidden layers and one output neuron. Each of the thick edge circles represents one neuron as illustrated in Figure 2.2.

that any function can be approximated to arbitrary accuracy by a neural network with two hidden layers and sigmoidal activation functions given that a sufficient number of neurons is used. [Cybenko, 1988, Mitchell, 1997]

A neural network like the one shown above in Figure 2.3 could be used for a

classification task. In the spam mail example, the network would be trained to give an output, with its single output neuron, to signify whether a mail is spam or not. We could, for example, train the network by minimizing the following simple cost function:

$$C = \frac{1}{n} \sum_{i=1}^{n} [N(I_i) - l_i]^2 \tag{2.3}$$

where $N(I_i)$ is the output of the network for the given input I_i, n is the number of training examples, and l_i is the label of a training example. Let's say we define the label of a spam mail to be $+1$ and the label of a none spam mail to be -1. The cost function is minimized, if the network learns to return $+1$ for a spam mail and -1 for a non spam mail. For the network to be able to fulfill this task, we need to provide suitable inputs, often called features. For spam detection, some of the features could for example be the number of times the words "money", "win" or "payout" occur in the mail.

Training of the network requires a labeled data set. This data set has to contain the chosen features for each example mail and the label signifying whether it is a spam mail or not. Then, typically, some variant of gradient descent is used to minimize the cost function with respect to the weights of the neural network. This leads to the following basic update rule:

$$w_{i,new} = w_{i,old} - \eta \frac{\partial C}{\partial w_{i,old}} \tag{2.4}$$

where C is the cost function, η is the learning rate, and w_i are the weights of the neurons. Often, this basic update rule is supplemented with other terms, such as a momentum term, to enhance the efficiency of the optimization/learning process (see also Section 2.1.1).

Figure 2.4 shows a different neural network architecture, designed for a different purpose. It shows a network with not just one output neuron but with the same number of output neurons as as there are inputs. Furthermore, it contains

a bottleneck layer: a layer with only few neurons in the middle of the network. Such a network architecture can be used as an autoencoder. An autoencoder is

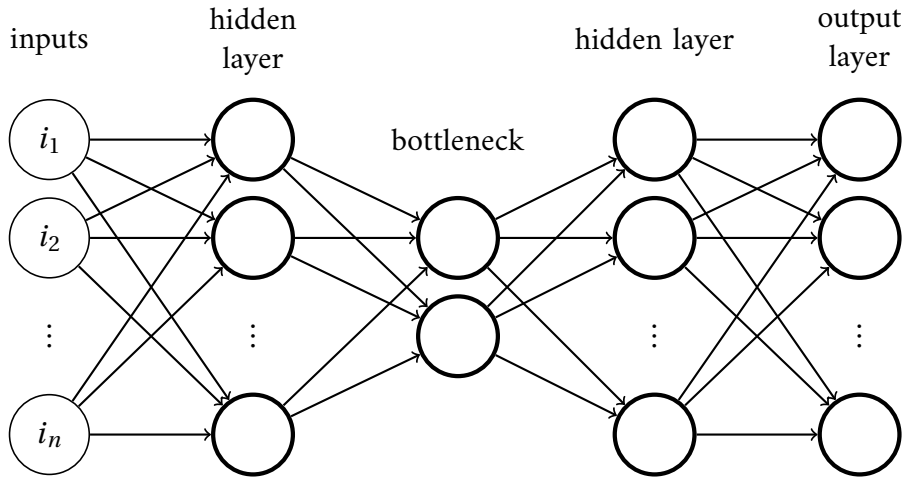


Figure 2.4: Example autoencoder architecture, where the number of output neurons is equivalent to the number of inputs. The network can be trained to reproduce its inputs while all the information is forced through the bottleneck. Each of the thick edge circles represents one neuron as illustrated in Figure 2.2.

a network that is trained to reproduce its inputs. A network which only returns its inputs seams to be pointless at first. The clue here is the bottleneck layer. When the network reproduces its inputs, the information necessary to do so is passed through the bottleneck. The neural network part to the left of the bottleneck encodes the information of the input space into a lower-dimensional representation. The neural network part to the right of the bottleneck decodes this information and reconstructs the-high dimensional input. The encoder part is valuable because it can be used for dimensionality reduction. It compresses data into a lower-dimensional space. The decoder part is valuable because it allows to generate new high-dimensional data points from low-dimensional inputs.

In conclusion, a neural network can be seen as a flexible model which can be trained, by minimizing some cost function, to fulfill some task. Typical appli-

cations include classification and regression tasks. With autoencoders, neural networks also provide a way to translate between representations of different dimensionality.

2.2 Problems and Solution Strategies

As described in Section 2.1 the aim of molecular simulations is to generate a representative ensemble of molecule conformations for given conditions such as a given temperature. The increasing complexity of the molecular systems under investigation causes two problems virtually all molecular simulation projects are facing: the growing complexity increases the computational cost required to obtain a representative ensemble, and the analysis of the resulting ensemble of conformations is becoming more and more a challenge in its own right. This section describes the two problems in more detail, presents literature known solution strategies, and points to the new solution strategies developed in the context of this thesis using machine learning.

2.2.1 Computational Cost of Simulations

The computational cost of a molecular simulation is strongly correlated with the number of atoms involved. The number of pair interactions that need to be evaluated each time step grows quadratically with the number of interacting atoms, and even a small chemical system like a protein, surrounded by water, can easily involve thousands of atoms. Larger entities like cell organelles or even complete cells are (and maybe always will be) out of reach for simulations with atomistic resolution.

The computational cost can also be a limiting factor, even with only few atoms in a system, whenever long timescales are of interest. In atomistic molecular dynamics simulations, the new positions of all atoms are, typically, evaluated each 2 fs of simulated time. Interesting rare events like nucleation or protein folding, however, occur on μs time scales and beyond. [Kubelka et al., 2004, Pound and Mer, 1952] To put this in comparison, to simulate 1 s with a femtosecond time step, requires roughly the same number of steps as a simulation of earths entire history with a one minute time step would take, including the formation of our planet, first life, dinosaurs, and the development of human life. [Allegre et al., 1995, Harris, 2018]

Fundamentally, there are two ways to tackle the problem of computational cost in a simulation. Either the cost per simulated time step needs to be reduced to allow for longer simulations, or a more representative ensemble of conformations needs to be sampled in the same amount of steps. Enhanced sampling approaches aim to enhance the sampling with the same number of simulation steps. On the contrary, coarse-grained simulations are a way to decrease the cost per simulated time step by reducing the resolution of the model. In the following, both strategies are discussed.

Enhanced Sampling

In systems with large energetic or entropic barriers, simulations tend to get trapped on one side of such barriers. Enhanced sampling approaches aim to facilitate crossing such barriers. One approach to enhance the sampling is to use biasing potentials to force the simulation into specified regions of interest or into new regions of the conformational space. Umbrella sampling [Torrie and Valleau, 1977] and Flooding/Metadynamics [Huber et al., 1994, Grubmüller, 1995, Barducci et al., 2008] are examples of such methods.

A different approach is to combine molecular dynamics with Monte-Carlo simulation methods (see also Section 2.1.2). The sampling can be enhanced when one Monte-Carlo move takes the simulation directly from one side to the other side of a high barrier instead of trying to climb the barrier with a series of unlikely, uphill steps. To take the simulation directly from on side of a barrier to the other, a large step size is required. Attempting large Monte-Carlo steps into a random direction, however, is not feasible in a dense system because virtually all attempted moves would result in conformations with overlapping atoms that would be rejected because of their high energy. The Replica exchange simulation scheme solves this problem by attempting Monte-Carlo moves to conformations sampled in parallel molecular dynamics simulations (replicas), which by construction do not contain any overlapping atoms. To enhance the sampling, the parallel simulations need to be manipulated in a way to facilitate the crossing of barriers. This can be achieved with elevated temperatures (temperature replica exchange [Sugita and Okamoto, 1999]) or with weakened interactions (Hamiltonian replica exchange [Fukunishi et al., 2002]). Even though the parallel simulations are manipulated, the correct weighting of states according to Boltzmann weights can be preserved due to the Metropolis-Monte-Carlo acceptance criterion (Equation 2.2).

In the publication: "Efficient sampling and characterization of free energy landscapes of ion-peptide systems" [Lemke et al., 2018] we used Hamiltonian replica exchange to enhance the sampling of ion-peptide systems. The main challenge of setting up a successful Hamiltonian replica exchange is to identify all important barriers preventing an efficient sampling. Ion-peptide systems suffer from slow sampling mainly because of large barriers caused by strong ion bridges, but also the backbone dihedral torsions hold large barriers [Kahlen et al., 2015]. In our publication, we provide a suitable set of parameters; including increased temperature, weakened ion interactions, and flattened dihedral torsions; for a successful Hamiltonian replica exchange of ion-peptide systems. The resulting

ensemble contains an abundance of different states with varying patterns of ion bridges which provides for an excellent example to test different characterization methods dealing with overwhelming amounts of data. The parameters that were changed and the resulting ensemble of conformations are illustrated in Figure 2.5. We will come back to this example in Section 2.2.2 which is focused on the analysis aspect of such ensembles.

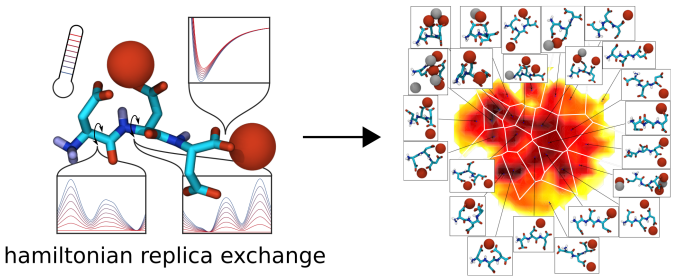


Figure 2.5: Illustration of the changes made to the Hamiltonian in the replica exchange scheme at the example of an aspartic acid trimer in the presence of calcium ions. Through these changes, sampling of a broad conformational space (depicted on the right) is possible. *Adopted with permission from J. Chem. Theory Comput. 2018, 14, 11, 5476-5488. Copyright 2018 American Chemical Society*

Both discussed enhanced sampling approaches, using biasing potentials or incorporating Monte-Carlo moves, do not reduce the cost per simulation step. They promote the sampling of a more representative ensemble in a given number of steps by facilitating the crossing of barriers. Coarse-graining, instead, is an approach where the cost per simulated time step is reduced.

Coarse-Graining

On an atomistic level of resolution, each atom is represented with one bead. In coarse-grained simulations, larger entities of the molecule, for example complete functional groups, are represented with one bead. [Noid et al., 2008] Be-

cause of the quadratic scaling, the reduced number of interacting beads greatly reduces the computational cost. On the down side, coarse-grained models, by construction, lack detail and might not be able to accurately model certain aspects of molecular systems. The goal of a coarse-grained model is still to reproduce the behaviour of the reference system as accurately as possible.

In bottom-up coarse-graining methods, the coarse-grained interactions are parameterized such that an accurate representation of a known atomistic sampling is reproduced. A molecular dynamics simulation of an atomistic system is propagated according to the force derived from the gradient of its potential energy function (see also Section 2.1.2). In a coarse-grained model, multiple atomistic conformations may be represented by the same conformation. Imagine a chain of three atoms where the coarse-grained model only contains the two outer atoms as shown in Figure 2.6. All atomistic conformations whose two

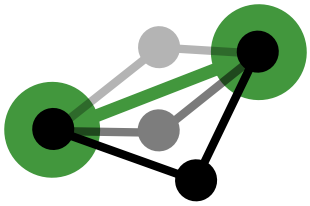


Figure 2.6: Sketch of a coarse-grained model that contains the two outer atoms of an atomistic chain with three atoms. One coarse-grained conformation bundles multiple atomistic conformations with different central atom positions.

outer atoms have the same distance result in the same coarse-grained conformation, irrespective of where the central atom is located. To mimic the behaviour of the atomistic system in the coarse-grained representation, we have to average over the forces acting in all the possible underlying atomistic conformations. So, while an atomistic simulation is based on a potential energy, a coarse-grained simulation is propagated according to a mean force which is derived from the "potential of mean force", which is a free energy. The required free energy function for the example shown in Figure 2.6 could be easily obtained through Boltz-

mann inversion of the probability density distribution from an atomistic ensemble. This would be easy in this case because the required probability density distribution has only one dimension: the distance between the two coarse-grained beads. For a more complex system, however, the required probability density distribution is high-dimensional; it is a function of all the bead positions. Extracting this high-dimensional, multi-body probability density distribution is problematic mainly because of sparsity issues. Imagine we would try to extract a probability density distribution through a binning approach with 100 bins for each dimension. Even for a ten dimensional case, which is tiny compared to a molecular system like a protein, this would result in $100^{10} = 10^{20}$ bins. With such a giant number of bins we would probably end up having more bins than sampled conformations in our ensemble, and most of the bins would end up being empty or would only contain a single conformation. Instead of trying to obtain the full high-dimensional free energy landscape directly, the free energy surface is therefore usually approximated as the sum of low-dimensional potentials with methods such as iterative Boltzmann inversion [Reith et al., 2003], relative entropy [Shell, 2015] or force matching [Noid et al., 2008].

In our publication: "Neural Network Based Prediction of Conformational Free Energies–A New Route toward Coarse-Grained Simulation Models" [Lemke and Peter, 2017] we introduce a new strategy to extract high-dimensional free energy surfaces from atomistic data to build coarse-grained models. The central idea of this approach is to convert the task of finding a probability density into a classification problem solved by an artificial neural network. A similar concept was used by [Garrido and Juste, 1998] and [Gutmann and Hyvärinen, 2010] in a different context and is also referred to as noise-contrastive estimation. In the classification problem constructed in [Lemke and Peter, 2017], the task of the neural network is to distinguish between real, sampled conformations and "fake" conformations drawn from a known distribution. Based on how confident the neural network is that a given conformation is from the en-

semble of sampled conformations and not a ”fake” conformation, it is possible to calculate the probability density for this point in the conformational space. With this approach, to our knowledge, we have been the first to apply machine learning with neural networks to obtain coarse-grained models for molecular simulations. Since then, several approaches have been published. [Zhang et al., 2018, Bejagam et al., 2018, Wang et al., 2019, Duan et al., 2019] Also, approaches using machine learning techniques other then neural networks have been described. [John and Csányi, 2017]

One advantage of our machine learning approach is that it allows for exceptionally coarse representations. The development of coarse-grained models is always a race with the progress in computer hardware. A system which is only accessible with a coarse-grained model today might already be accessible in full atomistic resolution, with new hardware, a few years later. Creating exceptionally coarse models, like our peptide models, is therefore especially valuable to stay further ahead of the hardware developments. Parameterizing exceptionally coarse models, however, is a challenge. An accurate description with a coarser representation requires the interaction potentials to be more sophisticated. This is where our proposed strategy to extract high-dimensional free energy surfaces can unfold its potential. We demonstrated this by creating a coarse-grained peptide model where only interesting functional groups in the side chains are represented with a bead, and where the rest of the peptide, including the backbone, is represented implicitly with the potential learned by the neural network. We showed that the created model is not only capable to reproduce the behaviour of the atomistic references it was trained on but is also able to make reasonable predictions for longer chains.

Creating simplified representations of molecular systems on a level where few atoms are represented with a single bead can be a powerful tool to reduce the computational cost of simulations. Such models, however, are still high-dimensional. A model representing a protein with only one bead per amino acid

can still easily have a conformational space with hundreds of dimensions. After a simulation, whether coarse-grained or atomistic, we still have to deal with this overwhelming amount of high-dimensional data.

2.2.2 Overwhelming Amount of Data

The primary result of a molecular dynamics simulation is a trajectory containing the positions of all atoms (or beads) for each time frame. While obtaining such data is already challenging, as described in the previous section, extracting all relevant information from this overwhelming amount of data can be equally challenging.

The simplest approach to analysing trajectory data is to make an (educated) guess for an important variable, for example some distance between two residues, and to analyse this variable. If the researcher has already some knowledge about the system, for example from experimental data, this can be a viable approach. The danger of this approach, however, is that the researcher will only find what he or she is specifically looking for.

A more systematic approach to identifying important variables and to extracting relevant information is given by dimensionality reduction algorithms. Dimensionality reduction can be seen as a special kind of machine learning where the correct solutions are not known beforehand. In case of spam detection for example, the training is typically done with labeled data, a data set where we, as a supervisor, already know which emails are spam and which are not. In the case of dimensionality reduction this is different. We do not know beforehand, what the best way to arrange the data in the low-dimensional projection is. The machine learning algorithm is supposed to come up with that on its own. Dimensionality reduction is, therefore, a form of unsupervised learning.

The most common dimensionality reduction algorithm is Principal Component Analysis (PCA) [Pearson, 1901]. PCA essentially finds the directions of maximum variance. This is illustrated in Figure 2.7 for two-dimensional examples. Figure 2.7 also illustrates that a linear method like PCA is not equally suitable for

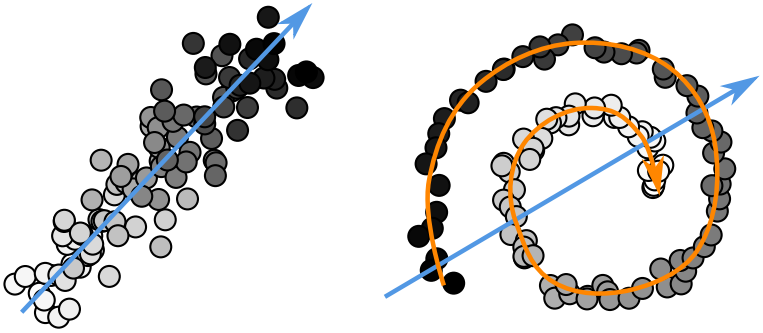


Figure 2.7: Dimensionality reduction for two-dimensional example data sets. For the left example, PCA, which finds the direction of maximum variance (blue arrow), is a suitable approach to finding a good one-dimensional descriptor. For the example on the right, a nonlinear path (orange) is much better suited to describing the data then any linear descriptor. This is an example where PCA and linear dimensionality reduction methods in general are not well suited.

all kinds of data sets. For some dimensionality reduction problems it is crucial to unravel nonlinear features. This applies, for example, to protein folding [Das et al., 2006] and presumably most other molecular systems studied in simulations. Nonlinear dimensionality reduction techniques, such as diffusion map [Coifman et al., 2005], time-lagged independent component analysis [Pérez-Hernández et al., 2013], neural network autoencoders [Wehmeyer and Noé, 2018, Chen et al., 2018, Sultan et al., 2018, Ribeiro et al., 2018, Hernández et al., 2018], or variants of multidimensional scaling [Cox and Cox, 2000] such as sketch-map [Ceriotti et al., 2011], are therefore popular in the simulation community. [Tribello and Gasparotto, 2019]

In our publication: "Efficient sampling and characterization of free energy landscapes of ion-peptide systems" [Lemke et al., 2018] we tested different dimen-

sionality reduction approaches. We did this for an ensemble of ion-peptide conformations obtained from an enhanced sampling approach with Hamiltonian replica exchange, as mentioned already in Section 2.2.1, and we placed special emphasis on combining the conformations sampled in the different simulations with different Hamiltonians into one unified map using the transition-based reweighting method (TRAM) [Wu et al., 2016]. We found that the multidimensional scaling method sketch-map is very useful to perform dimensionality reduction for such systems, however we also ran into problems with the computational efficiency for large data sets.

In our publication: "EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations" [Lemke and Peter, 2019] we introduced a new dimensionality reduction algorithm. An introductory video to EncoderMap can be found at `https://www.youtube.com/watch?v=JV59OABhNTY` [Lemke, 2019]. EncoderMap combines advantages of two known dimensionality reduction strategies: multidimensional scaling and neural network autoencoders (see also Section 2.1.3). Multidimensional scaling aims to arrange points in a low-dimensional representation in a way that preserves the pairwise distances of the points in the original, high-dimensional space. This involves solving an optimization problem where the positions of points in the low-dimensional space are varied to minimize the deviation between the high-dimensional and the low-dimensional pairwise distances. Similar to molecular dynamics simulations (as discussed in Section 2.2.1), this approach has an unfavorable scaling because the number of pairs that need to be evaluated grows quadratically with the number of points. EncoderMap circumvents this unfavorable quadratic scaling of multidimensional scaling. This is possible because it is not the arrangement of the points in the low-dimensional representation that is directly optimized. Instead, a neural network is trained to do the projection from the high-dimensional to a low-dimensional space with a multidimensional scaling metric as cost function. The advantage of this approach is that the network can be trained with batches

of data. The pairwise distances only need to be calculated within a batch of data used for one training step, and in the next training step a different batch of training points can be used. With this trick, low dimensional representations similar to the ones obtained with multidimensional scaling can be produced with a linear scaling with respect to the number of data points. Consequently, the time required to perform the dimensionality reduction is drastically reduced. A further advantage of the network trained to perform the dimensionality reduction is that it makes projecting additional points to the low-dimensional representation computationally very cheap. No further optimization problems have to be solved to fulfill this task. The neural network also provides a differentiable link between the high-dimensional and the low-dimensional space, which allows the use of EncoderMap in combination with enhanced sampling techniques where a biasing potential needs to be defined in a low-dimensional space (see also Section 2.2.1).

Additionally to the neural network projecting from the high-dimensional to the low dimensional space, we also train a network to reconstruct the full high-dimensional input from the low-dimensional projection. This makes the combined network a neural network autoencoder (see also Section 2.1.3) as illustrated in Figure 2.8. With this second neural network we obtain a tool to generate high-dimensional points for arbitrary points in the low-dimensional representation. In [Lemke and Peter, 2019] we demonstrated how useful this capability is by applying it to generate molecule conformations. We used EncoderMap to perform dimensionality reduction of sampled protein conformations from the 38-dimensional space of backbone dihedrals to a two dimensional map. Then, we generated new protein conformations for points distributed along different paths in the map. Given any two-dimensional input point in the map, the decoder part of the autoencoder generates a set of values for the 38 backbone dihedrals. From these dihedral values, the Cartesian coordinates of the protein backbone can be reconstructed, starting from some conformation, by

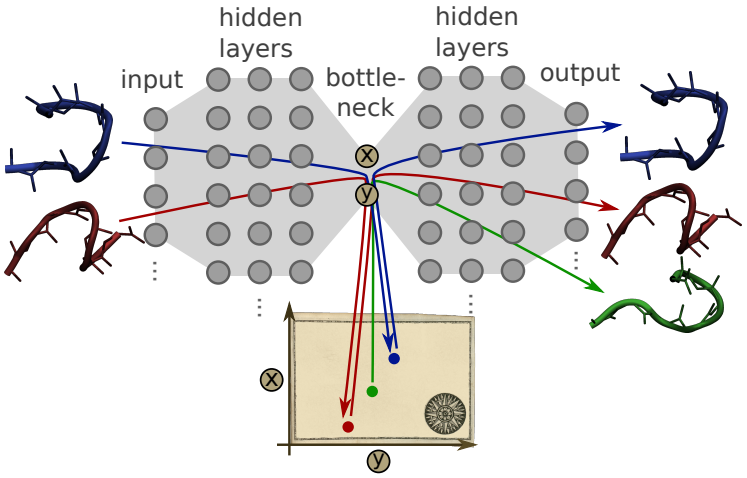


Figure 2.8: Example autoencoder architecture with two bottleneck neurons. This results in a projection of high-dimensional data, for example molecule conformations, to points in a 2-d space/map. The decoder part (right-hand side) can generate output conformations even for 2-d points in the map for which no corresponding input exists (green point/conformation). *Adopted with permission from J. Chem. Theory Comput. 2019, 15, 2, 1209-1215. Copyright 2019 American Chemical Society*

subsequent rotation around the different dihedral axis. We showed that these generated conformations for paths in the map can be very useful to visualize important conformational changes. The generated conformations only reflect the most important conformational changes identified during the dimensionality reduction process. These conformational changes are visualized without the distracting noisiness of a wiggling molecule, which allows for a unique perspective on high-dimensional conformational data.

In [Lemke and Peter, 2019] we showed that generating conformations in dihedral space works well for peptides or a mini protein like Trp-Cage. For larger proteins, however, this approach quickly reaches its limits. In our publication: "EncoderMap (II): Visualizing important molecular motions with improved generation of protein conformations" [Lemke et al., 2019] we extended the capa-

bilities of EncoderMap to accurately generate conformations of large proteins. The previously used backbone dihedrals are a good descriptor because they accurately describe the secondary structure and they are invariant to rotation and translation of the complete molecule. Furthermore, the reconstruction of a conformation in Cartesian coordinates from dihedral angles is quite straightforward. The long-range order, however, is not well captured in dihedral angles. Slight deviations in one dihedral can cause large offsets for amino acids further down the chain. As a consequence, the generated conformations of large proteins reconstructed from backbone dihedrals often contain the correct secondary structure elements, but these elements are usually not well aligned and might overlap. Pairwise distances, for example between C_α atoms, are a much better descriptor for long-range order and they are also translationally and rotationally invariant. Reconstructing Cartesian coordinates from pairwise distances, however, is not straight forward. Different distances might contradict each other, and an optimization problem has to be solved to find the conformation fitting best to the generated pairwise distances.

In EncoderMap(II), we combine the advantages of both descriptors: dihedral angles and pairwise distances. The neural network autoencoder still gets dihedrals as input and returns dihedrals as output, but from these dihedrals the Cartesian coordinates are reconstructed during the neural network training process. This allows to add a contribution to the cost function measuring the accuracy of the generated conformations in pairwise distance space. This way still unique conformations are generated and at the same time their long range order is improved because of the more sophisticated cost function. The challenge, here, is that the reconstruction of Cartesian coordinates from backbone dihedrals needs to be differentiable. As explained in Section 2.1.3, neural networks are typically trained with some variant of gradient descent; a weight in the network is updated according to the partial derivative of the cost function with respect to this weight. Including the reconstructed Cartesians in the cost function, there-

fore, requires to take the derivative of the cost function through the reconstruction and through the network back to each weight. To our knowledge, there is only one other example in literature where such a differentiable reconstruction of Cartesian coordinates has been implemented [AlQuraishi, 2019]. With the improvements introduced in EncoderMap(II), EncoderMap's unique ability to visualize important conformational changes, identified during the dimensionality reduction process, is now also available for large chain-like molecules. In [Lemke et al., 2019] we demonstrated this with the example of two multidomaine proteins: a Ubiquitine dimer and a part of the Ssa1 Hsp70 yeast chaperone.

So far, we have separately looked at simplified molecular representations, for saving computational cost and for analysing overwhelming amounts of data. The work by Hunkler et al. [Hunkler et al., 2019] shows how both can be combined in a unified framework. Our previously introduced neural network coarse-graining approach was used to obtain an ensemble of conformations in the coarse-grained space. Then, dimensionality reduction was used to obtain a two-dimensional map of the coarse-grained space, which is a task, our previously introduced dimensionality reduction method, EncoderMap is well suited for. Finally atomistic simulations were started from different clusters identified in the map to quickly obtain an extensive ensemble of possible atomistic conformations. With these different levels of resolution, we can profit from a detailed sampling in the atomistic space, a fast sampling in the coarse-grained space, and we can monitor and guide everything with maps obtained by dimensionality reduction.

Publications

Four publications are included in this thesis. The detailed author contributions to each publication are listed below in categories inspired by Brand et al. [Brand et al., 2015].

Lemke, T., Peter, C., and Kukharenko, O. (2018). Efficient sampling and characterization of free energy landscapes of ion–peptide systems. ***J. Chem. Theory Comput.***
Conceptualization: CP, OK; Methodology: TL, OK; Simulations: TL; Analysis: TL, OK ; Visualization: TL, OK ; Writing – Original Draft: TL, OK; Writing – Review & Editing: CP, TL, OK; Supervision: CP; Funding acquisition: CP, OK.

Lemke, T. and Peter, C. (2017). Neural network based prediction of conformational free energies-a new route toward coarse-grained simulation models. ***J. Chem. Theory Comput.***
Conceptualization: TL, CP; Methodology: TL; Simulations: TL; Software Development: TL; Analysis: TL; Visualization: TL; Writing – Original Draft: TL; Writing – Review & Editing: CP; Supervision: CP; Funding acquisition: CP.

Lemke, T. and Peter, C. (2019). Encodermap: Dimensionality reduction and generation of molecule conformations. ***J. Chem. Theory Comput.***
Conceptualization: TL, CP; Methodology: TL; Simulations: TL; Software Development: TL; Analysis: TL; Visualization: TL; Writing – Original Draft: TL; Writing – Review & Editing: CP; Supervision: CP; Funding acquisition: CP.

Lemke, T., Berg, A., Jain, A., and Peter, C. (2019). Encodermap (ii): Visualizing important molecular motions with improved generation of protein

conformations. ***J. Chem. Inf. Model.***

Conceptualization: TL, CP; Methodology: TL; Simulations: AB, AJ; Software Development: TL; Analysis: TL; Visualization: TL; Writing – Original Draft: TL; Writing – Review & Editing: CP, AB, AJ; Supervision: CP; Funding acquisition: CP.

JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. 2018, 14, 5476−5488

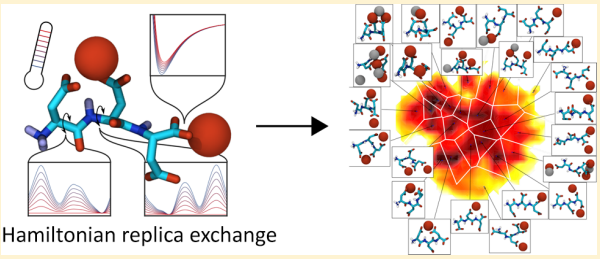
pubs.acs.org/JCTC

Efficient Sampling and Characterization of Free Energy Landscapes of Ion−Peptide Systems

Tobias Lemke, Christine Peter,* and Oleksandra Kukharenko*

Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Germany

ABSTRACT: Proteins that influence nucleation, growth, or polymorph selection during biomineralization processes are often rich in glutamic- or aspartic acid. Here, the interactions between carboxylate side chains and ions lead to an interplay of peptide conformations and ion structuring in solution. Molecular dynamics simulations are an ideal tool to mechanistically investigate these processes. Unfortunately, the formation of strong ion-peptide contacts and ion bridges drastically impedes structural reorganization of ionic bonds and conformational transitions of the polymers. Thus, to obtain a complete thermodynamical picture of such systems, enhanced sampling techniques become necessary as well as the methods to characterize the conformational states of these partially disordered polymer-ion systems. Here, we propose a new set of Hamiltonian replica exchange (HRE) parameters for efficient simulations of peptide−ion systems, with an aspartic acid trimer in the presence of Ca^{2+} and Cl^- ions as a test system. We introduce dimensionality reduction and clustering strategies to characterize the states of such a multicomponent system and to analyze the outcome of the proposed HRE with different reweighting methods.



1. INTRODUCTION

Biominerals often exhibit outstanding mechanical properties outperforming pure mineral phases. These unique properties arise from the complex hierarchical structure of biominerals. To create such complex structures, crystal morphologies have to be controlled, polymorphs have to be selected, and crystal orientations have to be set.[1] Some of the main components of biominerals besides the mineral itself are macromolecules which appear to be the key for understanding control mechanisms in organisms responsible for the formation of such complex structures. Macromolecules isolated from calcium carbonate biominerals typically contain parts rich in acidic amino acids such as aspartic and glutamic acids.[2] The influence of those amino acids on mineral formation can be explained with the ability to bind to Ca^{2+} ions with the negatively charged carboxylate group in the side chain.[3]

While analyzing the macroscopic result of crystallization is relatively straightforward, gaining insight into the early stages of crystallization through experiment is difficult and often indirect.[4] The small size of these early stage structures, which is challenging for many experimental techniques, is an advantage for molecular dynamics (MD) simulations. MD has been very usefully applied to investigations of systems prior to nucleation.[5−7] Some of these MD studies suggest that structural motives which are later found in the crystal might already be present in solution before nucleation sets in.[8−11]

Given available computational resources there are still severe limitations in size- and time-scales accessible by MD simulations which limits the ability to study slow processes and rare events. Both mineral nucleation and biomolecular folding processes are rare events. Thus, the interplay of peptide conformations, the formation of ionic prenucleation species, and ultimately mineral formation are an extremely difficult task for atomistic MD simulations. This can be already seen with seemingly simple systems such as acidic macromolecules in contact with ion solutions. For example, an aspartic acid rich peptide which is in the presence of Ca^{2+} ions exhibits multiple types of barriers involving the rotation around the peptide backbone and the formation of very stable ion-side chain contacts. The stable ion bridges formed between two or more carboxylate groups and a Ca^{2+} ion are − once formed − unlikely to open up again during a regular MD simulation, so that the system is arrested in metastable, conformational states. Thus, to overcome these barriers and to be able to explore the whole conformational space, advanced sampling techniques are needed.

Enhanced sampling methods such as parallel[12] or simulated[13] tempering, metadynamics,[14] umbrella sampling,[15] or replica exchange molecular dynamics (REMD)[16] and others[17−21] perform simulations at different Hamiltonians or temperatures to promote rare events. One of the most commonly used techniques to overcome energetic barriers is temperature REMD. However, Bulo et al.[22] showed that ion bridges become even stronger at higher temperatures as they are dominated by entropic influences. A more versatile tool to overcome such barriers is the Hamiltonian replica exchange (HRE), which allows to play with different temperatures and furthermore to specifically manipulate the interaction potentials which are the origin of the impeding barriers.

Received: June 7, 2018
Published: September 21, 2018

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

Kahlen et al.[8] applied HRE to simulations of glutamic acid oligomers in the presence of Ca^{2+} ions in solution. To promote the conformational sampling, biasing potentials were applied to the peptide backbone dihedrals. This resulted in flattened backbone potentials allowing Kahlen et al. to successfully equilibrate the structural ensemble of glutamic acid oligomers in the presence of Ca^{2+} ions and draw conclusions regarding the structures during early aggregation stages.

Besides glutamic acid, the structurally very similar aspartic acid is often used in experiments to influence the crystallization process.[23−25] Unfortunately, the transfer of the simulation approach proposed by Kahlen et al. to other cases where ion bridges arrest the conformational sampling of a peptide turned out not to be straightforward. Application of the HRE settings of glutamic acid oligomers with Ca^{2+} ions to aspartic acid oligomers did no result in a similarly improved sampling. Hence a new set of HRE parameters had to be determined, and the enhanced sampling strategy of Kahlen et al. had to be further refined to make the prenucleation stage in the presence of aspartic acid additives also accessible to MD simulations. Here, we present HRE settings that are optimized for an efficient simulation of aspartic acid peptides in the presence of ions − with an aspartic acid trimer in an aqueous solution with Ca^{2+} and Cl^{-} ions as an example system. In the proposed HRE scheme biasing the backbone dihedral angles, weakening the Ca^{2+}-side chain interactions and increasing the temperature were combined to loosen the ion bridges and speed up the peptide's conformational transitions.

Furthermore, we apply different reweighting techniques to explore how to most efficiently use the exploration of sampling of phase space by all replicas for such a system. To be able to apply reweighting techniques and to judge the quality of both sampling and reweighting, a good definition of the conformational states of the system was needed. Most importantly, such a characterization of conformational states can be subsequently used in kinetic modeling and in the comparison of different peptides and the interplay of peptide conformations and prenucleation ion structures. First we describe the conformational space of the peptide itself, focusing on its structural changes. The dimensionality reduction technique sketch-map[26,27] is used to project the conformational space to two dimensions. Additionally, we use an ion bridge-based clustering, which divides the configuration space based only on the information about ion bridges formed between side chain and terminal residues through connection with Ca^{2+} ions. We compare these techniques and combine both conformational and ion−peptide interaction information for a finer separation of the conformation space. For the estimation of the free-energies of the so-obtained states we compare the standard "direct counting" in the lowest Hamiltonian with different reweighting approaches: the multi-state Bennett acceptance ratio (MBAR)[28] reweighting takes the information about the population of the respective states in the higher Hamiltonians into account, while the recently proposed transition-based reweighting analysis method (TRAM)[29] additionally includes information about transitions between the states. Finally we discuss the efficiency of combining the proposed HRE scheme, clustering methods, and reweighting techniques to study peptide−ion systems.

2. METHODS

2.1. Simulation Details. The Gromacs 4.6.7 simulation package[30,31] was used with a modified version of the GROMOS 54A7[32] force field and the SPC/E water model.[33] To accurately describe the interactions between calcium ions and the carboxylate oxygen atoms in the glutamate and aspartate side chains, the force field was modified regarding the calcium−oxygen Lennard-Jones parameters. The original values were replaced by the parameters shown in Table 1

Table 1. Lennard-Jones Parameters for the Calcium−Oxygen Interaction

	C_6 [kJ nm^6 mol^{-1}]	C_{12} [kJ nm^{12} mol^{-1}]
GROMOS 54A7	0.00150765	2.16509×10^{-06}
used parameters	0.00203533	1.53721×10^{-06}

which were determined following the methods described in ref 8. The parameters were tuned to match experimental association constants of calcium and carboxylate groups as well as reference data from first-principles molecular dynamics simulations. The temperature was kept constant at 300 K (at least for the lowest replica) with stochastic velocity rescaling by Bussi et al.[34] The pressure was not readjusted during the main simulation. In preceding equilibration simulations a Berendsen barostat[35] was used to bring the simulation to 1 bar. The leapfrog algorithm with a time step of 2 fs was used to integrate the equations of motion. Long range interactions were calculated with the particle mesh Ewald method[36] with a grid spacing of 0.12 nm and a pme-order of 4. Both Coulomb and Lennard-Jones interactions were truncated at 1.4 nm. Bonds have been constrained using the linear constraints solver (LINCS) algorithm[37] with an eighth order expansion.

2.2. Optimized HRE Parameters for Peptide−Ion Systems. The HRE simulations were performed with the PLUMED2.1-hrex[38] plug-in for Gromacs. Eight replicas were used, and exchange attempts between neighboring replicas were made every 2 ps [In this paper we use "replica" to describe all the simulation that has been done at a given Hamiltonian; we use "trajectory" whenever we talk about a consecutive molecular dynamics trajectory in phase space which is traveling through the different Hamiltonians during the HRE.]. A system with 1 aspartic acid trimer, 10 calcium ions, 17 chloride ions, and 3536 water molecules was simulated. In total this adds up to a neutral charge as all carboxylate groups were deprotonated and the N-terminal amino group was protonated. Three different changes were made to the Hamiltonian in higher replicas: (1) the potential of the backbone dihedrals was flattened by adding a bias potential, (2) the Ca−O interaction was weakened, and (3) the temperature was increased. An illustration of the changes is shown in Figure 1.

In order to flatten the rotational barrier around the backbone torsional angles by adding a biasing potential, the unbiased potential of mean force for rotation around the backbone torsions was calculated. Following the process proposed by Kahlen et al.[8] a central amino acid of an uncharged aspartic acid pentamer was used to determine the general rotational barrier, which makes it applicable to homo-oligomers and to be representative of the intrinsic peptide rotational barrier − i.e. not accounting for the influence of strong electrostatic interactions between side chains. With the dihedral angles Φ and Ψ of the central amino acid as collective variables, 2D well-tempered metadynamics[39] was performed. Two-dimensional Gaussians were deposited with a starting height of 1.2 kJ/mol and a width sigma of 10° every 120 fs. A

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

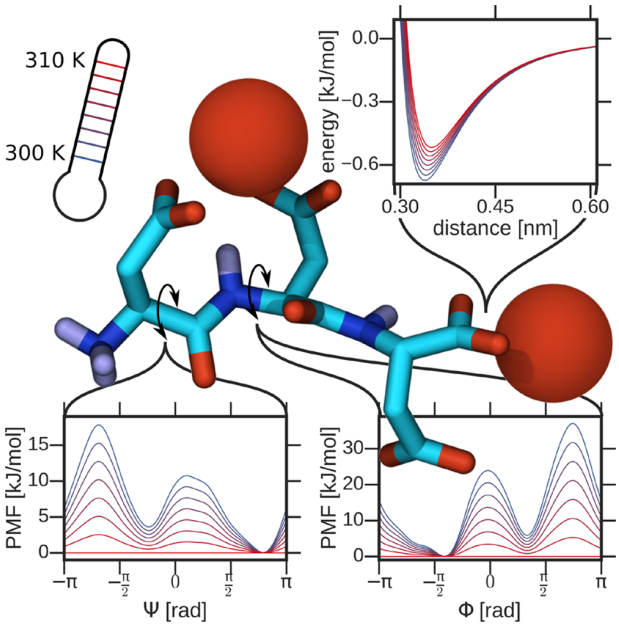


Figure 1. Illustration of the different changes made in higher replicas: the potential of the backbone dihedrals was flattened by adding a bias potential, the Ca−O interaction was weakened, and the temperature was increased. The color changes from blue to red showing the Hamiltonian changes made from the lowest to the highest replica.

bias factor of 5 was chosen corresponding to a ΔT of 1200 K. By averaging the obtained 2D free energy surface along the Φ and Ψ axis, respectively, 1D biasing potantials were obtained. These biasing potentials were implemented as tabulated potentials, scaled by a factor α which was varied from 0 to −1 in equal intervals (see Table 2), and added to the regular

Table 2. Parameters That Were Varied in the Replicas of the HRE Simulations

replica	T [K]	α (see text)	C_6 [kJ nm^6 mol^{-1}]	C_{12} [kJ nm^{12} mol^{-1}]
7	310.5	−1.00	1.88×10^{-03}	1.72×10^{-06}
6	309.0	−0.86	1.91×10^{-03}	1.69×10^{-06}
5	307.5	−0.71	1.93×10^{-03}	1.67×10^{-06}
4	306.0	−0.57	1.95×10^{-03}	1.64×10^{-06}
3	304.5	−0.43	1.97×10^{-03}	1.61×10^{-06}
2	303.0	−0.29	1.99×10^{-03}	1.59×10^{-06}
1	301.5	−0.14	2.01×10^{-03}	1.56×10^{-06}
0	300.0	0.0	2.04×10^{-03}	1.54×10^{-06}

dihedral potentials in the respective replicas. The two graphs at the bottom of Figure 1 illustrate the respective resulting dihedral free energy barriers (PMFs) for both backbone torsions in the different replicas going from blue to red. The application of this backbone dihedral bias ensures fast sampling of conformational transitions for a peptide in water (see Figure 2).

However, as soon as ions and charges are added to the system, new high energy barriers emerge − most importantly due to the formation of ion bridges. To overcome those barriers, the interaction strength between calcium ions and carboxylate oxygens was weakened in the higher replicas by altering the C_6 and C_{12} parameters of the corresponding Lennard-Jones potentials. The graph in the top right corner of Figure 1 illustrates the potentials used for the different replicas.

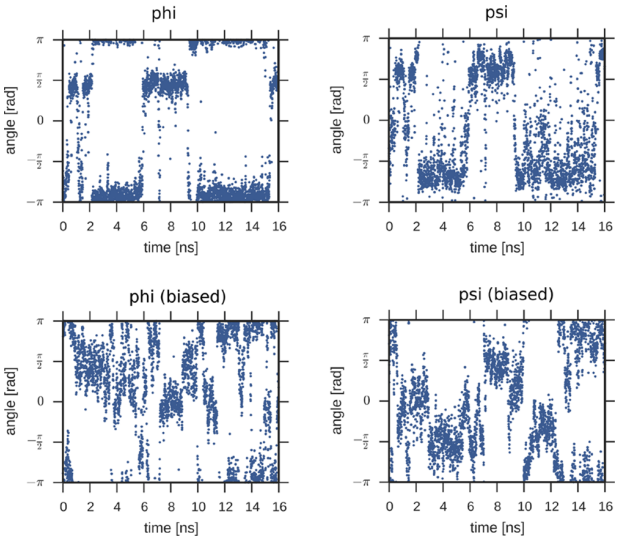


Figure 2. Torsion angles phi and psi of the central amino acid of an aspartic acid pentamer. In the top row the torsions are shown for an unbiased simulation. In the bottom row the same is shown for a simulation where the torsions were biased as described in section 2.2 with $\alpha = -1$.

The conformational sampling is not only dependent on the peptide itself and the ions attached to it. Processes such as reorganization of solvent molecules and diffusion of ions from and to the peptide certainly play a key role as well. To accelerate these we increased the temperature from 300 K for the lowest to 310.5 K for the highest replica. A large temperature increase would be counterproductive, since it might even further stabilize ion bridges due to their entropy driven formation.[8]

All the parameters, that were used for the HRE, are summarized in Table 2.

2.3. State Characterization. Even the rather small peptide−ion system investigated here exhibits a highly complex conformational free energy landscape. We have applied three different methods to characterize states and structural transitions. These clustering approaches focus on different features of the system and can be complementary to each other. One method is based on the ion bridges, a second is based on the peptide conformation, and a third is taking both the peptide and the ion positions into account. Figure 3 shows illustrations of the different state characterization methods which in the following will be used as labels for these methods.

2.3.1. Ion Bridge-Based State Characterization. First we used the number and position of ion bridges as input for a clustering approach. This is done in the following way: All carboxylate C atoms present in a sphere with a radius of 0.5 nm around one calcium ion are considered as connected. This connection information can be represented in a matrix of booleans. An example structure, the founded connections, and the subsequent matrix are shown in Figure 4.

Of course, half of the matrix is sufficient, as it is symmetric along the diagonal. Structures with the same matrices are considered to belong to the same cluster. All structures in simulations are analyzed and assigned to the corresponding cluster. The total number of all possible clusters is $2^{N(N-1)/2}$, where N is a number of side chains plus C-terminus. This clustering does not account for the structural changes in a

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

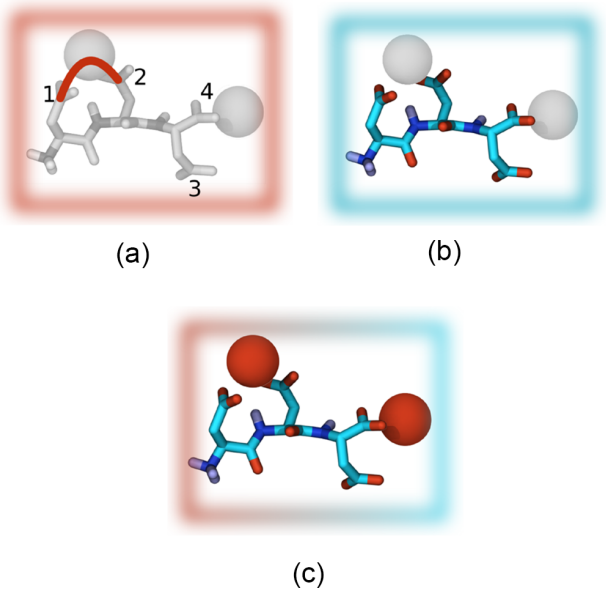
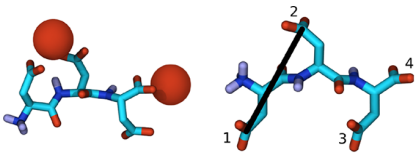


Figure 3. Three different state characterization methods are used in this work (a) based on ion bridges, (b) based on peptide conformations, and (c) combining peptide conformation and ion position information.



	1	2	3	4
1		x		
2	x			
3				
4				

Figure 4. Example structure on the left with the connections shown on the right, found by the developed algorithm. The connection information is stored in matrices like the one on the far right.

peptide explicitly and cannot differentiate between structures with no ion bridges. If one is interested in the conformational changes, the next clustering method will be advantageous.

2.3.2. Conformation-Based State Characterization. To characterize the peptide conformations, we first used only the information about its structural changes without any consideration of ion positions. As high-dimensional descriptors (collective variables (CVs)) the pairwise distances between all C_α and C_γ atoms were chosen, resulting in $\frac{K(K-1)}{2}$ dimensions, where K is the number of C_α and C_γ atoms. For the tripeptide under consideration, this results in 15 dimensions. As direct clustering of high-dimensional data faces a number of challenges, we first reduced the dimensionality of CVs, which is the natural step in many cases. It has been found that one of the most common dimensionality reduction algorithms – principal component analysis (PCA)[40] – can be thought of as a continuous solution of the cluster membership indicators in the commonly used K-means clustering method.[41] This shows that a connection between clustering methods and dimensionality reduction approaches exists,[42] suggesting that dimensionality reduction may be a natural extension to the clustering method and thereby provides a continuous approach to estimate densities of high-dimensional data from small samples.[43]

Here, we used a nonlinear dimensionality reduction method based on multidimensional scaling techniques called sketch-map.[44] It was shown to be successful in distinguishing systems with a vast amount of possible configurations.[44,45] Sketch-map iteratively optimizes the low-dimensional projections to minimize the dissimilarity between sigmoid functions of distances between points in high- and low-dimensional spaces

$$s(r, a, b) = 1 - (1 + (2^{a/b} - 1)(r/\sigma)^a)^{-b/a} \quad (1)$$

where r is the Euclidean distance in high- or low-dimensional space. In this Article we used the following parameters for the sigmoid function: $\sigma = 0.25$, $A = 15$, $B = 4$ (in high-dimensional space), $\sigma = 0.25$, $a = 2$, $b = 4$ (in low-dimensional space). An example for the obtained (2D) projection can be seen in Figure 7a. Next clusters (dens regions) should be identified in the resulting projection to characterize conformational states. Details about subsequent clustering of sketch-map projections are described in Sec. 2.3.4.

2.3.3. Combining Conformational and Ion Bridge Information. To obtain a two-dimensional representation which combines both information about the peptide conformation and the peptide–ion coordination, we chose a new set of CVs: in addition to the 15 pairwise distances (see Sec. 2.3.2) we calculated all distances between the peptide and ions, more specifically, between Ca^{2+} and Cl^- ions on the one hand and the carboxylate C atoms as well as the (negatively charged) C-terminus of the peptide on the other hand (distances between Ca^{2+} and Cl^- ions are not included). We obtained 108 distances in total for each frame. We are interested only in the ions interacting with the peptide and exclude all others. To do so, distances larger than a cutoff (0.5 nm as in Sec. 2.3.1) were set to zero. Not all of the remaining values add new information to a description of the peptide–ions conformation, meaning in the case of a compact structure, we will have many nonzero distances out of 108, but they will be correlated. We used PCA[40] to get an idea of the number and type of distances to include. For this system it was sufficient to keep the 9 longest distances, while the short ones were highly correlated. Intuitively, we could have simply used the same number of distances as in ion bridge-based clustering in Sec. 2.3.1, but it would result in more dimensions, mostly containing zero values. Eventually, the 9 longest ionic distances (sorted from lowest to highest) were added to the 15 conformational CVs (see Sec. 2.3.2), resulting in 24 CVs in total. We reduced the dimensionality to two dimensions using the sketch-map algorithm with the following parameters of the sigmoid function (see eq 1): $\sigma = 0.25$, $A = 24$, $B = 3$ (in high-dimensional space), $\sigma = 0.25$, $a = 2$, $b = 3$ (in low-dimensional space). The obtained projections can be seen in Figures 7b and 10a. We found that this method was very well suited to distinguish between different states of the combined peptide–ion system: regions with different numbers of ion contacts are nicely separated going from no ions in contact with the peptide (left-hand side of the projection) to the maximum number of ion contacts (right-hand side). In addition, a separation of different peptide conformations is achieved inside those regions. A detailed comparison of the states that can be identified in the different projections will be presented in Sec. 3.2.

2.3.4. Clustering of Projections. Both types of two-dimensional projections described above were clustered using the following scheme: first the minima in the free energy surface were automatically detected by binning the two-

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476–5488

dimensional projection to rectangles (cells) and selecting iteratively local maxima (most populated cells, separated by change in the density gradient); they were selected as cluster centers, and all other points were assigned to the closest center. This scheme is similar in spirit with density peak clustering.[46] While the latter is in principle more general because it is independent from the number of dimensions, we used the scheme proposed here for two reasons: In the case of a two-dimensional projection of a vast number of data points, binning the points and determining local density peaks are more efficient compared to the calculation of a distance matrix for all points (as required by density peak clustering). The second reason is that it does not require input such as a cutoff distance, an a priori definition of a desired number of clusters, or a minimum density; but contrary to density peak clustering, the here proposed scheme is sensitive to the choice of bin sizes. The number of cluster centers and cluster boundaries can be regulated (if needed) by either providing a maximum number of clusters and/or by defining the minimum number of points in a cell for this cell to be considered as a density peak. If the number of defined peaks is greater than the desired number of clusters, smaller clusters will be iteratively aggregated with the nearest larger one (i.e., containing more points) to obtain the desired number of states (see an example for such an aggregation in Figure 7b).

2.4. Reweighting. Different estimation techniques are used to extract the information of interest (such as free energy differences between states) from the multiensemble simulation data generated using HRE. The simplest estimate is "direct counting".[47] It is done by computing ensemble averages of properties and populations of different states based on the sampling of a Hamiltonian of interest (relying on a proper, e.g. canonical, sampling at this Hamiltonian/in this replica). The probability density is inferred from this histogram and used to calculate the free energy. This approach is the foundation for the "direct counting" potential of mean force (PMF) calculations.

This method served as a basis to a more sophisticated and widely used technique − the weighted histogram analysis method (WHAM)[48−50] − which allows use of information from all replicas (as opposed to the direct counting scheme, where only the data generated at the Hamiltonian of interest is used). WHAM requires the assignment of each structure to a state and to a bias value, which was used in the simulation of this structure. Most commonly several simulations with different bias values are run, and some low-dimensional (1 or 2 dimensions) reaction coordinate is chosen for all simulations. The distributions of the reaction coordinate values from all simulations are clustered (e.g., binned). The obtained histogram is reweighted taking into account the biasing factors. This is done by solving the following two equations:

$$p_i = \frac{\sum_{k=1}^{K} n_i^{(k)}}{\sum_{k=1}^{K} n^{(k)} f^{(k)} \exp(-b_i^{(k)})} \tag{2}$$

$$f^{(k),\mathrm{new}} = \frac{1}{\sum_{j=1}^{N_s} \exp(-b_i^{(k)}) p_j} \tag{3}$$

where $n_i^{(k)}$ is the number of structures in a state (bin) i at the thermodynamic state k, $n^{(k)}$ is the total number of structures simulated at the thermodynamic state k, K is the total number of thermodynamic states, N_s is the total number of bins/states, $b_i^{(k)}$ is the reduced bias energy of structures in bin i at the thermodynamic state k, p_i is the best estimate of unbiased stationary probabilities, $f^{(k)}$ is the dimensionless free energy shift from simulation at the thermodynamic state k; and p_i and $f^{(k)}$ are unknown and should be found by e.g. iteratively solving these equations (self-consistent iterations).

For systems, where a biasing potential cannot be trivially scaled by one linear force field parameter (e.g., temperature), the number of bins is significant even for a small number of conformational states, making the application of WHAM computationally intractable. To overcome this limitation and to avoid a possible bias introduced by binning of the energy histogram,[51] the multistate Bennett acceptance ratio (MBAR)[28] was introduced. MBAR is equivalent to bin-less WHAM or UWHAM.[48,52] Its free energy estimating equation is

$$f^{(k),\mathrm{new}} = -\ln \sum_{i=1}^{K} \sum_{j=1}^{n^{(i)}} \frac{\exp(-b^{(k)}(x_j^{(i)}))}{\sum_{l=1}^{K} n^{(l)} \exp(f^{(l)} - b^{(l)}(x_j^{(i)}))} \tag{4}$$

where $x_j^{(k)}$ is the jth configuration sample from a thermodynamic state k (all other variables are the same as in eq 2). UWHAM and MBAR provide estimates for statistical uncertainties in free energy differences and their correlations. In some applications histogram-based methods can require less computational resources than MBAR or UWHAM at the expense of introducing bias caused by binning.

The above-mentioned estimators produce statistically optimal estimates of free energy surfaces only if the simulations have sampled the full canonical ensemble corresponding to their respective thermodynamic states/Hamiltonians, i.e. are in global equilibrium. This is rarely fully correct for the systems to which HRE techniques are usually applied. To correct for this problem, transition-based reweighting methods were introduced. They combine the maximum likelihood approach of WHAM and MBAR with the Markov model theory[53] by taking into account not only counts inside states but also information about transition probabilities between them. They can roughly be divided into histogram-based (the discrete transition-based reweighting analysis method (dTRAM)[54] and the dynamic histogram analysis method (DHAM)[55]) and bin-less estimators (the expanded (nonmaximum likelihood) (xTRAM)[56] and the general transition-based reweighting method (TRAM)[29]). These methods present an option to obtain accurate free energy estimates as well as kinetic information for correlated samples under the condition of local equilibrium (which is not always easy to meet).

It was suggested that TRAM is the statistically optimal estimator and is superior to global equilibrium-based estimators such as WHAM or MBAR and other transition-based estimators (xTRAM, dTRAM, DHAM),[29] because it can give correct statistical and thermodynamic estimates without relying on the global equilibrium assumption. This is done by taking into account the information about transitions between configuration states (clusters) at the appropriate lag time (time at which the observed transitions became independent/uncorrelated). TRAM equations, which can also be solved iteratively, are

$$\nu_i^{(k),\mathrm{new}} = \nu_i^{(k)} \sum_{j=1}^{N_s} \frac{c_{ij}^{(k)} + c_{ji}^{(k)}}{\exp(f_j^{(k)} - f_i^{(k)})\nu_j^{(k)} + \nu_i^{(k)}} \tag{5}$$

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

$$f_i^{(k),\mathrm{new}} = -\ln \sum_{j=1}^{n_i} \frac{\exp(-b^{(k)}(x_j))}{\sum_{l=1}^{K} R_i^{(l)} \exp(f_i^{(l)} - b^{(l)}(x_j))} \quad (6)$$

where

$$R_i^{(k)} = \sum_{j=1}^{N_s} \frac{(c_{ij}^{(k)} + c_{ji}^{(k)})\nu_j^{(k)}}{\exp(f_i^{(k)} - f_j^{(k)})\nu_i^{(k)} + \nu_j^{(k)}} + n_i^{(k)} - \sum_{j=1}^{N_s} c_{ij}^{(k)} \quad (7)$$

where $\nu_i^{(k)}$ are Lagrange multipliers, n_i is the number of all samples in configuration state i independent from the thermodynamic state k, $c_{ij}^{(k)}$ is the number of transitions at the thermodynamic state k from configuration state i at time t to state j calculated at time $t + \tau$, and $\tau > 0$ is the lag time.

We found that a system proposed here and simulation setup is a nice test case for the applicability of TRAM. The main motivations for us to choose this method are as follows: TRAM does not require binning of the bias energies and is therefore suitable for the analysis of multitemperature simulations in combination with bias potentials, and it requires data to be only in local equilibrium. The calculation of the estimates was performed using the PyEMMA software.[57]

3. RESULTS AND DISCUSSION

3.1. Simulation with the New HRE Parameters. The system consisting of an aspartic acid tripeptide with Ca^{2+} and Cl^- ions in water (see Sec. 2) is used to evaluate the utility of HRE simulations with the new parameters and to demonstrate clustering approaches. For a successful HRE mainly two requirements have to be met: (1) the highest replica has to sample the phase space quickly and (2) trajectories have to frequently travel through the whole range of Hamiltonians, i.e. there has to be a frequent exchange between neighboring replicas, and no separation of replicas must occur.

First, we tested if the highest replica is able to sample phase space quickly. We ran two separate simulations with the parameters of the lowest (no bias) and the highest replica respectively starting from the same structure. Figure 5 illustrates how successful this set of parameters speeds up sampling in the highest replica by monitoring Ca-peptide contacts. Steady contacts indicate slow sampling, while frequently changing contacts indicate fast sampling. While the lowest replica simulation is stuck in very similar structures, the highest replica simulation is able to overcome this initial structure and to quickly sample many different structures.

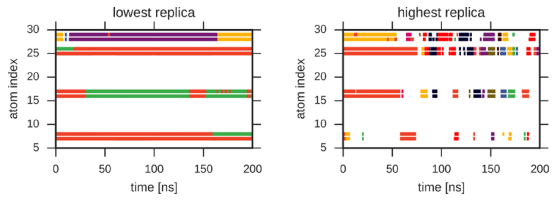


Figure 5. Calcium oxygen contacts over time. Two separate simulations with the parameters corresponding to the highest and the lowest replica were performed. Each of the 10 calcium ions present in the simulation has its own color which is displayed whenever this ion has contact with an oxygen atom of the peptide. Two atoms are considered to be in contact when their distance is smaller than 0.3 nm. The peptide atoms are indexed starting from the amino end. This atom index is shown on the *y*-axis. Note the pattern of double lines which shows that an ion is typically in contact with both oxygen atoms of a carboxylate group.

Then we performed extensive 1 μs long HRE simulations with 8 replicas. To evaluate how frequently the trajectories are traveling through the different Hamiltonians, the Hamiltonian index over time was plotted for each trajectory (the right-hand side of Figure 6). To get further insight on the sampling taking place, ion bridge-based state characterization (see Sec. 2.3.1) was performed. Our system allows for in total 64 different combinations of pairwise ion bridges, meaning 64 different clusters. We selected the 10 most populated clusters, and all other structures were considered to be in the cluster number 0. The cluster which contains structures with no ion bridges at all is the most populated cluster and therefore has the cluster number 1.

The time series plots in Figure 6 are colored according to the 10 most occurring ion bridge connection patterns (see images in the top row). Most trajectories frequently travel through the whole range of Hamiltonians. These trajectories also frequently move from one cluster to another. Figure 6 also shows the fraction of time spent in different clusters for each trajectory. Most of the structures can be found in all trajectories, indicating a HRE being near convergence. Trajectory 1 behaves a little different compared to the others. It stays in cluster 4 (yellow) for around 500 ns. During that time the trajectory no longer travels through the whole range of Hamiltonians but stays in the lowest replicas. This behavior indicates that a very stable structure was found here. The question is why is it so hard for the HRE to leave this structure. As we will later find in the other characterization methods, there are structures which are not only stabilized by single ion bridges but rather by a small cluster of calcium and chloride ions. This stable cluster 4 mainly contains such structures. It has a large overlap with cluster 1 of the conformation-based clustering (see Table 3 and Figure 7a) or Region II in Figure 10a. The HRE parameters were developed with slow peptide dynamics and stable ion bridges in mind. Figures 5 and 6 show that the selected parameters successfully overcome these issues, but apparently this set of parameters does not work equally well for structures with calcium chloride clusters. There is no parameter change directly targeted to break up such clusters. A single carboxylate group might detach from the ion cluster due to the weakened Lennard-Jones potential for Ca–O interactions. However, as long as the calcium chloride cluster does not dissolve, the carboxylate group will soon reattach. It is very unlikely that all carboxylate groups detach from the cluster at the same time.

In contrast to temperature replica exchange, HRE allows to focus on very specific interactions. This example shows that HRE can be used very efficiently to overcome free energy barriers. However, this example also gives a warning that it is always possible that new unexpected constellations appear where the initially targeted interactions are no longer those preventing the system from further sampling.

In the subsequent analysis we use a reweighting scheme which allows us to use information from higher replicas and give more accurate probability (free energy) estimates for the explored states even if the sampling may still have not reached full global convergence.

3.2. Analysis and Free Energy Estimates. To apply reweighting methods it is inevitable to address the question of how to define states (e.g., by binning of the potential energy or some reaction coordinate/collective variable, by clustering of

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

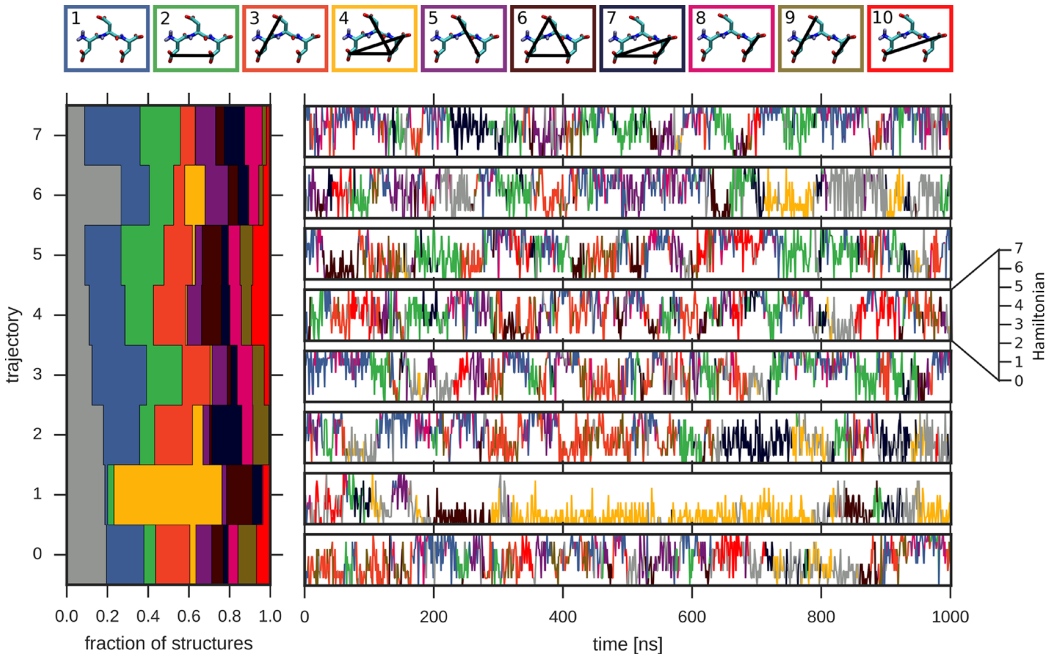


Figure 6. Overview of the sampling of peptide−ion states visited during the HRE MD simulations. In the top row the 10 most frequently occurring ion bridge patterns of all 8 trajectories are shown. On the left the fraction of the structures sampled by the different trajectories is shown (color code according to the frames around the 10 structures, all other ion bridge patterns are represented in gray). On the right a temporal trace of the traveling of each trajectory through the 8 Hamiltonians is shown. The color of the line plots is varied according to the ion bridge patterns.

Table 3. Similarities in Ion Bridge-Based (See Sec. 2.3.1) and Conformation-Based (see Sec. 2.3.2) Characterization of Structures[a]

Conf. based clusters \ Ion-bridge	0	1	2	3	4	5	6	7	8	9	10
0	154986	682	14016	16338	39638	1282	225595	27339	132	2253	55
1	59493	1844	1986	3401	326956	27234	6508	3311	594	1658	643
2	31916	20424	365599	843	1058	652	903	40473	2999	215	941
3	18984	10154	101425	627	619	284	261	161496	1586	83	105
4	7909	56246	696	215604	1338	591	126	92	18468	81297	506
5	97837	38603	212	829	334	138932	161	42	3735	155	1810
6	18200	128833	610	578	217	1150	97	54	45510	170	1918
7	111522	2944	815	2233	3634	10183	404	577	715	1169	13056
8	13880	6175	103	83513	592	209	166	22	2099	60807	2573
9	9198	985	139	90156	880	447	2546	154	261	24821	380
10	31251	21974	248	1066	374	407	37	603	7245	589	141071
11	16567	143394	18287	1121	198	2017	691	61	16994	256	214
12	8556	155549	582	3905	217	708	81	25	58096	1392	1183
13	8467	4523	57	148	58	91	6	163	3088	34	27909
14	2118	1957	460	68	303	52463	1864	307	292	24	5
15	17084	9085	35	81	77	21474	33	10	1129	39	23
16	2071	188	494	11501	169	10	15682	56	15	1487	36
17	3271	26804	107	152	70	215	27	10	9536	35	29
18	681	14724	259	17595	279	54	30	28	2280	2131	66
19	1183	63863	136	2614	79	166	27	23	30046	1673	27
20	43	6027	17	17	7	53	2	0	964	3	8
21	15	617	8	107	6	6	2	3	322	155	8
22	243	4008	14	13	14	495	3	14	618	2	22

[a]Columns represent the ion bridge-based clusters. Columns 1 to 10 represent the ten most probable clusters. All remaining structures are combined in cluster 0. The rows represent conformation-based clusters where the largest numbers of structures are highlighted in gray.

high-dimensional data, etc.). In Sec. 2.3 characterization strategies for obtaining an optimal description of states were presented. The results of their application are shown in this section.

3.2.1. Obtaining Thermodynamic Estimates Using Peptide Conformation- and Ion Bridge-Based Clustering Separately. First we used the conformation-based characterization approach (see Sec. 2.3.2). All structures of all replicas

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

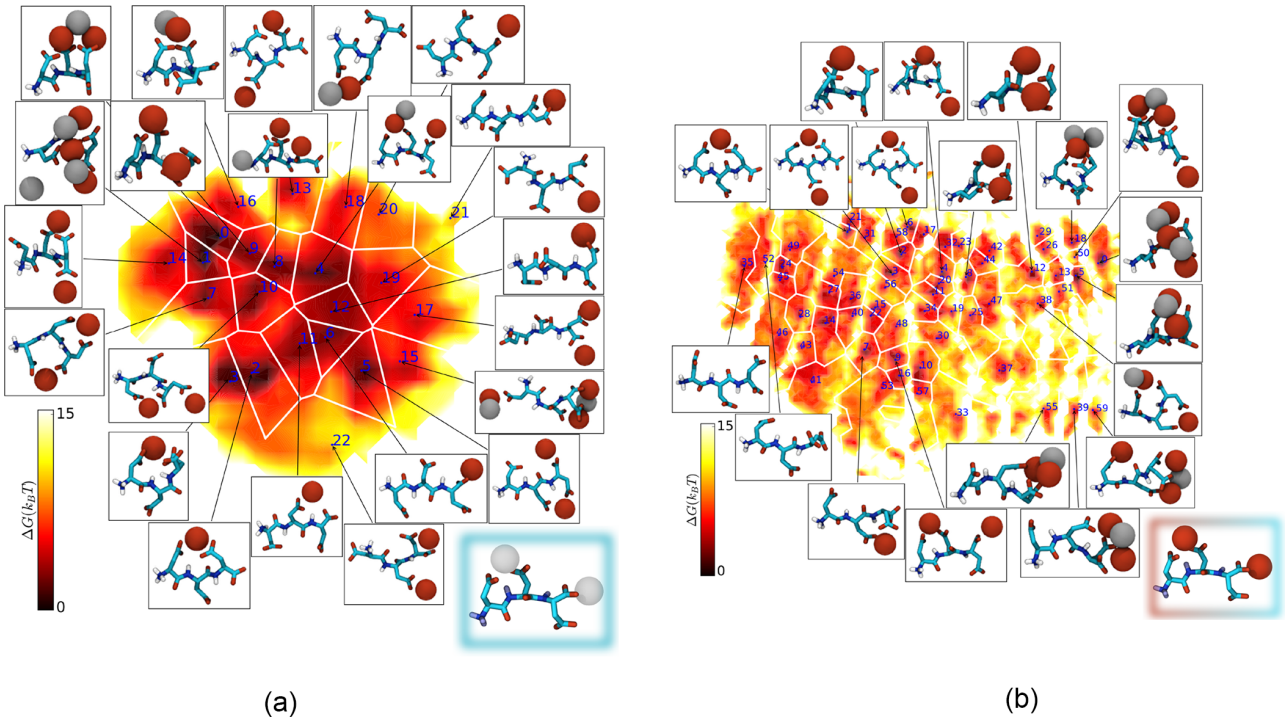


Figure 7. Conformation phase spaces sampled by all 8 replicas in 1 μs of simulation time in each replica. The cluster boundaries are the white lines; the cluster numbers and some representative structures with ions (calcium and chloride ions are red and gray, respectively) are shown. The projection was made using the sketch-map algorithm. The following collective variables were used: (a) distances between C_α and C_γ atoms; (b) distances between C_α and C_γ atoms and the nine largest distances within a cutoff from the C_γ atoms and the C-terminus to ions.

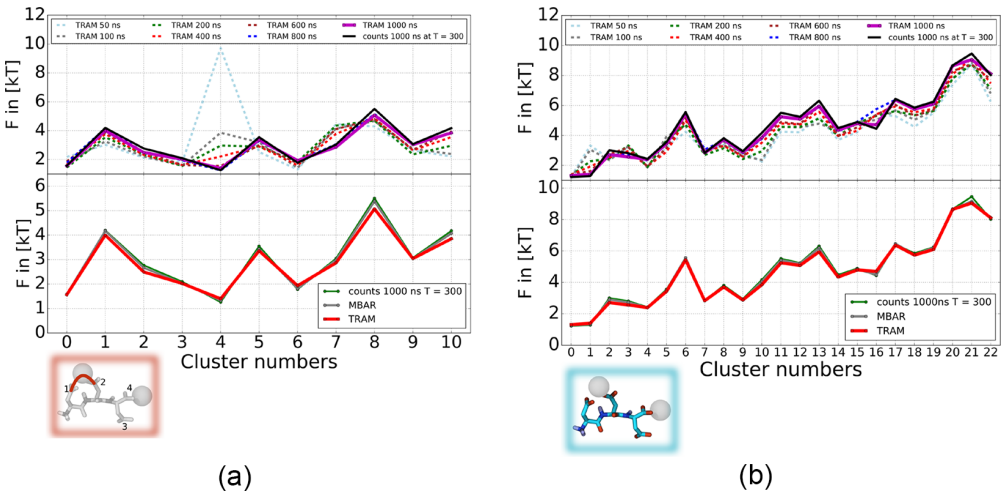


Figure 8. Free energy values calculated using TRAM for the conformation- and ion bridge-based clustering methods. Convergence of free energy values calculated using TRAM for different simulation lengths (in comparison with direct counting of the lower replica of 1 μs simulation) for (b, upper plot) conformation-based clustering and (a, upper plot) ion bridge-based clustering. Direct counts, TRAM, and MBAR free energy estimates for states of 8 Hamiltonians simulated for 1 μs obtained by applying (b, lower plot) conformation-based clustering (numbering of clusters correspond to the numbers in Figure 7a) and (a, lower plot) ion bridge-based clustering (numbers of clusters correspond to the numbers in Figure 6).

were projected into two dimensions using 1000 landmarks. The projection was discretized into bins. Following the procedure described in Sec. 2.3.2, bins with more than 30 points inside, which are separated by a change in the density gradient, were selected as cluster centers. All other points were assigned to the closest cluster center. In total we obtained 23 clusters. Figure 7a shows the low-dimensional projection, the cluster centers with respective numbers, the cluster boundaries, and representative structures with ions (water is removed) for each cluster center. The same projection method, but with different CVs, was already successfully applied by Kahlen et al.[8] to describe a conformational space of a peptide in the presence of ions. As in their case, accessible regions of the conformational space of a peptide are nicely separated (see Figure 7a).

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

The presence of ions on the offset plots can be misleading: ion positions were not taken into account during the projection, meaning that if peptides have the same conformation irrespective of the number of ions in proximity, they are assigned to the same cluster. Next we used the ion bridge-based characterization of states described in Sec. 2.3.1 and Sec. 3.1 getting 11 clusters (10 main clusters are shown in Figure 6).

To obtain free energy estimates of the simulated ensembles we used TRAM. We also included the estimates from direct counting in the lowest replica and MBAR. A lag time of 1 (2 ps) was chosen for TRAM. It has to be mentioned that exchanges between replicas were done very often. This, significantly reduces the possible length of the lag time which is used in TRAM to determine the transitions between states (the key difference to MBAR). The obtained estimates are shown in Figure 8. To get an idea of convergence of our HRE scheme we monitor the values of TRAM for different simulation lengths (we chose to use TRAM values rather than counts in the lowest replica, to use as much information as accessible to us). The upper plots of Figure 8 show the free energy estimates for different simulation times starting from 50 ns. Remarkable differences in the free energy values (especially for the cluster 4 of Figure 8a) can be seen only for shorter simulation lengths (upper plots of Figure 8) due to the fact that not all states were populated enough at these time intervals. The differences in the free energy estimates obtained for longer times (800 ns−1 μs) (upper plots of Figure 8) and using TRAM, MBAR, and direct counting for 1 μs simulations (lower plots of Figure 8) are small. We assume that it is due to the fact that we were able to obtain enough sampling using the proposed HRE parameters, i.e. for long simulation times the application of reweighting techniques (adding data from all replicas as well as information about the transitions between the states) does not significantly alter (and presumably improve) the estimation of the free energy differences between states.

3.2.2. Comparing Peptide Conformation- and Ion Bridge-Based Clustering. Here, we compare the free energy estimates obtained by the conformation- and ion bridge-based approaches. Although none of them includes the information on the other explicitly, they have strong correlations. This can be seen in Table 3 where we show how the structures assigned to one cluster in one clustering scheme are distributed among the clusters of the other clustering scheme. The highest values in the table are highlighted in gray. Keep in mind that clusters 0 and 1 of the ion bridge-based clustering are somewhat special. Cluster 0 contains all structures which are not among the 10 most probable ion bridge connection patterns. Cluster 1 contains all structures without ion bridges. It is therefore not surprising that this ion bridge cluster 1 contains structures from various peptide conformation-based clusters. Apart from these exceptions, clusters in one clustering scheme correspond to only a few clusters in the other scheme. A visualization of overlapping clusters from both schemes is shown at the top of Figure 9. We selected the four biggest overlaps with cluster number 7 of conformation-based clustering: clusters 0, 1, 5, and 10 from the ion bridge clustering. Also, the states 5 and 10 from the conformation-based clustering were compared with the corresponding clusters 5 and 10 from the ion bridge-based characterization. One can nicely see how all structures from a conformation-based cluster have similar backbone conformations but different ion bridges, while structures from an ion bridge-based cluster have similar ion bridges but quite different peptide conformations.

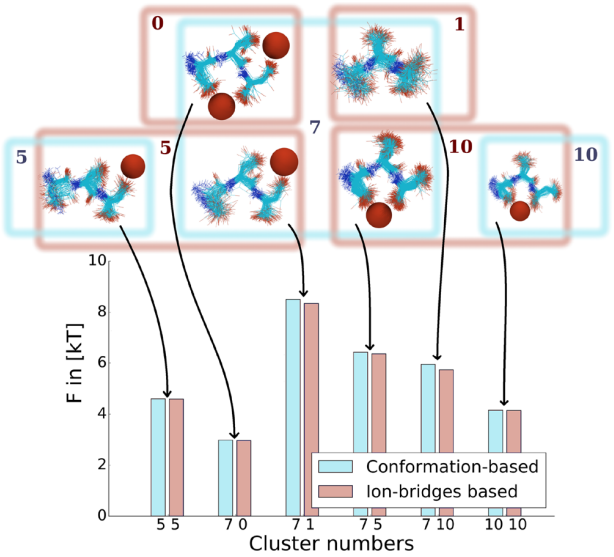


Figure 9. Free energy values for the states of interests. The first cluster number corresponds to the conformation-based state characterization (blue bars, see Figure 7a); the second cluster number corresponds to the ion bridge-based characterization (red bars, see Figure 6) (see Table 3). The inset plots show representative structures from the overlap between both clustering methods.

We use these overlaps of clusters for a test of the calculated free energy values. The TRAM algorithm allows to directly recalculate the free energy values for the new discrete states based on the statistical weights assigned to each frame in the trajectory. We compare the estimates obtained with each characterization approach for the overlap clusters in separate runs of TRAM. The free energies of these 6 states and representative clusters are shown in Figure 9. The estimates obtained by both characterization methods are almost equal. This suggests that none of the proposed discretization schemes in combination with reweighting distorts the statistical weights of structures, so each of them is equally applicable for defining states.

3.2.3. Combined Peptide Conformation- and Ion Position-Based State Characterizations. As we discussed in Sec. 2.3.3 both conformation and ion position information can efficiently be combined. We repeated the procedure described in Sec. 3.2.1 for states defined using a sketch-map projection based on the descriptors, which includes both structural information on the peptide and the ion positions with respect to it. The obtained two-dimensional representation, clusters, and some representative structures are shown in Figure 7b. As mentioned earlier, this approach nicely separates regions with different numbers of ions near the peptide (from no ions to the maximum number of ions).

In Figure 10a we show how different replicas populate the conformational space: In the lowest Hamiltonian, states with no ion contacts are rare (highlighted region I). In the highest Hamiltonian, the opposite is the case. States with no ions are highly populated (highlighted region I), and structures stabilized by ion bridges are rare (highlighted region II). This again shows that the parameters selected for the HRE work as intended. They reduce the number ions bound to the peptide in the higher replicas to facilitate sampling.

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

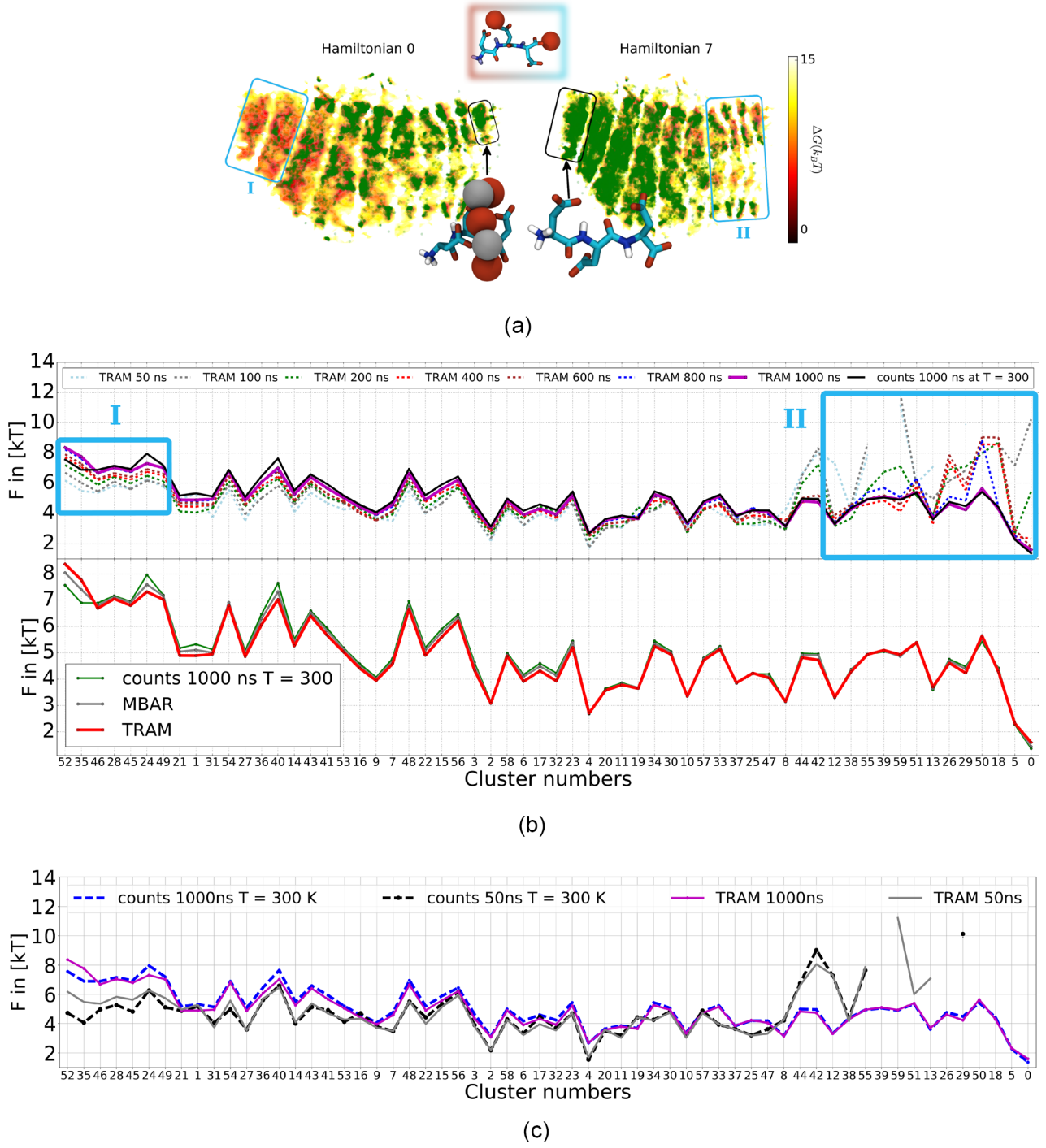


Figure 10. (a) Both plots show the sketch-map projection of all structures from 8 replicas. The green points are projections of Hamiltonian 0 (left-hand side) and Hamiltonian 7 (right-hand side). Regions of a phase space with different numbers of ions near the peptide are nicely separated. The regions with the minimum (I) and the maximum (II) number of ions are highlighted, and representative structures are shown. (b) Free energy estimates for the states defined using the method described in Sec. 2.3.3 are shown. The cluster numbers correspond to the clusters shown in Figure 7b and are ordered from left to right according to the projection. The upper plot shows the difference in the TRAM estimates depending on the simulation lengths. The lower plot shows the estimates obtained by applying TRAM and MBAR to all 8 Hamiltonians (each 1 μs long) and direct counting in the lowest Hamiltonian after 1 μs. The areas I and II have the biggest difference in the estimates and are highlighted in the projections in (a). (c) Comparison of TRAM and direct counting values for 50 ns and 1 μs of simulation.

As for the other state characterization methods we used TRAM to get free energy estimates for different simulation lengths and also compare the results of direct counting, MBAR, and TRAM. The results are shown in Figure 10b where region I (few ion contacts) and region II (many ion contacts) are again highlighted. The comparison of free energy estimates

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

after different simulation times shows severe differences in region II. It is not surprising that these structures with many ions in contact take the longest to occur and to equilibrate. Such drastic differences were not visible in the free energy estimates based on the other two state characterization methods (Figure 8). This characterization scheme combining peptide conformation and ion positions, therefore, can give a more detailed picture to decide on convergence.

Differences between TRAM, MBAR, and direct counting (lower plot of a Figure 10b) can be mainly found for states with high free energy e.g. the states in region I. These states are not highly populated in the lowest replica. Here, improving the statistics by including information from higher replicas by reweighting with MBAR or TRAM can be beneficial for the free energy estimates. The difference between direct counting and reweighting is even more prominent for shorter simulation times, as can be seen in Figure 10c. Here we show the estimates after 50 ns of simulation in comparison with the results after 1000 ns. For low-free-energy states TRAM does not significantly improve the 50 ns free energy estimates since low-free-energy structures are handed down to the lowest replica anyway and are thus included in the direct counting. Here, better free energy estimates are only achieved by simulating longer. For high-free-energy states (region I) the 50 ns TRAM free energy estimates differ significantly from the direct counting result. The 50 ns TRAM free energy estimates are closer to the values obtained after 1000 ns, which shows that the results do not only differ but are actually really improved. For longer simulation times, there are only small differences between direct counting and the reweighted results. This leads us to the assumption that differences between direct counting and reweighted values might be a useful indicator of nonconvergence.

4. CONCLUSIONS

Simulations of ion−peptide systems suffer from slow sampling due to high energetic barriers caused by strong ion bridges. Here we proposed HRE parameters that can efficiently enhance sampling for such systems by specifically targeting ion−peptide interactions and facilitating rotation around backbone dihedrals. We also found that this specificity of a HRE can be problematic if not all sampling-hindering interactions can be anticipated beforehand. In our case this comes up as soon as peptide conformations are no longer locked by separate ion bridges but small clusters of calcium chloride formed at the peptide.

To be able to judge the conformational variety and stability of different structures (formation and distortion), and subsequently the success of the HRE, we introduced and compared three methods for state characterization. Those methods concentrate on different ways of defining a state (building of ion bridges between the peptide side chains, structural changes in the peptide, both structural changes of the peptide and ions in proximity) and have their drawbacks and advantages. The most prominent advantage of the ion bridge-based clustering (Sec. 2.3.1) is that clusters can be defined beforehand, and one can monitor the behavior of the system from the very beginning of the simulation. However, for bigger systems it can lead to a large number of states to monitor. One should also take into account that this characterization method is insensitive to the changes in peptide conformations and the number of ions around. If one is interested in peptide rearrangements under the different simulation conditions, the conformation-based clustering (Sec. 2.3.2) is the most appropriate choice. However, we showed that using this clustering method one might get the impression of earlier convergence of the simulation because some important ion bridge stabilized structures are not revealed. To add information about ions explicitly, the combined conformation- and ion position-based clustering (Sec. 2.3.3) was used. It gives a detailed overview over the peptide conformations and the position of ions around it. Despite the fact that it can also lead to a large number of clusters, the possibility of working with 2D projections makes it still feasible to gain insight into the simulation progress and results. A joint drawback of the last two clustering methods is the necessity for reprojection and reclustering, when one wants not only to analyze the final outcome but also to monitor HRE while the simulation is running and new states are appearing.

Using these three characterization methods we discretized the conformational space into states. To get quantitative estimates for the statistical weights of the defined states, unbiasing techniques were tested and compared. In contrast to directly counting the number of state occurrences in the lowest replica, the reweighting methods (MBAR, TRAM) allowed us to also include information from higher replicas. Reweighting appears to be useful for states without ions attached to the peptide, because those are comparably high energy states which are hardly present in the lowest replica. For low energy states reweighting was not beneficial in combination with HRE as low energy structures are passed down to the lowest replica anyway. With increasing simulation length, differences between the reweighting and the direct counting result were decreasing. While this means that reweighting might not bring much improvement for long HRE simulations close to convergence, it also shows that differences between direct counting and reweighting might be a valuable indicator for nonconvergence.

All the methodology we described combined yields a complete and efficient workflow to sample and characterize free energy landscapes of ion−peptide systems and will allow for further insights into research questions like the role of peptides in the early stages of biomineralization.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: Christine.Peter@uni-konstanz.de.
*E-mail: Oleksandra.Kukharenko@uni-konstanz.de.

ORCID

Tobias Lemke: 0000-0002-0593-2304
Christine Peter: 0000-0002-1471-5440
Oleksandra Kukharenko: 0000-0002-3285-1403

Funding

This work was supported by a fellowship of the Zukunftskolleg of the University of Konstanz and the Carl Zeiss Foundation to O.K. and by the SFB1214 funded from the German Research Foundation. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1134-1 FUGG.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We would like to thank F. Noé for very helpful discussions and A. Jain for providing the modified force field parameters.

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

REFERENCES

(1) Lowenstam, H. A.; Weiner, S. *On biomineralization*; Oxford University Press: USA, 1989.
(2) Weiner, S.; Addadi, L. Design strategies in mineralized biological materials. *J. Mater. Chem.* **1997**, *7*, 689−702.
(3) Meldrum, F. C.; Cölfen, H. Controlling Mineral Morphologies and Structures in Biological and Synthetic Systems. *Chem. Rev.* **2008**, *108*, 4332−4432.
(4) Gebauer, D.; Cölfen, H. Prenucleation clusters and non-classical nucleation. *Nano Today* **2011**, *6*, 564−584.
(5) Wallace, A. F.; Hedges, L. O.; Fernandez-Martinez, A.; Raiteri, P.; Gale, J. D.; Waychunas, G. A.; Whitelam, S.; Banfield, J. F.; De Yoreo, J. J. Microscopic Evidence for Liquid-Liquid Separation in Supersaturated CaCO3 Solutions. *Science* **2013**, *341*, 885−889.
(6) Demichelis, R.; Raiteri, P.; Gale, J. D.; Quigley, D.; Gebauer, D. Stable prenucleation mineral clusters are liquid-like ionic polymers. *Nat. Commun.* **2011**, *2*, 590.
(7) Yang, Y.; Cui, Q.; Sahai, N. How does bone sialoprotein promote the nucleation of hydroxyapatite? A molecular dynamics study using model peptides of different conformations. *Langmuir* **2010**, *26*, 9848−9859.
(8) Kahlen, J.; Peter, C.; Donadio, D. Molecular simulation of oligo-glutamates in a calcium-rich aqueous solution: insights into peptide-induced polymorph selection. *CrystEngComm* **2015**, *17*, 6863−6867.
(9) Jain, A.; Jochum, M.; Peter, C. Molecular Dynamics Simulations of Peptides at the Air-Water Interface: Influencing Factors on Peptide-Templated Mineralization. *Langmuir* **2014**, *30*, 15486−15495.
(10) Chen, J.; Trout, B. L. Computational Study of Solvent Effects on the Molecular Self-Assembly of Tetrolic Acid in Solution and Implications for the Polymorph Formed from Crystallization. *J. Phys. Chem. B* **2008**, *112*, 7794−7802.
(11) Raiteri, P.; Demichelis, R.; Gale, J. D.; Kellermeier, M.; Gebauer, D.; Quigley, D.; Wright, L. B.; Walsh, T. R. Exploring the influence of organic species on pre-and post-nucleation calcium carbonate. *Faraday Discuss.* **2012**, *159*, 61−85.
(12) Hukushima, K.; Nemoto, K. Exchange Monte Carlo Method and Application to Spin Glass Simulations. *J. Phys. Soc. Jpn.* **1996**, *65*, 1604−1608.
(13) Marinari, E.; Parisi, G. Simulated Tempering: A New Monte Carlo Scheme. *Europhys. Lett.* **1992**, *19*, 451.
(14) Laio, A.; Parrinello, M. Escaping free-energy minima. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99*, 12562−12566.
(15) Torrie, G.; Valleau, J. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **1977**, *23*, 187−199.
(16) Sugita, Y.; Okamoto, Y. Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.* **1999**, *314*, 141−151.
(17) Grubmüller, H. Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **1995**, *52*, 2893−2906.
(18) Hamelberg, D.; Mongan, J.; McCammon, J. A. Accelerated molecular dynamics: A promising and efficient simulation method for biomolecules. *J. Chem. Phys.* **2004**, *120*, 11919−11929.
(19) Liu, P.; Kim, B.; Friesner, R. A.; Berne, B. J. Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 13749−13754.
(20) Wang, L.; Friesner, R. A.; Berne, B. J. Replica Exchange with Solute Scaling: A More Efficient Version of Replica Exchange with Solute Tempering (REST2). *J. Phys. Chem. B* **2011**, *115*, 9431−9438.
(21) Oleinikovas, V.; Saladino, G.; Cossins, B. P.; Gervasio, F. L. Understanding Cryptic Pocket Formation in Protein Targets by Enhanced Sampling Simulations. *J. Am. Chem. Soc.* **2016**, *138*, 14257−14263.
(22) Bulo, R. E.; Donadio, D.; Laio, A.; Molnar, F.; Rieger, J.; Parrinello, M. Site Binding" of Ca2+ Ions to Polyacrylates in Water: A Molecular Dynamics Study of Coiling and Aggregation. *Macromolecules* **2007**, *40*, 3437−3442.
(23) Bewernitz, M. A.; Gebauer, D.; Long, J.; Cölfen, H.; Gower, L. B. A metastable liquid precursor phase of calcium carbonate and its interactions with polyaspartate. *Faraday Discuss.* **2012**, *159*, 291−312.
(24) Wolf, S. L.; Jähme, K.; Gebauer, D. Synergy of Mg 2+ and poly (aspartic acid) in additive-controlled calcium carbonate precipitation. *CrystEngComm* **2015**, *17*, 6857−6862.
(25) Zou, Z.; Bertinetti, L.; Politi, Y.; Fratzl, P.; Habraken, W. J. E. M. Control of Polymorph Selection in Amorphous Calcium Carbonate Crystallization by Poly(Aspartic Acid): Two Different Mechanisms. *Small* **2017**, *13*, 1603100.
(26) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 13023−13028.
(27) Tribello, G. A.; Ceriotti, M.; Parrinello, M. Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 5196−5201.
(28) Shirts, M. R.; Chodera, J. D. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.* **2008**, *129*, 124105.
(29) Wu, H.; Paul, F.; Wehmeyer, C.; Noé, F. Multiensemble Markov models of molecular thermodynamics and kinetics. *Proc. Natl. Acad. Sci. U. S. A.* **2016**, *113*, E3221−E3230.
(30) van der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, flexible, and free. *J. Comput. Chem.* **2005**, *26*, 1701−1718.
(31) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435−447.
(32) Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; Gunsteren, W. F. Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur. Biophys. J.* **2011**, *40*, 843−856.
(33) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.
(34) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, 014101.
(35) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81*, 3684−3690.
(36) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(37) Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18*, 1463−1472.
(38) Bussi, G. Hamiltonian replica exchange in GROMACS: a flexible implementation. *Mol. Phys.* **2014**, *112*, 379−384.
(39) Barducci, A.; Bussi, G.; Parrinello, M. Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Phys. Rev. Lett.* **2008**, *100*, 020603.
(40) Pearson, K. L., III On lines and planes of closest fit to systems of points in space. *Philos. Mag.* **1901**, *2*, 559−572.
(41) Lloyd, S. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129−137.
(42) Ding, C.; He, X. K-means Clustering via Principal Component Analysis. *Proceedings of the Twenty-first International Conference on Machine Learning*; New York, NY, USA, 2004; pp 29−36, DOI: 10.1145/1015330.1015408.
(43) Lindorff-Larsen, K.; Ferkinghoff-Borg, J. Similarity Measures for Protein Ensembles. *PLoS One* **2009**, *4*, 1−13.
(44) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Demonstrating the Transferability and the Descriptive Power of Sketch-Map. *J. Chem. Theory Comput.* **2013**, *9*, 1521−1532.
(45) Kukharenko, O.; Sawade, K.; Steuer, J.; Peter, C. Using Dimensionality Reduction to Systematically Expand Conformational Sampling of Intrinsically Disordered Peptides. *J. Chem. Theory Comput.* **2016**, *12*, 4726−4734.

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

(46) Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492−1496.
(47) van Gunsteren, W.; Daura, X.; Mark, A. Computation of Free Energy. *Helv. Chim. Acta* **2002**, *85*, 3113−3129.
(48) Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A. THE weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comput. Chem.* **1992**, *13*, 1011−1021.
(49) Souaille, M.; Roux, B. Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations. *Comput. Phys. Commun.* **2001**, *135*, 40−57.
(50) Bereau, T.; Swendsen, R. H. Optimized convergence for multiple histogram analysis. *J. Comput. Phys.* **2009**, *228*, 6119−6129.
(51) Kobrak, M. N. Systematic and statistical error in histogram-based free energy calculations. *J. Comput. Chem.* **2003**, *24*, 1437−1446.
(52) Tan, Z.; Gallicchio, E.; Lapelosa, M.; Levy, R. M. Theory of binless multi-state free energy estimation with applications to protein-ligand binding. *J. Chem. Phys.* **2012**, *136*, 144102.
(53) *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*; Bowman, G. R., Pande, V. S., Noé, F., Eds.; Springer Netherlands: Dordrecht, 2014; Vol. *797*, DOI: 10.1007/978-94-007-7606-7.
(54) Wu, H.; Mey, A. S. J. S.; Rosta, E.; Noé, F. Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. *J. Chem. Phys.* **2014**, *141*, 214106.
(55) Rosta, E.; Hummer, G. Free Energies from Dynamic Weighted Histogram Analysis Using Unbiased Markov State Model. *J. Chem. Theory Comput.* **2015**, *11*, 276−285.
(56) Mey, A. S. J. S.; Wu, H.; Noé, F. xTRAM: Estimating Equilibrium Expectations from Time-Correlated Simulation Data at Multiple Thermodynamic States. *Phys. Rev. X* **2014**, *4*, 041018.
(57) Scherer, M. K.; Trendelkamp-Schroer, B.; Paul, F.; Pèrez-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. *J. Chem. Theory Comput.* **2015**, *11*, 5525−5542.

DOI: 10.1021/acs.jctc.8b00560
J. Chem. Theory Comput. 2018, 14, 5476−5488

JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

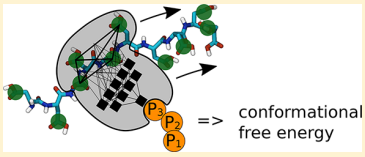
Neural Network Based Prediction of Conformational Free Energies - A New Route toward Coarse-Grained Simulation Models

Tobias Lemke and Christine Peter*

Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Germany

Supporting Information

ABSTRACT: Coarse-grained (CG) simulation models have become very popular tools to study complex molecular systems with great computational efficiency on length and time scales that are inaccessible to simulations at atomistic resolution. In so-called bottom-up coarse-graining strategies, the interactions in the CG model are devised such that an accurate representation of an atomistic sampling of configurational phase space is achieved. This means the coarse-graining methods use the underlying multibody potential of mean force (i.e., free-energy surface) derived from the atomistic simulation as parametrization target. Here, we present a new method where a neural network (NN) is used to extract high-dimensional free energy surfaces (FES) from molecular dynamics (MD) simulation trajectories. These FES are used for simulations on a CG level of resolution. The method is applied to simulating homo-oligo-peptides (oligo-glutamic-acid (oligo-glu) and oligo-aspartic-acid (oligo-asp)) of different lengths. We show that the NN not only is able to correctly describe the free-energy surface for oligomer lengths that it was trained on but also is able to predict the conformational sampling of longer chains.



1. INTRODUCTION

Molecular dynamics (MD) simulation is a popular tool to investigate the structure and dynamics of molecular systems. For large systems and systems with slow dynamics, however, MD simulations become computationally intractable. One approach to tackling this problem is to reduce the number of particles from an atomistic to a coarse level of resolution. Typically several atoms/fine-grained particles are mapped into a coarse-grained (CG) bead, and in many commonly used mapping approaches each atom is represented by one of the CG beads.[1] Another coarse-graining strategy is to select structurally important atoms as CG beads and to omit all other atoms.[2] A coarse-grained protein model could for example only consist of alpha-carbon atoms. In such alpha-carbon based (or similar) models, the connectivity of the protein is typically very well reflected in the mapping. In this paper, we present a different approach. In our model only atoms or groups which are believed to be important for the function of interest, such as positions of specific chemical groups in the side chains, are represented. All other atoms, also those central to the molecule structure, such as the complete peptide backbone, are not represented in this sparse CG representation. This type of coarse-graining will require an entirely different approach to CG interaction functions, since for such a representation, typical bonded and nonbonded (pair) interactions are potentially ill-suited. We will show a neural network based approach toward suitable CG potentials for such a scenario.

In bottom-up coarse-graining methods, CG interactions are parametrized such that an accurate representation of a (known) atomistic sampling of configurational phase space is achieved. Different methods like iterative Boltzmann inversion,[3] relative entropy,[4] or force matching[1] can be used to approximate the underlying multibody potential of mean force (i.e., the free energy surface which is the Boltzmann-inverse of the probability density of configurations in phase space evaluated in CG coordinates), with low-dimensional potentials.[5−7] While it might be enough to use typical low-dimensional potentials like pair and angular potentials on a rather fine-grained scale, a drastic reduction of resolution demands for a rather complex high-dimensional description of the multibody potential of mean force (from now on denoted as free energy surface (FES)). For atomistic resolution, there are many approaches of how to obtain high-dimensional potential energy surfaces (PES).[8−10] One successful class of methods is based on artificial neural networks (NN) which are trained to reproduce and interpolate potential energies typically calculated with ab initio methods.[11−15] Here, we take this NN approach to a coarser level of resolution. In contrast to atomistic models, which are parametrized on a PES, a CG model is based on the above-mentioned FES, where multiple fine-grained structures are mapped to the same CG structure. One of the main challenges is that in contrast to potential energies, free energies are not a direct output of a MD simulation. They can be calculated by Boltzmann inversion of the probability density of sampling configurations in phase space. The most common way to estimate a probability density is to bin the space and to count the number of data points in each bin. This comes with two problems: (1) information is lost in the binning procedure, and (2) the number of bins required grows exponentially with the number of dimensions. To give an example, for a molecule conformation described in 10 dimensions with 100 bins in each dimension this would result in $100^{10} = 10^{20}$ bins. Pushing aside the fact that such a large number of bins could not be handled,

Received: August 14, 2017
Published: November 9, 2017

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

it would still not be useful. The number of data points would be much smaller than the number of bins, and most of the bins would therefore be empty or just contain a single data point. This problem could be tackled with dimensionality reduction algorithms,[16−18] but this relies on finding a low-dimensional representation of the FES where the states of the system are well represented by a few collective variables. Here, we propose a method that relies neither on low-dimensional potential functions such as typical bonded and nonbonded potentials nor on a dimensionality reduction of the FES to a few collective variables and that circumvents the problem of binning in high-dimensional space.

Modha and Fainman[19] described a method of how to train neural networks to return probability densities without any binning needed. This method, however, relies on normalization of the probability density function for each training step, which involves numerical integration and is therefore again limited to low dimensions. Galvelis and Sugita[20] recently published a very interesting method based on nearest neighbor density estimation. However, also here the high computational cost of the nearest neighbor density estimation limits the accessible dimensionality. Garrido and Juste[21] described a method of how to train neural networks to return probability densities without any binning or normalization required and that does not involve nearest neighbor search. The basic idea is not to use probabilities as learning targets directly but to create a classification task instead. The task of the NN is to distinguish between real data points and "fake" points drawn from a known distribution. Based on how confident the NN is that a given point is from the real data distribution and not from the "fake distribution", it is possible to calculate the probability density at this point in the data distribution. A very similar approach was also published more recently under the name noise-contrastive estimation.[22]

We evaluate how useful such a classification based method is to analyze high-dimensional free energy surfaces of molecule conformations and to open up a new path to create coarse-grained models. We use oligo-glu and oligo-asp peptides with different chain lengths as test systems. Oligo-glu and oligo-asp acid are known mineralization modifiers.[23−25] It is believed that their influence on mineralization originates from the carboxylic acid groups in the side chain binding to ions and to forming minerals. The sparse CG representation that we chose for these peptides is therefore focused on these carboxylic acid groups. It consists of the carbon atoms of the side-chain carboxylic acid groups (see Figure 1). The method that we propose, however, is not limited to this kind of representation and could in principle also be applied to any other definition of coarse-grained beads.

There are 3 major challenges of increasing difficulty that we pose with these test systems: (1) a NN trained with data obtained from atomistic MD simulations of a given peptide length should be able to return the FES of this peptide; (2) a NN trained with data of different peptide lengths should be able to return the FES for all the lengths that it was trained on; (3) a NN trained with data of some peptide lengths should be able to predict the FES of a peptide length that it was not trained on. In the next part we give an overview of the different steps involved in our method, followed by a more detailed description of each step. In the last part we show results of the application to different peptide lengths and evaluate how close we come to the above-mentioned goals.

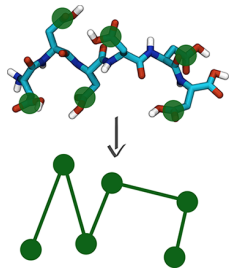


Figure 1. Illustration of the coarse-grained model. It is only based on the positions of the carboxylic acid carbon atoms in the side chain.

2. METHODS

2.1. NN Training Framework for Conformational Free Energies. A NN is trained by adjusting its weights to minimize a cost function. The cost function used in this work is

$$C(N) = \frac{1}{n}\sum_{i=1}^{n}[N(R_i) - l_i]^2 \tag{1}$$

where N is the Neural Network returning a number between 0 and 1, R_i is a set of variables used to describe a CG molecule conformation (see section 2.2), l_i is the label of a conformation which is $l_i = 1$ if it was sampled in the atomistic MD simulation and $l_i = 0$ if it is a fake conformation, and n is the number of molecule conformations used in one training batch. We refer to "data distribution" as the distribution sampled in the atomistic MD simulation. The "fake distribution" can be any known distribution of molecule conformations that is nonzero anywhere the data distribution is nonzero. To train the NN, both conformations sampled according to the data distribution (MD) and conformations sampled according to a known fake distribution have to be provided. It was shown[21] that for a trained neural network that minimizes the cost function (eq 1) the probability density of the data distribution $P_{\text{data}}(R)$ is given by

$$P_{\text{data}}(R) = P_{\text{fake}}(R)\frac{\alpha_{\text{fake}}}{\alpha_{\text{data}}}\frac{N_{\text{min}}(R)}{1 - N_{\text{min}}(R)} \tag{2}$$

where $P_{\text{fake}}(R)$ is the probability density function of the fake conformations, α_{fake} and α_{data} are the proportions of fake and data conformations used for training, and $N_{\text{min}}(R)$ is the output of the trained neural network that minimizes the cost function (eq 1). Under the condition that we use the same number of data and fake conformations for training and using $\Delta F = -k_{\text{B}}T\ln\left(\frac{P_{\text{data}}(R)}{P_{\text{fake}}(R)}\right)$ it follows that

$$\Delta F(R) = -k_{\text{B}}T\ln\left(\frac{N_{\text{min}}(R)}{1 - N_{\text{min}}(R)}\right) \tag{3}$$

where $\Delta F(R)$ is the difference between the free energy of a given conformation in the data distribution and the free energy of the same conformation in the fake distribution, k_{B} is the Boltzmann constant, and T is the temperature.

Using the above framework to evaluate and predict conformational free energies involves three major parts as illustrated in Figure 2: (1) The training data has to be prepared. As data distribution we use conformations sampled in an atomistic MD simulation, mapped to the CG coordinates R (here based on the side chain carboxylic acid positions). The fake distribution could contain conformations of any known distribution. The most simple approach is to use conformations from a

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

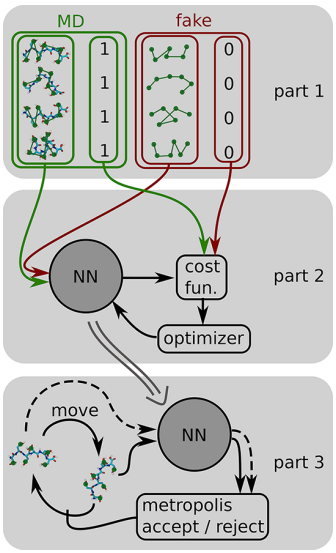


Figure 2. NN training framework for conformational free energies: part 1, input preparation; part 2, training of the NN; part 3, Monte Carlo simulation with the NN as Hamiltonian.

(inside certain boundaries) uniform distribution, meaning that all conformations have the same free energy in the fake distribution. (2) A neural network has to be trained. The weights and biases are optimized in order to minimize the cost function (eq 1). Here, a challenge is that the dimensionality of the set of variables R, used to describe a conformation, grows with the length of the peptide. To be able to use the same NN for different chain lengths the NN has to be able to deal with different numbers of inputs. We use a convolutional NN to achieve this. (3) The training result needs to be validated. Once the NN is trained, we have a tool that predicts the probability of sampling a given conformation which is directly related to the respective free energy of this conformation. To sample peptide conformations according to their correct Boltzmann weights, we now perform Monte Carlo (MC) simulations in the coarse-grained phase space using the multidimensional FES from the NN as Hamiltonian. We compare the distributions of conformations sampled in the MD simulation and the NN MC simulation.

The following sections describe these three parts in further detail before the results are shown and discussed.

2.2. Input Preparation. For comparison and for the use as training data, a set of peptide MD simulations was created. It consists of 5 μs MD trajectories of uncharged hexamer, heptamer, octamer, and decamer chains for oligo-glu and oligo-asp each. This fully protonated state corresponds to a pH of around 3 and below.[26] The only structural difference between oligo-asp and oligo-glu is the additional methylene group in the side chains of oligo-glu. Despite their structural similarity, oligo-asp and oligo-glu are two very different test systems. Oligo-glu conformations are dominated by helical structures, while oligo-asp conformations are rather disordered.

The MD simulations were performed with the Gromacs 2016 simulation package[27] with the GROMOS 54A7[28] force field and the SPC/E water model.[29] The temperature was kept constant at 300 K with stochastic velocity rescaling (Bussi thermostat).[30] No barostat was used in the main simulations. In preceding equilibration simulations a Berendsen barostat[31] was used to adjust the pressure to 1 bar. The leapfrog algorithm with a time step of 2 fs was used to integrate the equations of motion. Long range interactions were calculated with the particle mesh Ewald method[32] with a grid spacing of 0.12 nm and a pme-order of 4. Both Coulomb and Lennard-Jones interactions were truncated at 1.4 nm. Bonds have been constrained using the linear constraints solver (LINCS) algorithm[33] with an eighth order expansion. The coordinates were written to the trajectory every 10 ps.

The coarse-grained representation we build here is based on positions of carboxylic acid groups (see Figure 1). There is only one bead per amino acid which is located at the side chain carboxylic acid carbon atom. The backbone is not represented at all. All the information about the positions of side chain beads that is usually contained in the backbone has to be learned by the NN. To make it as easy as possible for the NN to learn this kind of information it is important how the data is presented to the NN.

Molecule conformations can easily be described in Cartesian coordinates of atoms. However, these coordinates change with molecule translation and rotation, while the conformational free energy of the molecule stays the same. Internal measures like pairwise distances are therefore more suitable descriptors. For a small molecule, the system can be described by all pairwise distances. For larger molecules like oligomers, the number of all pairwise distances becomes unfeasible as NN inputs. Furthermore, the number of pairwise distances depends on the oligomer length, and it would not be possible to use the same NN for different chain lengths. Here, we describe a chain as a sequence of overlapping tetramer segments. In other words, the tetramer segment description is convolved along the chain. For chains of different lengths only the number of segments changes, while the number of descriptors per segment is constant. Using only pairwise distances, it would not be possible to distinguish between a conformation and its mirror image. Using the triple product of the vectors shown in Figure 3 as additional

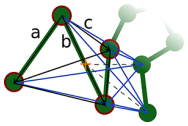


Figure 3. Illustration of the 15 descriptors used for each tetramer segment. The respective tetramer segment is highlighted in red. The 6 black lines show the internal distances inside the segment. The 8 blue lines illustrate the distances to the nearest two beads which are not part of the segment. The dashed lines symbolize the distances used to determine the two nearest beads to the central point between the two inner beads of the tetramer segment(orange cross). The 15th descriptor is the triple product of the vectors a, b, and c.

input adds this chiral information. With these descriptors containing only information about the tetramer segment itself it would be possible for two distant tetramer segments in a long chain to overlap. Additional information about other beads nearby is needed. We use all eight distances of the four segment beads i to the two spatially closest beads j that are not part of the segment as additional inputs. A cutoff of d_{cut} = 0.7 nm is applied to these nonbonded distances in the following way

$$d_{i,j;\mathrm{cut}} = \min(d_{\mathrm{cut}}, d_{i,j}) \tag{4}$$

which ensures that there is no discontinuity at the cutoff. The nearest beads to the segment are determined based on their distance to the central point between the two inner beads of the

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

segment. This adds up to 15 descriptors for each tetramer segment illustrated in Figure 3. These descriptors can easily be calculated for each conformation sampled in the MD simulations. Each of these data points is given the label $l_i = 1$ (see eq 1). To train the NN to minimize the cost function (eq 1) a fake distribution is required of which each point is given the label $l_i = 0$. The simplest approach is to use a uniform distribution as the fake distribution. The idea is that each conformation in the fake distribution is equally probable and therefore has the same free energy. It is not feasible to sample all possible conformations of n beads. Therefore, we restrict the sampling to certain boundaries. The boundaries are given by the extrema of each descriptor found in the MD sampling. The fake conformations are sampled in a MC simulation. For each move random numbers are drawn from a Gaussian distribution centered around 0 with a standard deviation of 0.03 nm. Independent random numbers are drawn for each bead and each of the Cartesian dimensions and are added to the Cartesian coordinates of all beads. The move is rejected whenever any of the above descriptors are outside the boundaries. It is accepted in all other cases. Five million fake conformations were generated with this procedure for each peptide length. The next section describes the architecture of the neural network that is trained on these data. There, we will also introduce an additional NN input which is not relevant for the generation of the fake distribution and is easier to understand after already knowing the NN architecture.

2.3. Neural Network. In a common feed forward NN[34,35] the output $o(I)$ of each neuron is given by

$$o(I) = f\left(\sum_{j=1}^{m} [I_j w_j] + b\right) \tag{5}$$

where f is the activation function used, w and b are the weights and biases tuned during the training process, and I_j is the m inputs of the neuron. For neurons in the first layer the inputs are the molecule conformation descriptors mentioned in the previous section. For neurons in later layers, the inputs are the outputs of neurons in the previous layer.

Each peptide chain length results in a different number of overlapping tetramer segments and therefore in a different number of inputs for the neural network. To be able to train one NN on data of different chain lengths it is important that the number of weights of the NN must not depend on the number of segments. This can be achieved with convolutional layers.[36] In a convolutional layer not all inputs are processed at once. Only a given window of inputs is processed at once. This window is then moved along a given axis in the tensor of inputs. In our case, the first layer of neurons only processes inputs of one tetramer segment at a time. It then moves along the chain and returns outputs for each tetramer segment. For homo-oligomers and polymer type molecules one can expect segments to behave similarly irrespective of the segment position in the chain. In this case, a convolutional layer, where all segments are evaluated with the same shared neurons, appears to be the natural approach. A schematic for the NN architecture is shown in Figure 4. The outputs of the first layer are then processed in a second convolutional layer. This convolutional layer has a filter size of 3 meaning that it processes the outputs of three adjacent overlapping tetramer segments at once. As a consequence of this filter size, neurons in this layer have a broader "field of view". They have access to information about all beads in a hexamer segment. At the edges, a zero padding is

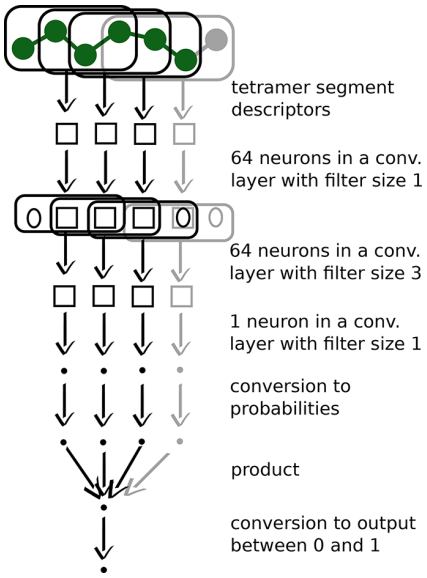


Figure 4. Illustration of the NN architecture for the chain length 6 (black) and length 7 (black and gray). All operations are symbolized with arrows and described on the right. The outputs and inputs of these operations are symbolized by squares if it is an array of numbers or by dots if it is a single number. The zeros stand for the zero padding used in the convolutional layer with filter size 3.

added in order to keep the number of outputs constant. The final layer is a convolutional layer with filter size one, consisting of only one single neuron. The output of this layer is one number per tetramer segment. A crucial point is how to reduce the output to a single number for the whole chain. To make it multiplicative, we first convert the output of each segment $N_{seg}(R_i)$ into a probability $P_{seg,i}$ in analogy to eq 2 with

$$P_{seg,i} = \frac{N_{seg}(R_i)}{1 - N_{seg}(R_i)} \tag{6}$$

All these probabilities are then multiplied

$$P_{join}(R) = \prod_{i=1}^{n} P_{seg,i} = \prod_{i=1}^{n} \frac{N_{seg}(R_i)}{1 - N_{seg}(R_i)} \tag{7}$$

where n is the number of segments. We can now calculate what neural network output would correspond to this joint probability

$$N(R) = \frac{P_{join}(R)}{1 + P_{join}(R)} \tag{8}$$

where $N(R)$ is the final NN output that is used in the cost function (eq 1). Another crucial point is to think about what $P_{seg,i}$ (see eq 7) stands for. If all segments were completely independent of each other, the probability of the whole chain would just be the product of the segment probabilities. $P_{seg,i}$ would then be the probability of finding a segment in a certain conformation. In the case of overlapping segments, the segments are clearly not independent of each other. $P_{seg,i}$ therefore has to be a conditional probability taking into account all the other overlapping segments. To give the NN the possibility to estimate these conditional probabilities we introduce an overlap indicator as additional input. The overlap indicator O is given by

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

$$O = \frac{\sum_i \alpha_i}{O_{\max}} \tag{9}$$

where α_i is the number of overlapping beads for each overlapping segment, and $O_{\max}$ is the highest possible overlap for a given segment length. Figure 5 illustrates how the overlap

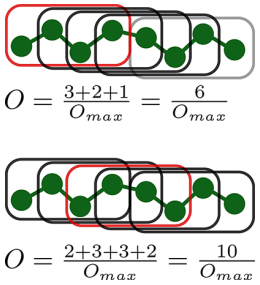


Figure 5. Two examples of how the overlap indicator is calculated for the segments highlighted in red. In this case of tetramer segments the highest possible overlap is given by $O_{\max} = 1 + 2 + 3 + 3 + 2 + 1 = 12$, which would occur in the middle of long chains.

indicator is calculated with two examples. The following equation describes how to calculate the overlap indicator as a function of the segment position in the chain

$$O(k) = \frac{\sum_{x=a}^{b} (-|x| + l) - l}{\sum_{x=-l+1}^{l-1} (-|x| + l) - l} \tag{10}$$

where $a = \max(-k, -l+1)$, $b = \min(a + n_{\text{seg}} - 1, l - 1)$, l is the length of a segment, $k = 0, 1, 2 \cdots (n_{\text{seg}} - 1)$ is the position in the chain, and n_{seg} is the number of segments in the whole chain. The main idea behind the overlap indicator is to give the NN the freedom to apply different weightings to contributions of segments with different overlap. The information on a segment with a great deal of overlap to other segments along the chain is probably to a large extent also contained in those segments (and therefore multiply counted as input). The contributions of such a segment should be weighted less compared to contributions of segments with little overlap to other segments. An important side effect of the overlap indicator is that the NN in general gets the freedom to treat segments at the ends of the chain differently compared to segments in the middle.

All NN weights (eq 5) are initialized with random numbers drawn from a Gaussian distribution centered at 0 with a standard deviation of 1. For all neurons the sigmoid function was used as an activation function. In a preprocessing step all inputs are rescaled to a comparable range by subtracting the mean and dividing by the standard deviation. The weights and biases are optimized using the Adam-optimizer[37] with a learning rate of 0.001 and the exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as implemented in TensorFlow 1.1.0.[38] L2 regularization[39] is used with a regularization constant of 10^{-6}. With each training step a batch of 512 (256 MD and 256 fake) conformations is processed.

The next section describes how the trained NN is used and compared to the MD data.

2.4. MC Based on NN Prediction. After training, the NN output for any given conformation can be converted to a free energy difference ΔF between the conformation in the fake distribution and the conformation in the distribution sampled in the MD simulation (eq 3). If we choose the fake distribution to be such that each conformation inside certain boundaries has the same free energy, the absolute free energy is given by

$$F(R) = F_{\text{fake}}(R) + \Delta F(R) = \text{const} + \Delta F(R) \tag{11}$$

As the free energy is not a priori known there is no direct comparison. To evaluate the accuracy of the NN predicted free energies, Metropolis Monte Carlo is used to sample a distribution of conformations according to the predicted free energy surface. All moves leading outside the boundaries of the uniform fake distribution are rejected. All other moves are accepted or rejected according to the Metropolis criterion

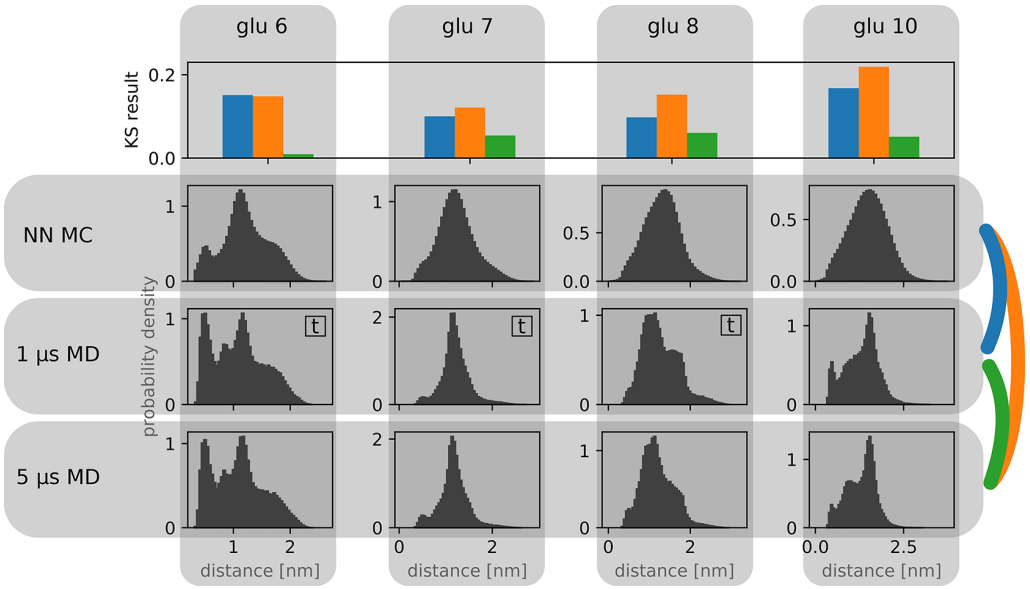


Figure 6. End-to-end distance distributions for MC simulations based on a NN trained on the 1 μs MD data of chain length 6 to 8 (labeled with t) and a uniform fake distribution, in comparison with MD data after 1 and 5 μs of simulation. The KS results are shown in the top panels where NN MC vs 1 μs MD is shown in blue, NN MC vs 5 μs MD is shown in orange, and 1 μs MD vs 5 μs MD is shown in green.

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

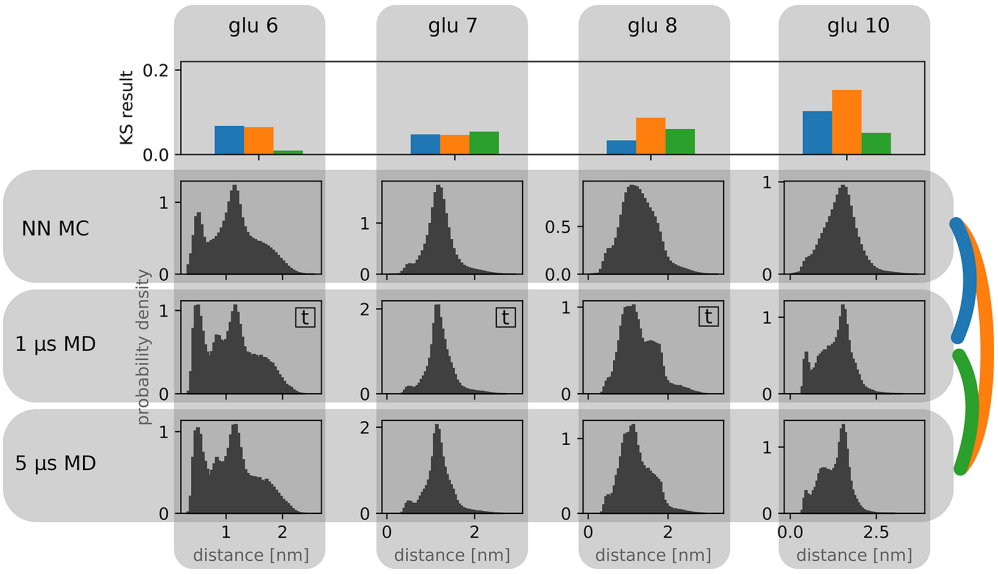


Figure 7. End-to-end distance distributions of MC simulations based on a NN trained on the 1 μs MD data of chain length 6 to 8 (labeled with t) and the NN MC distributions of the first run as fake distributions, in comparison with MD data after 1 and 5 μs of simulation. The KS results are shown in the top panels where NN MC vs 1 μs MD is shown in blue, NN MC vs 5 μs MD is shown in orange, and 1 μs MD vs 5 μs MD is shown in green.

$$p_A = \min\left(1, \exp\left(-\frac{F(R_{new}) - F(R_{old})}{kT}\right)\right) \quad (12)$$

where p_A is the acceptance probability, and R_{old} and R_{new} are the variables describing the conformation before and after the proposed move. These MC simulations based on the NN predictions allow us to compare different measures like end-to-end distance distributions in the MD sampling as well as in the NN MC sampling.

3. RESULTS AND DISCUSSION

To achieve transferability over different chain lengths, the NN network is trained with different chain lengths simultaneously. An example for training with a single chain length is not discussed in the main article but can be found in the Supporting Information (Figure S1). Here, conformations from 1 μs MD trajectories of penta-, hexa-, and heptaglutamic acid are used as data distribution, and uniformly distributed conformations of the same chain lengths are used as fake distribution. For an initial evaluation of the agreement of the conformational sampling in the MC simulations based on the NN prediction with the atomistic MD data we use end-to-end distance distributions. They are based on the side chain carboxylic acid C atoms, as these are the only positions that are also present in the coarse-grained representation. Figure 6 shows these end-to-end distance distributions. Additionally, it contains the Kolmogorow-Smirnow statistic (KS)[40] for different pairs of distributions which is a measure for the similarity of two distributions. It is given by the maximal deviation between the cumulative sums over the two distributions normalized to one. A low number is an indicator for a small deviation of the two distributions. The comparison shows that this first NN was able to learn fundamental knowledge about the molecule conformations. The maxima of the distributions are at the correct positions, and the overall width of the distributions is correct. However, the distributions are comparatively washed out, with less pronounced features such as distinct peaks and shoulders. This is also reflected in the KS results, where we find larger deviations between NN MC and the 1 μs training data than between MD data of 1 and 5 μs length (which were produced as control data). Also for chain length 10 we find a similar behavior. The NN MC distribution is not as structured, but still the maximum is located at the correct position. This is the case although chain length 10 was not part of the training data. The prediction for length 10 is entirely based on the knowledge learned from lengths 6 to 8. The deviations from 1 μs MD to 5 μs MD indicate convergence problems in the MD simulations. Especially for longer chains these sampling issues become evident which underlines the need for coarse-grained models.

One possibility to improve the result is to do a second training run on top of the first run. The NN MC conformations sampled in the first run are used as a fake distribution for the second run. The task of the NN in the second run is then to distinguish between conformations generated by the first run and conformations actually sampled in the MD simulation. This way, the NN in the second training run can concentrate on details the first run missed. The subsequent NN MC simulation is then based on the sum of the free energy differences predicted in the first and second run. Figure 7 shows the results after the second run. This time the NN MC distributions are much closer to the MD distributions. This is also confirmed by the KS results. The deviations between the NN MC and the MD distributions are now in the same range as the deviation between 1 μs MD and 5 μs MD. For length 10 the deviations are slightly larger, but the similarity is still astonishing considering the fact that length 10 was not part of the training data.

One can still observe that some finer details in the end-to-end distributions are not captured in the NN MC (for example the peak at 0.8 nm for length 6). This is probably because these are specific features of the given chain length. As the NN is trained on multiple chain lengths simultaneously it cannot focus on features of a single chain length but is forced to generalize between multiple chain lengths. How well this works

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

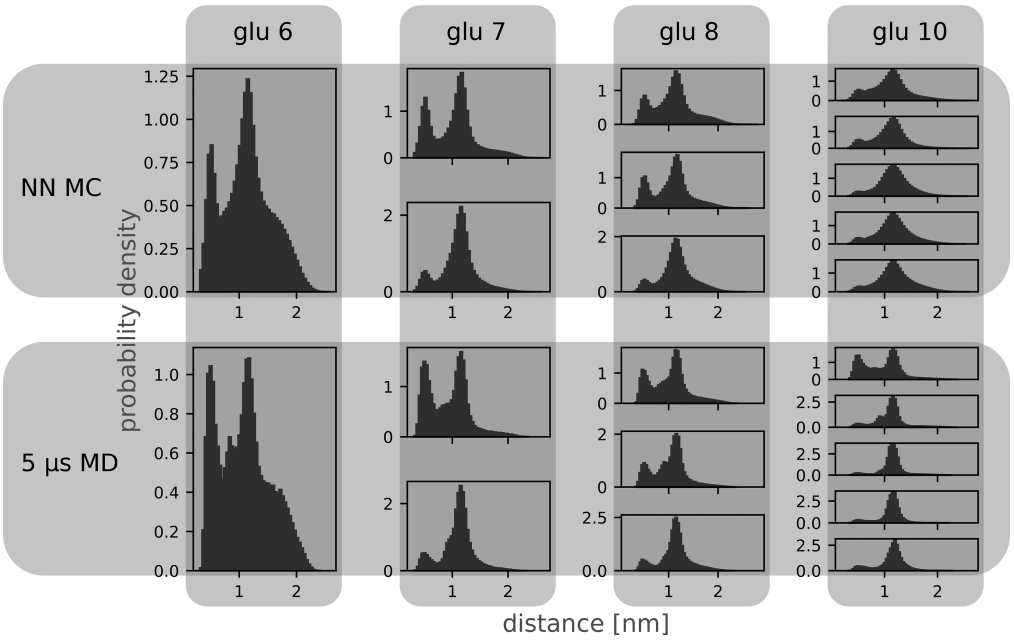


Figure 8. End-to-end distributions of hexamer segments in different chain lengths. For a given chain length the hexamer segments are ordered form the N-terminus to the C-terminus from top to bottom.

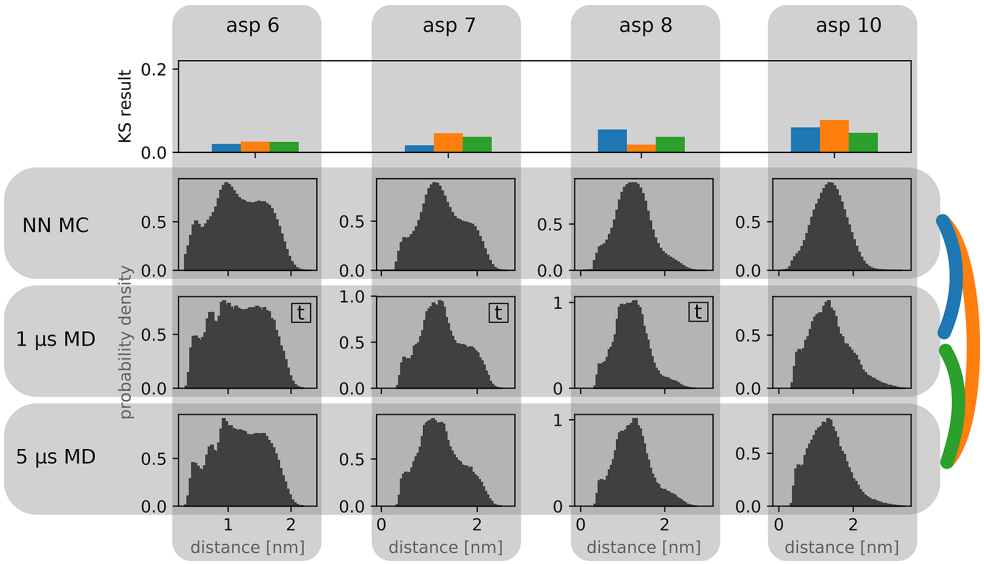


Figure 9. End-to-end distance distributions for NN MC trained on 1 μs MD data of asp-6 to asp-8. This figure is in complete analogy to Figure 7 except oligo-asp was used instead of oligo-glu.

out for different training runs with different initial weights is shown in the Supporting Information (Figure S2).

To get further insight into the similarities and differences of the distributions, it is interesting to look not only at the end-to-end distances of the complete peptide chain but also at other internal distance distributions. We focus our analysis on end-to-end distributions of hexamer segments within the chains of different lengths. For chain length 6 there is obviously only one hexamer segment. For chain length 7 there are two overlapping hexamer segments, for chain length 8 there are three, and so on. The distance distributions of all possible hexamer segments in the different chain lengths are shown in Figure 8. We chose hexamer segments for this analysis because when the NN moves along the chain, it gets information from a hexamer segment at a time. This is because the inputs are based on tetramer segments, and with the convolutional layer with filter size 3 the NN is able to combine information on three tetramer segments. As shown in Figure 8 it is not simply the case that all hexamer segments regardless of their position in the chain have the same distance distribution. This makes it a challenging task to describe multiple chain lengths with the same model. On the other hand, all the hexamer distributions also have features in common. They are dominated by two peaks, one at around 0.5 nm and one at 1.2 nm. The main difference between the different hexamer segment distributions is the relative height of these two peaks. This is for example the case for chain length 7.

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

In the hexamer segment at the N-terminus of the peptide (shown on top) both peaks have almost equal height. In the hexamer segment at the C-terminus (shown in the bottom) the peak at 1.2 nm is clearly dominating. This is a feature that is excellently captured in the NN MC simulation. There is a similar behavior for the hexamer distributions in chain length 8. In the case of chain length 10, the hexamer distributions are dominated by the peak at 1.2 nm even more drastically. This cannot be observed for any of the shorter peptides to this extent. Even though the NN was only trained on the lengths 6−8, it correctly predicts this behavior for length 10. The peaks are not as narrow as in the MD reference, but still the general shape is captured very well. This shows that the NN did not just memorize hexamer distributions but must have captured some elementary relationships which are decisive to evaluate peptide conformations.

Oligo glutamic acids are molecules which can be found in helical conformations most of the time. This is reflected in narrow distance distributions. An exception is given by glu-6 which tends to form more disordered structures which is reflected in broader distance distributions.

Despite the structural similarity to glutamic acid, aspartic acid oligomers exhibit a distinctly different behavior. The aspartic acid oligomers are generally more disordered compared to glutamic acid oligomers. The exact same approach with two training runs as described above for glutamic acid peptides was also applied to equivalent oligo-asp data. The results are shown in Figure 9. Also, for this more disordered system there is a very good agreement between the MD distributions and the NN MC distributions. As the KS statistics show, the deviations of the NN MC distributions compared to the MD distributions are in the same range as the deviation between 1 μs MD and 5 μs MD distributions. Again, the NN was only trained on the 1 μs data of the lengths 6 to 8. Therefore, it is not surprising that the deviations in the case of length 10 are slightly larger. Still, the deviations are small, and the maximum is again at the correct position.

4. CONCLUSIONS

We have shown that it is possible to train a neural network to predict conformational free energies by creating a classification problem between real MD conformations and fake conformations of a known distribution. With the convolutional network architecture based on short chain segments, it is not only possible to train a NN on a single peptide length but also simultaneously train the same NN with data of different chain lengths. After training the NN on different chain lengths it is even able to make meaningful predictions for chain lengths that were not part of the training data. The analysis of hexamer segments in different chain lengths showed that these hexamer segments in short chains behave quite differently compared to the segments in the middle of a longer chain. Still, the NN performed surprisingly well predicting the free energies for the longer chain length. To get more accurate predictions for longer chain lengths it would be helpful to extend the training data to a length where the behavior of chain segments in the middle of the chain no longer depends on the chain length. For longer chains it is harder to get converged MD data. One could use some enhanced sampling technique to speed up sampling of the MD simulation. However, it might not even be necessary to reach convergence for each chain length of the training set, as long as there is enough data for all lengths combined.

The two chosen test systems, oligo-glu and oligo-asp peptides, demonstrate that the method is able to deal with disordered as well as with highly structured systems. In the next step, we will combine the sampling of the NN MC ensemble with suitable backmapping strategies, to reintroduce atomistic degrees of freedom of the peptide chains. This will open up a venue toward well-equilibrated atomistic structures of long polymers which are not easily accessible by atomistic simulations alone. To make use of the learned peptide conformations in more realistic systems, it would be interesting to combine the NN predicted free energies of the peptide with external influences such as ions interacting with the peptide. Such a combination of NN free energies with other interaction potentials will open up paths to explore various systems of interest and to reuse the learned peptide conformations in different environments.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.7b00864.

Figure S1, comparison of end-to-end distance distributions for simulations based on a NN trained solely on glutamic acid heptamer; Figure S2, comparison of multiple training runs with different initial NN weights (PDF)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: Christine.Peter@uni-konstanz.de.

ORCID

Tobias Lemke: 0000-0002-0593-2304

Christine Peter: 0000-0002-1471-5440

Funding

Financial support by the DFG (SFB1214) is gratefully acknowledged. This work was performed on computational resources funded within the bwHPC program by the state of Baden-Württemberg and the DFG.

Note added after proof

Note that during proof stage a recently published paper has come to our attention [41]. Schneider et al. use a neural network-based approach to represent conformational free-energy landscapes in classical simulations relying on free energy or gradient data which are generated via enhanced-sampling methods.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Simon Hunkler and Nele Klinkenberg for proofreading and helpful discussions.

■ REFERENCES

(1) Noid, W.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*, 244114.

(2) Tozzini, V. Minimalist models for proteins: a comparative analysis. *Q. Rev. Biophys.* **2010**, *43*, 333−371.

(3) Reith, D.; Pütz, M.; Müller-Plathe, F. Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.* **2003**, *24*, 1624−1636.

(4) Shell, M. S. Coarse-graining with the relative entropy. *Adv. Chem. Phys.* **2016**, 395−441.
(5) Noid, W. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **2013**, *139*, 090901.
(6) Potestio, R.; Peter, C.; Kremer, K. Computer simulations of soft matter: Linking the scales. *Entropy* **2014**, *16*, 4199−4245.
(7) Wagner, J. W.; Dama, J. F.; Durumeric, A. E.; Voth, G. A. On the representability problem and the physical meaning of coarse-grained models. *J. Chem. Phys.* **2016**, *145*, 044108.
(8) Ischtwan, J.; Collins, M. A. Molecular potential energy surfaces by interpolation. *J. Chem. Phys.* **1994**, *100*, 8080−8088.
(9) Babin, V.; Leforestier, C.; Paesani, F. Development of a "first principles" water potential with flexible monomers: Dimer potential energy surface, VRT spectrum, and second virial coefficient. *J. Chem. Theory Comput.* **2013**, *9*, 5395−5403.
(10) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.
(11) Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **1995**, *103*, 4129−4137.
(12) Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* **2004**, *395*, 210−215.
(13) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.
(14) Handley, C. M.; Popelier, P. L. Potential energy surfaces fitted by artificial neural networks. *J. Phys. Chem. A* **2010**, *114*, 3371−3383.
(15) Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930−17955.
(16) Das, P.; Moll, M.; Stamati, H.; Kavraki, L. E.; Clementi, C. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103*, 9885−9890.
(17) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 13023−13028.
(18) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, 015102.
(19) Modha, D. S.; Fainman, Y. A learning law for density estimation. *IEEE Transactions on Neural Networks* **1994**, *5*, 519−523.
(20) Galvelis, R.; Sugita, Y. Neural Network and Nearest Neighbor Algorithms for Enhancing Sampling of Molecular Dynamics. *J. Chem. Theory Comput.* **2017**, *13*, 2489−2500.
(21) Garrido, L.; Juste, A. On the determination of probability density functions by using Neural Networks. *Comput. Phys. Commun.* **1998**, *115*, 25−31.
(22) Gutmann, M.; Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010; pp 297−304.
(23) Weiner, S.; Addadi, L. Design strategies in mineralized biological materials. *J. Mater. Chem.* **1997**, *7*, 689−702.
(24) Meldrum, F. C.; Cölfen, H. Controlling Mineral Morphologies and Structures in Biological and Synthetic Systems. *Chem. Rev.* **2008**, *108*, 4332−4432. PMID: 19006397.
(25) Kahlen, J.; Peter, C.; Donadio, D. Molecular simulation of oligo-glutamates in a calcium-rich aqueous solution: insights into peptide-induced polymorph selection. *CrystEngComm* **2015**, *17*, 6863−6867.
(26) Harada, A.; Kataoka, K. Formation of polyion complex micelles in an aqueous milieu from a pair of oppositely-charged block copolymers with poly (ethylene glycol) segments. *Macromolecules* **1995**, *28*, 5294−5299.
(27) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19−25.
(28) Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; van Gunsteren, W. F. Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur. Biophys. J.* **2011**, *40*, 843−856.
(29) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.
(30) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, 014101.
(31) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81*, 3684−3690.
(32) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(33) Hess, B.; Bekker, H.; Berendsen, H. J. C.; Fraaije, J. G. E. M. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18*, 1463−1472.
(34) Bishop, C. M. *Neural networks for pattern recognition*; Oxford University Press: 1995.
(35) Ripley, B. D. *Pattern recognition and neural networks*; Cambridge University Press: 2007.
(36) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278−2324.
(37) Kingma, D.; Ba, J. *Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980* 2014.
(38) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*; 2015. Software available from tensorflow.org (accessed Nov 15, 2017).
(39) Ng, A. Y. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. Proceedings of the twenty-first international conference on Machine learning. 2004; p 78, DOI: 10.1145/1015330.1015435.
(40) Kolmogorov, A. N. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari* **1933**, *4*, 83−91.
(41) Schneider, E.; Dai, L.; Topper, R. Q.; Drechsel-Grau, C.; Tuckerman, M. E. Stochastic Neural Network Approach for Learning High- Dimensional Free Energy Surfaces. *Phys. Rev. Lett.* **2017**, *19* (15), 150601.

DOI: 10.1021/acs.jctc.7b00864
J. Chem. Theory Comput. 2017, 13, 6213−6221

Supporting Information for "Neural network based prediction of conformational free energies - a new route towards coarse grained simulation models"

Tobias Lemke and Christine Peter*

Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Germany

E-mail: Christine.Peter@uni-konstanz.de

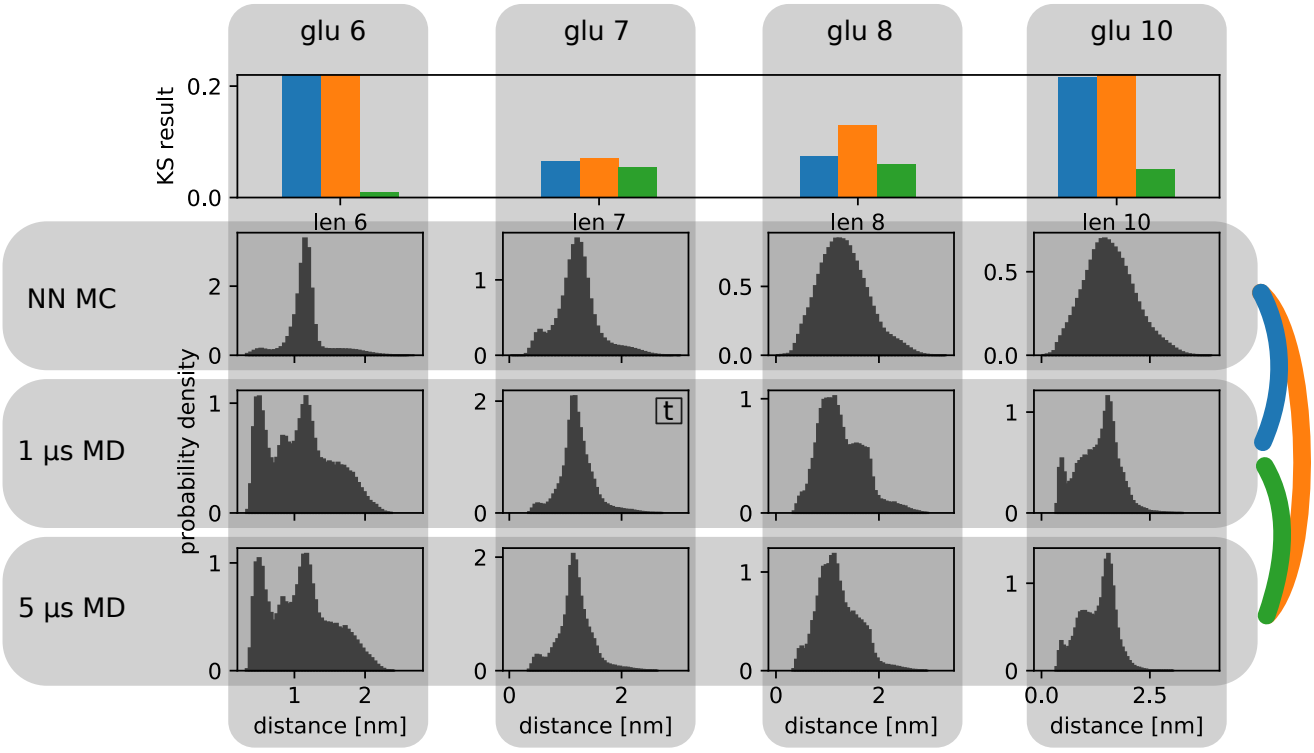


Figure S1: End-to-end distance distributions of MC simulations based on a NN trained on the 1 μs MD data of chain length 7 (labeled with t) and the NN MC distributions of a first run as fake distributions, in comparison with MD data after 1 and 5 μs of simulation. The KS results are shown in the top panels where NN MC vs. 1 μs MD is shown in blue, NN MC vs. 5 μs MD in orange, and 1 μs MD vs. 5 μs MD in green. This data was produced in complete analogy to the data presented in the main article. The only difference is that only glu7 data was used for training and not the data of glu6 - glu8. Using only glu7 data for training, other chain lengths can not be predicted very accurately. Especially for glu6 differences in the in the end-to-end distance distribution are obvious. This behavior is not surprising as glu7-glu10 are mainly characterized by helical structures and glu6 is still too short to form a stable helix. Accordingly, the predictions for glu8 and glu10 are not as bad. The end-to-end distance distribution for glu7 (the lenght the NN was trained on) turns out decently well but also not better compared to the training procedure with the glu6-glu8 data. This indicates that the NN structure is flexible enough to express the FES of multiple chain lengths.

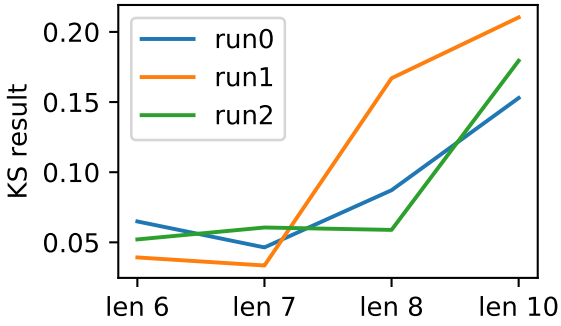


Figure S2: Comparison of the KS results NN MC vs. 5 μs MD for repetitions of the two step learning procedure based on the glu6 - glu8 data. Run0 is the run described in the main article. The Figure shows that all runs result in NNs with an overall very good performance. In detail, the performance of the different runs is slightly different. Run1 for example seams to perform better for shorter chains but worse for longer ones. Run2 turned out to be very similar to the initial run (run0). To address this variance of trained NNs one could train multiple NNs and either select the ones which perform best or combine the predictions of multiple NNs.

JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. 2019, 15, 1209−1215

pubs.acs.org/JCTC

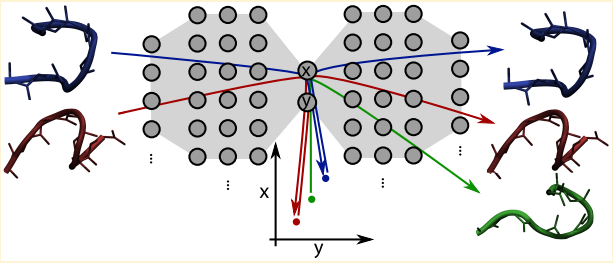
EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations

Tobias Lemke and Christine Peter*

Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Germany

Supporting Information

ABSTRACT: Molecular simulation is one example where large amounts of high-dimensional (high-d) data are generated. To extract useful information, e.g., about relevant states and important conformational transitions, a form of dimensionality reduction is required. Dimensionality reduction algorithms differ in their ability to efficiently project large amounts of data to an informative low-dimensional (low-d) representation and the way the low and high-d representations are linked. We propose a dimensionality reduction algorithm called EncoderMap that is based on a neural network autoencoder in combination with a nonlinear distance metric. A key advantage of this method is that it establishes a functional link from the high-d to the low-d representation and vice versa. This allows us not only to efficiently project data points to the low-d representation but also to generate high-d representatives for any point in the low-d map. The potential of the algorithm is demonstrated for molecular simulation data of a small, highly flexible peptide as well as for folding simulations of the 20-residue Trp-cage protein. We demonstrate that the algorithm is able to efficiently project the ensemble of high-d structures to a low-d map where major states can be identified and important conformational transitions are revealed. We also show that molecular conformations can be generated for any point or any connecting line between points on the low-d map. This ability of inverse mapping from the low-d to the high-d representation is particularly relevant for the use in algorithms that enhance the exploration of conformational space or the sampling of transitions between conformational states.



1. INTRODUCTION

With modern computational resources the amount and complexity of data that can be generated, processed, and stored grows rapidly. At the same time, we as humans remain unfit to grasp such complex data directly, and one of the key research questions is how to condense massive amounts of data into a human understandable form.

Molecular simulation is one such example where huge data sets are generated.[1] The primary result of such a simulation is typically a trajectory containing atom positions of a given (bio)molecular system for multiple (time) frames. With three spatial coordinates per atom, each frame of the trajectory is a point in a phase space with easily several thousands or millions of dimensions. Not all of these are equally important for mechanistic interpretation. For a protein in aqueous solution it is, for example, possible to describe much of a conformation through the backbone-dihedral angles. With two backbone-dihedral angles Φ and Ψ per amino acid, this still leads to a representation with tens or hundreds of dimensions. To make sense of such data, i.e., to characterize processes such as folding or conformational transitions and to identify relevant (conformational) states, one needs to further reduce the dimensionality.

Besides "simply guessing" relevant individual or collective variables that are well suited to describe processes or distinguish states, a whole variety of systematic dimensionality reduction approaches are available: algorithms such as principle component analysis (PCA),[2] sketch-map,[3,4] diffusion maps,[5] or time-lagged independent component analysis (TICA)[6] have been successfully established in the simulation community. There are different aspects that determine how well a given dimensionality reduction algorithm is suited for the data at hand. For molecular simulation data, three relevant criteria are (1) how informative is the low-d representation, and how well is it able to separate the data points into distinct states, (2) how fast is it, and (3) how are the high-d and low-d representations linked?

An efficient link from the high-d to the low-d representation is important whenever additional data points should be projected to the low-d representation. This is especially important if dimensionality reduction is not performed solely for analysis purposes. For example, biasing of simulations for enhanced sampling purposes requires a link from the high-d to the low-d representation that allows to apply a biasing potential/force defined in the low-d representation.[7,8] Likewise, a link in the opposite direction, i.e., the ability to generate high-d coordinates corresponding to a given point in the low-d representation, is highly desirable, for example, to accelerate/enhance the sampling of phase space by initiating new

Received: September 27, 2018
Published: January 11, 2019

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

simulations from specific regions or states. Such inverse mapping or back-mapping problems from a coarse representation back to a more detailed representation are well-known in scale-bridging between simulation levels,.[9,10]

The various dimensionality reduction algorithms exhibit particular individual strengths (and weaknesses) with regard to the above criteria. Although the expressiveness of the low-d representation is a vague and hard to define criterion, it is clear that linear techniques such as PCA lack the capability to unravel inherently nonlinear features of the data as found, for example, in the folding of proteins.[11] Here nonlinear techniques such as sketch-map[3] have proven to be more suitable. Sketch-map is a multidimensional-scaling like algorithm,[12] which aims for a reproduction of the (high-d) pairwise distances between data points in the low-d representation. The main idea behind sketch-map is that not all of these pairwise distances are equally important. If one imagines the high-d molecule conformations to be associated in clusters connected in a spider-web like fashion, then small distances originate from fluctuations inside such clusters. Long distances originate from the relative position of these clusters in high-d space but cannot be well represented in low-d. Intermediate distances contain information about neighboring clusters and are therefore most important to capture the connectivity of a high-d network. Sketch-map therefore transforms the pairwise distances with sigmoid functions to place the focus on intermediate distances where these sigmoid functions are tuned to be most steep. A low-d representation is obtained by minimizing the following cost function:

$$C_{\text{sketch}} = \frac{1}{m} \sum_{i \neq j} [\text{SIG}_{\text{h}}(R_{ij}) - \text{SIG}_{\text{l}}(r_{ij})]^2 \tag{1}$$

where m is the number of pairwise distances, R_{ij} are the pairwise distances in the high-d space, r_{ij} are the pairwise distances in the low-d space, and SIG_{h} and SIG_{l} are sigmoid functions of the following form:

$$\text{SIG}_{\sigma,a,b}(r) = 1 - (1 + (2^{a/b} - 1)(r/\sigma)^a)^{-b/a} \tag{2}$$

where σ defines the location of the inflection point of the sigmoid and a and b determine how quickly the function approaches 0 and 1, respectively. Minimizing this cost function is computationally expensive and scales unfavorably because the number of pairwise distances grows quadratically with the number of data points. The sketch-map algorithm overcomes this scaling problem by performing the dimensionality reduction with a representative subset of the data, so-called landmarks. The remaining data points are then projected into the resulting low-d map. Note that the inverse mapping, i.e., the reconstruction of high-d coordinates given a point in the low-d representation is not easily possible because no functional relationship between high-d (atomistic) and low-d sketch-map coordinates exists. In summary, sketch-map is a nonlinear method that can nicely capture quite complex network like features in a low-d representation, however, it has limitations in the efficiency of finding the low-d representation and projecting data in and out of the low-d representation.

Neural network autoencoders,[13] which have just recently come to the attention of the simulation community,[8,14−17] are quite the opposite. Autoencoders have their strengths where sketch-map has its weaknesses. The main concept of an autoencoder is to train an artificial neural network to reproduce its inputs. This seemingly pointless undertaking becomes useful by introducing a bottleneck in the middle of the neural network. The autoencoder can then be viewed as consisting of two parts. One encoder part that is trained to encode as much information as possible into a low-d representation and a decoder part that is trained to reconstruct the initial high-d input from this low-d representation. Autoencoders are usually trained by gradient descent with mini-batches of data. In other words, creating the low-d representation has a very favorable linear scaling with the number of data points used. It is therefore not necessary to select landmarks. Instead, large amounts of data can be used to obtain the low-d representation. Also, due to the recent boom in machine learning, there are highly optimized computational libraries like Torch,[18] Theano,[19] or TensorFlow[20] available that make the training procedure fast and efficient. Once the autoencoder is trained, the encoder part of the network is a differentiable mathematical function that directly maps high-d inputs to the low-d space. This makes projecting additional points into the low-d representation very efficient and, for example, allows to use the low-d representation for enhanced sampling with biasing techniques[8] like umbrella sampling[21] or metadynamics.[22,23] With the decoder part of the network an autoencoder also provides a mathematical function to reconstruct a high-d point given any point in the low-d representation. A drawback of autoencoders, however, is how data are expressed in the low-d representation. During the training process the low-d representation is optimized in such a way that the decoder part of the network can reproduce the data most accurately. This is not necessarily the most intuitive or informative way to present the data to humans.[24] Note that distances between data points are not necessarily preserved in the low-d space.

Here we propose a dimensionality reduction algorithm called EncoderMap that combines the advantages of autoencoders and the sketch-map cost function. The autoencoder provides the efficient functional links between the high-d and low-d representations and the nonlinear distance metric-based cost function forces the autoencoder to arrange points in the low-d representation in a meaningful way. In the following we explain the method and show and discuss results for two example data sets from molecular dynamics simulations: The small peptide aspartic acid heptamer (Asp-7) and the mini protein Trp-cage (PDB 1L2Y).[25] The method, however, is not limited to such data.

2. ENCODERMAP

EncoderMap unites the advantages of two methods in a single algorithm. It combines a neural network autoencoder[13] with the cost function of sketch-map.[3] The autoencoder used in this work consists of several fully connected layers: an input layer, several hidden layers, a bottleneck layer with few neurons, further hidden layers, and an output layer with the same number of neurons as in the input layer (see Figure 1). In a fully connected layer the activation of each neuron is calculated as a weighted sum of the activations of all neurons in the previous layer to which an activation function is applied. This can efficiently be noted and calculated in the form of a matrix multiplication:

$$\mathbf{a}_l = f_l(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l) \tag{3}$$

where $\mathbf{a}_l$ are the activations of layer l, f_l is the activation function, $\mathbf{W}_l$ is a matrix of weights, $\mathbf{a}_{l-1}$ are the activations of

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

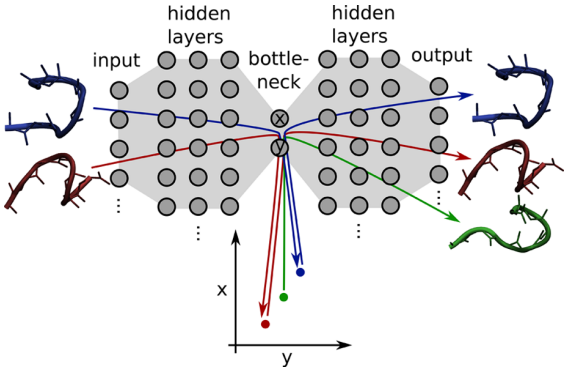


Figure 1. Example autoencoder architecture with two bottleneck neurons. This results in a projection of high-d data, e.g., molecule conformations, to points in a 2-d space. The decoder part (right-hand side) can generate output conformations for 2-d data points for which no corresponding input exists (green point/conformation).

the previous layer, and $\mathbf{b}_l$ are bias values added. The activation function is important to add nonlinearity to the network, and thus the flexibility to fit nonlinear features. The weights and biases are the parameters that are tuned during the training procedure to minimize a cost function. For an autoencoder this cost function naturally involves some distance measure between the inputs and outputs of the neural network:

$$C_{\text{auto}} = \frac{1}{n}\sum_{i=1}^{n} \text{dist}(\mathbf{x}_i, \tilde{\mathbf{x}}_i) \tag{4}$$

where n is the number of training examples, x_i is the vector of inputs for the ith training example, $\tilde{\mathbf{x}}_i$ is the vector of outputs of the autoencoder and dist is some distance metric. With this cost function, the network is trained to reproduce the inputs while it is forced to encode the information in a few dimensions due to the bottleneck in the network. How the information is encoded, however, is very arbitrary. To make this low-d representation better defined and interpretable, EncoderMap combines the autoencoder cost function (eq 4) with the sketch-map cost function (eq 1):

$$C = k_a C_{\text{auto}} + k_s C_{\text{sketch}} + \text{REG} \tag{5}$$

The sketch-map cost function is based on the pairwise distances between data points in the high-d input space and the low-d space given by the bottleneck layer. The constants k_a and k_s allow us to shift the priority between minimizing the autoencoder and the sketch-map cost and REG is a regularization term. Regularization is used to adjust the complexity or roughness of a neural network to prevent overfitting. Here we used the following regularization term:

$$\text{REG} = \frac{k_{l2}}{2}\sum_i w_i^2 + \frac{k_c}{n}\sum_n \sum_j a_{b,j,n}^2 \tag{6}$$

where the first term restricts the weights w_i of the network and the second term restricts the activation in the bottleneck layer a_b to prevent the low-d representation to be shifted toward large numbers. k_{l2} and k_c are constants to scale the influence of these terms. The network is trained using the Adam optimizer.[26] As the training is done with batches of data, no selection of landmarks is necessary. Large amounts of data can be used for training one batch at a time.

In the following examples we use the backbone dihedral angles, Φ and Ψ, as input for EncoderMap as they are

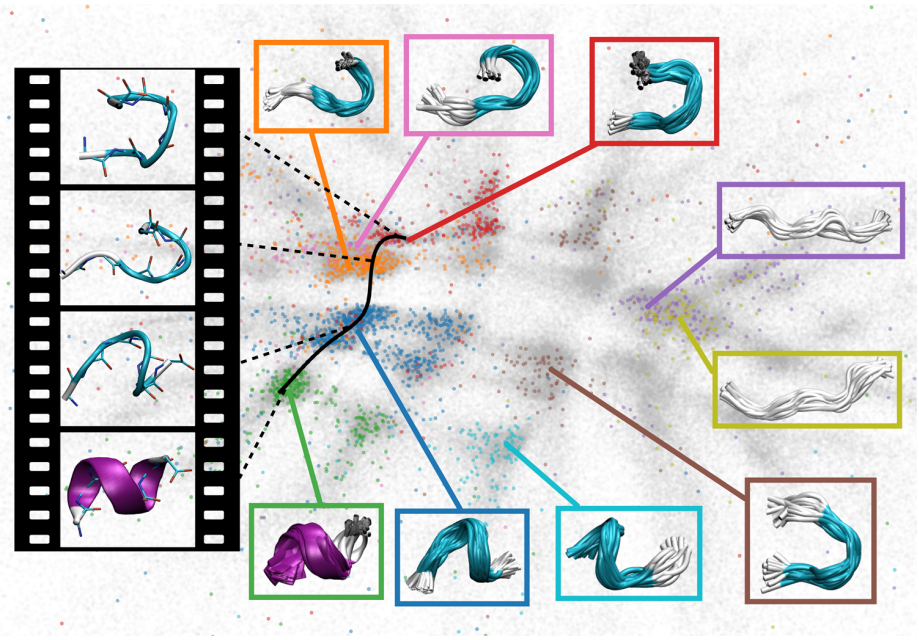


Figure 2. 2-d EncoderMap of Asp-7. Colored dots represent conformations from the nine largest clusters in an rmsd-based clustering.[29] Superimposed molecule conformations from these clusters are shown in the respectively colored frames. All other conformations are represented as gray dots. The black line shows the path along which molecule conformations were generated with the decoder part of the network. Some snapshots of these generated structures are shown in the film strip on the left. The complete sequence of all generated conformations can be found in Movie S1. A comparison of this EncoderMap projection with projections from other dimensionality reduction techniques can be found in Figure S1. Further analysis based on the Asp-7 data set are shown in Figures S2, S3, and S4.

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

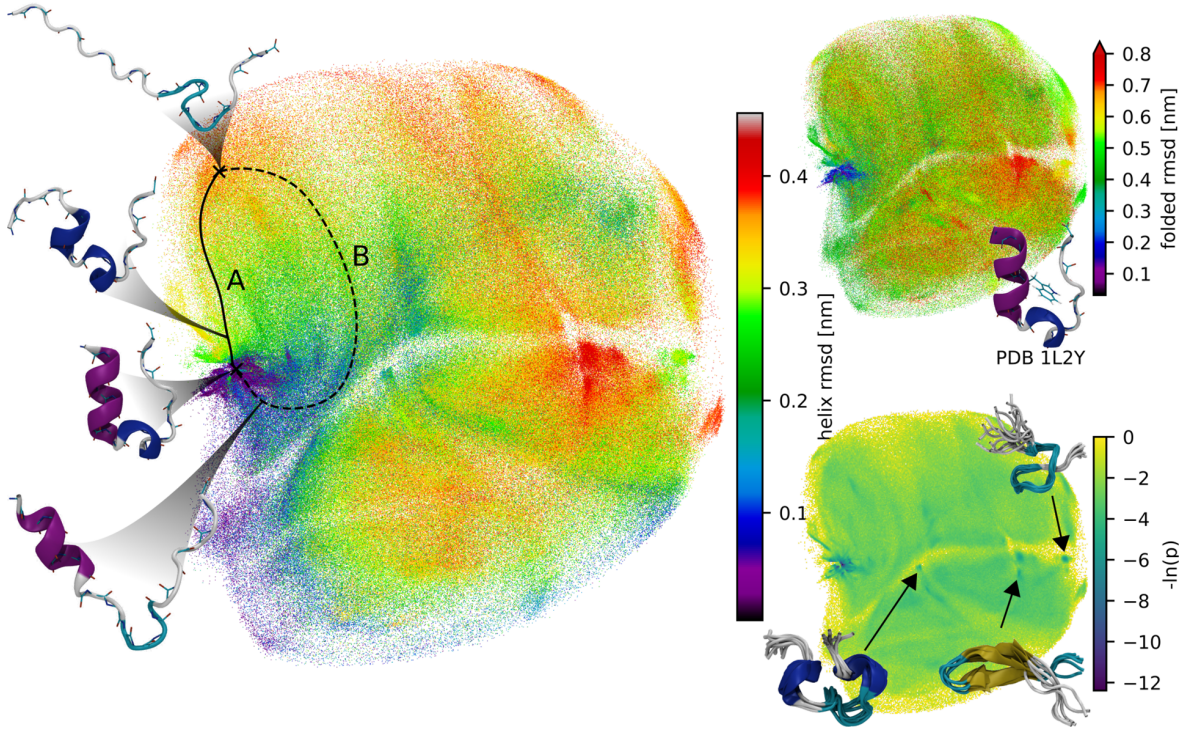


Figure 3. EncoderMap of Trp-cage in different colorings. The top-right map is colored according to the C_α rmsd compared to the shown native structure. The bottom-right map is a 2-d-histogram of the map with 300 × 300 bins colored according to the negative natural logarithm of the bin counts. Superimposed structures from selected dense regions outside the main folded region are shown. The map on the left is colored according to the C_α rmsd of residues 2−8 compared to an α-helical structure. Molecule conformations where generated for points on the two paths shown as solid and dashed lines. Selected generated conformations are shown in the plot and all generated conformations can be found in Movie S3 and Movie S4.

rotationally and translationally invariant and independent from each other. At the same time, they describe the conformation of the peptide backbone almost completely and allow simple reconstruction of a backbone conformation starting with an extended conformation by rotation around the respective dihedral axis. The periodicity of these inputs, however, adds some complexity to the approach. To incorporate periodicity into the network we map all dihedral angles to their sin and cos values before passing them to the first hidden layer. This transforms each angle into two nonperiodic Cartesian coordinates of a point on a unit circle. Pairs of activations of the output layer are then also treated as such Cartesian unit circle coordinates and are mapped back to an angle. Also, the distance metric used for the cost function has to respect periodicity. Here, we use the smallest euclidean distance in a periodic space. With these adjustments EncoderMap works well with periodic inputs like dihedral angles as we show in the following examples.

3. RESULTS AND DISCUSSION

3.1. Example 1: Asp-7. For a first test of EncoderMap, we chose an example of medium complexity. Aspartic acid heptamer with its 12 backbone dihedrals has a dimensionality well beyond what is directly graspable and is still a fairly simple molecule. From our previous studies of this system,[27,28] we know that it exhibits a variety of states and only a low fraction of completely folded helical structures. The data set used consists of 500 000 molecule conformations from a 5 μs molecular dynamics simulation. The EncoderMap was obtained on the basis of the 12 dihedral angles of each conformation. This takes around 3 min on a desktop computer (i7-8700K GTX 1080) with our code provided on https://github.com/AG-Peter/EncoderMap. Details about the EncoderMap training procedure are given in the Method Details section. The resulting 2d EncoderMap is shown in Figure 2. All peptide conformations of the data set are represented as gray dots. For comparison we performed the C_α root-mean-square deviation (rmsd) based gromos clustering[29] with a cutoff of 0.1 nm on a subset of 10 000 conformations. The conformations assigned to the nine largest clusters are represented with dots colored according to their cluster numbers. Overlain conformations from each cluster are shown in the accordingly colored frames. Figure 2 shows that EncoderMap nicely separates the conformations from different clusters. Some clusters such as the blue or green ones are further separated into subclusters. Furthermore, clusters with similar structures are located in proximity to each other and connections between these clusters become visible. For example, the green cluster that represents the helical structure is placed close to the two blue clusters where fragments of the helix are already present. One of the big advantages of the EncoderMap algorithm is that it not only allows us to efficiently generate expressive low-d representations of large data sets but also the reconstruction of high-d data based on any point in the low-d map. This works also for new points that were not part of the training data. To illustrate this feature, 450 equally spaced points on the black path shown in Figure 2 were fed into the decoder part of the autoencoder. The decoder yields 12 dihedral angles for each of these 2-d points. On the basis of

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

these dihedral angles backbone conformations can be reconstructed by simple rotation around the respective dihedral axis. Selected generated conformations are shown in the filmstrip depicted in Figure 2. The generated conformations nicely resemble the conformations of the clusters through which the path was drawn, showing that the decoder predicts realistic conformations for regions where reference data exist. Between the clusters, in regions where only few conformations are present in the original data set, the neural network smoothly interpolates the conformations. This is demonstrated in Movie S1 where all generated conformations on the black path are shown. Note that this is not meant to imply that a path that is drawn along the 2-d map necessarily corresponds to real transition paths in the simulations. Nevertheless, it demonstrates the ability of EncoderMap to generate conformations that connect given data points.

While 2D projections are most practical for visualization purposes, it is possible to create projections of any dimensionality with EncoderMap. Movie S2 shows a 3d EncoderMap of the asp-7 data set. Projections with more than 3 dimensions are impractical for visualization but might still be useful for, e.g., density-based clustering algorithms. Here, the dimensionality affects the separability of states into well identifiable clusters, but a too high dimensionality will again make the density distribution very sparse and clustering problematic. Figure S4 compares projections with different dimensionalities based on the obtained values for the autoencoder and sketch-map part of the cost function. Next, we test the EncoderMap algorithm with a more complex example.

3.2. Example 2: Trp-cage. Trp-cage (PDB 1L2Y) is a well studied 20 residue protein.[25,30] The folded structure of Trp-cage consists of an N-terminal α-helical part and a hydrophobic core around the tryptophan residue. Several computational studies have revealed two folding pathways for the Trp-cage.[31−34] One path where the helix folds after the hydrophobic core is formed and one path where the helix forms before the hydrophobic core. The data set used here consists of 1.5 million Trp-cage conformations from a temperature replica exchange simulation with 40 replicas in a range of temperatures from 300 to 570 K and a total simulated time of 3 μs. Conformations from all temperature replicas are included in the data set (a separate analysis of the lowest replica only can be found in Figure S5). In complete analogy to the Asp-7 example an EncoderMap based on the 38 backbone dihedrals of Trp-cage was obtained. Figure 3 shows the resulting 2-d map in different colorings (further colorings based on the different dihedrals are available in Figure S6). The top-right map is colored according to the C_α rmsd compared to the native folded (PDB 1L2Y) structure. This coloring reveals an area of folded structures in the left part of the map surrounded by a broad area of unfolded structures. The map in the bottom-right shows the density of points on a logarithmic scale. As to be expected, the area of fully folded structures has the highest density of points and the surrounding unfolded or partially folded region is less densely populated. Several regions of increased density can be found outside of the folded area. The overlays of conformations corresponding to three of these regions (highlighted with arrows in the figure) show that the EncoderMap is well suited for the identification of distinct (albeit rarely occurring) clusters of conformations. The map in the left is colored according to the C_α rmsd of residues 2−8 compared to an α-helical structure. This map shows that the folded area exhibits low helix rmsd values since residues 2−8 form a helix in the folded structure. The coloring reveals that this helix is also present in parts of the unfolded/partially folded regions. Approaching the area of fully folded conformations from the bottom leads through unfolded/partially folded structures where the helix is already formed. In contrast, approaching the fully folded area from the top leads through unfolded structures where no helix is present. To further illustrate theses different routes toward the folded structure we use the high-d data generation capabilities of EncoderMap. A total of 450 equidistant points on two connecting lines from a given unfolded structure to the native structure (indicated in black in Figure 3) were fed into the decoder part of the network. Some of the generated conformations are presented in Figure 3; all generated conformations along the two paths are shown in Movie S3 and Movie S4. On path A we find structures close to the folded area where the hydrophobic core is present but the helix region is unfolded. On path B, in contrast, we find structures where the helix is folded but the hydrophobic core is not collapsed. Note that while these paths do not correspond to real transitions found in the simulation, they do identify approaches toward the folded structure that are in good agreement with the known Trp-cage folding pathways from the literature.[31−34]

4. CONCLUSIONS

We have introduced EncoderMap, a dimensionality reduction algorithm that combines a neural network autoencoder with the nonlinear-distance-metric based cost function of sketch-map. The presented molecular example systems described in dihedral space demonstrate EncoderMaps ability to represent complex high-d data in an informative low-d form. Similar data points are aggregated in clusters, and important proximity relations between these clusters are nicely captured. The method is not limited to dihedrals but can be used with different kinds of data such as Cartesian coordinates or distance based measures describing molecular conformations.[35] Even with large data sets of more then a million points, creating an EncoderMap is very efficient and takes only few minutes on a common desktop computer. Additional points can be projected to the map with a differentiable function that the algorithm yields. This allows for efficient projection of additional points to the map and allows us to combine EncoderMap with enhanced sampling schemes that require biasing potentials defined in a low dimensional space. With EncoderMap, however, it is not only possible to project additional high-d data to the map. Our examples show that it is also possible to generate reasonable high-d points for any points in the low-d map. In that sense the EncoderMap can be used as an interesting type of molecular model. A model that is not defined as a high-d energy function or force-field that requires simulations to obtain probable conformations, but a model where conformations can directly be generated from selected points in the low-d map. This inverse-mapping ability opens up new avenues toward the use of EncoderMap for enhanced sampling. Search schemes where new simulations are initiated from sparsely populated or transitional regions in conformational space could benefit from generated structures slightly extrapolated away from low density regions in the probability density distribution, path sampling schemes could benefit from EncoderMap's ability to generate plausible paths between known strucures.

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

5. METHOD DETAILS

5.1. EncoderMap Parameters. EncoderMap has multiple parameters that can be tuned. For both examples in this paper, however, we use the exact same parameters, which indicates that these might be a good starting point for a broad variety of molecular systems described in dihedral space.

The autoencoder used in this work consists of 9 fully connected layers: an input layer, 3 hidden layers with 128 neurons each, a bottleneck layer with 2 neurons, another 3 hidden layers with 128 neurons each, and an output layer with the same number of neurons as in the input layer. We use tanh for all hidden layers and the identity function for all other layers as activation function. The exact number of neurons in the hidden layers is not important. There have to be enough neurons to give the network the required complexity for the task. A larger number of neurons is, except for computational efficiency, reasons usually not a problem if regularization is used to adjust the roughness of the representation. The used cost function parameters from eq 5 and eq 6 are $k_a = 1$, $k_s = 500$, $k_{l2} = 0.001$, and $k_c = 0.0001$. Figure S2 shows the influence of different k_a and k_s settings. The large k_s value chosen here emphasizes the minimization of the sketch-map part of the cost function. To ensure that accurate generation of high-d data are still given, it is helpful to analyze the deviation between inputs and outputs of the autoencoder, as shown in Figure S7. The regularization parameters can be tuned on the basis of the training and validation cost or to obtain a desired roughness of the map. A smooth map might highlight the most important differences in the data set while a rougher map might give more detail. To obtain the training and validation cost, we have performed 5-fold cross validation. The results of this cross validation study for different neural network settings are shown in Figure S3.

The used sketch-map cost function parameters from eq 2 are $\sigma = 4.5$, $a = 12$, $b = 6$ for the sigmoid applied to the high-d distances, and $\sigma = 1$, $a = 2$, $b = 6$ for the low-d distances. The sketch-map literature[3] gives more details on how to select these parameters. The cost function is minimized in 100 000 steps with batches of 256 training points by the Adam optimizer[26] with a learning rate of 0.001 and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as implemented in TensorFlow 1.8.[20]

5.2. Simulation Details Asp-7. The Asp-7 simulation was performed with Gromacs 5.1.3[36] and the GROMOS 54A7[37] force field with SPC/E water.[38] Under periodic boundary conditions, 3610 water molecules and one asp-7 molecule where all residues were selected to be in their uncharged state were simulated. The temperature was kept at 300 K with stochastic velocity rescaling.[39] The pressure was adjusted to 1 bar in with a Berendsen barostat[40] in a preceding equilibration simulation. In the production simulation no barostat was used. The leapfrog algorithm was used to integrate the equations of motion with a time step of 2 fs. Long range interactions were calculated with the particle mesh Ewald method[41] with a grid spacing of 0.12 nm and a pme-order of 4. All interactions were truncated to 1.4 nm. All bonds were constrained using the linear constraints solver algorithm[42] with an eighth-order expansion.

5.3. Simulation Details Trp-Cage. The Trp-cage simulations were performed with Gromacs 2016.3[36] and the Amber99SB-ildn force field with 3000 tip3p water molecules.[43] An equilibration run was performed starting form the folded PDB 1L2Y structure where the temperature was adjusted to 300 K with stochastic velocity rescaling[39] and the pressure was adjusted to 1 bar with a Berendsen barostat.[40] Interactions, constraints, and integrator were set in analogy to the asp-7 simulation; however, interactions were truncated at 1 nm. Based on the equilibrated folded structure, a replica exchange with 40 replicas between 300 and 570 K was set up. Exchange attempts between neighboring replicas were made every 500 simulation steps.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.8b00975.

Comparison of EnoderMap with other dimensionality reduction methods, comparison of different EncoderMap settings, further analysis of the Trp-cage EncoderMap, EncoderMap for a 3d toy data set of points distributed on the edges of a cube (PDF)

Movie containing generated conformations for asp-7 (MPG)

Movie containing 3-dimensional EncoderMap for asp-7 (MPG)

Movie containing Trp-cage conformations generated along path A (MPG)

Movie containing Trp-cage conformations generated along path B (MPG)

■ AUTHOR INFORMATION

Corresponding Author

E-mail: Christine.Peter@uni-konstanz.de.

ORCID

Tobias Lemke: 0000-0002-0593-2304

Christine Peter: 0000-0002-1471-5440

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We thank Oleksandra Kukharenko for many helpful discussions. Financial support by the DFG (SFB1214) is gratefully acknowledged. This work was performed with computational resources at the Zentrum für Datenverarbeitung of the University of Tübingen funded within the bwHPC program by the state of Baden-Württemberg and the DFG through grant no INST 37/935-1 FUGG

■ REFERENCES

(1) Dror, R. O.; Dirks, R. M.; Grossman, J.; Xu, H.; Shaw, D. E. Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.* **2012**, *41*, 429−452.

(2) Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **1901**, *2*, 559−572.

(3) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 13023−13028.

(4) Kukharenko, O.; Sawade, K.; Steuer, J.; Peter, C. Using dimensionality reduction to systematically expand conformational sampling of intrinsically disordered peptides. *J. Chem. Theory Comput.* **2016**, *12*, 4726−4734.

(5) Coifman, R. R.; Lafon, S.; Lee, A. B.; Maggioni, M.; Nadler, B.; Warner, F.; Zucker, S. W. Geometric diffusions as a tool for harmonic

analysis and structure definition of data: Diffusion maps. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 7426−7431.
(6) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, No. 015102.
(7) Tribello, G. A.; Ceriotti, M.; Parrinello, M. Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proc. Natl. Acad. Sci. U. S. A.* **2012**, *109*, 5196.
(8) Chen, W.; Tan, A. R.; Ferguson, A. L. Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.* **2018**, *149*, No. 072312.
(9) Peter, C.; Kremer, K. Multiscale simulation of soft matter systems−from the atomistic to the coarse-grained level and back. *Soft Matter* **2009**, *5*, 4357−4366.
(10) Das, P.; Frewen, T. A.; Kevrekidis, I. G.; Clementi, C. *Coping with Complexity: Model Reduction and Data Analysis*; Springer, 2011; pp 113−131.
(11) Das, P.; Moll, M.; Stamati, H.; Kavraki, L. E.; Clementi, C. Low-dimensional, freeenergy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103*, 9885−9890.
(12) Cox, T. F.; Cox, M. A. *Multidimensional scaling*; Chapman and hall /CRC, 2000.
(13) Hinton, G. E.; Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504−507.
(14) Wehmeyer, C.; Noé, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **2018**, *148*, 241703.
(15) Sultan, M. M.; Wayment-Steele, H. K.; Pande, V. S. Transferable neural networks for enhanced sampling of protein dynamics. *J. Chem. Theory Comput.* **2018**, *14*, 1887−1894.
(16) Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, No. 072301.
(17) Hernández, C. X.; Wayment-Steele, H. K.; Sultan, M. M.; Husic, B. E.; Pande, V. S. Variational encoding of complex dynamics. *Phys. Rev. E: Stat. Phys., Plasmas, Fluids, Relat. Interdiscip. Top.* **2018**, *97*, No. 062412.
(18) Collobert, R.; Bengio, S.; Marithoz, J. Torch: A Modular Machine Learning Software Library 2002.
(19) Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints abs/1605.02688*, 2016.
(20) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*; TensorFlow, 2015; https://www.tensorflow.org/.
(21) Torrie, G. M.; Valleau, J. P. Nonphysical sampling distributions in Monte Carlo freeenergy estimation: Umbrella sampling. *J. Comput. Phys.* **1977**, *23*, 187−199.
(22) Huber, T.; Torda, A. E.; Van Gunsteren, W. F. Local elevation: a method for improving the searching properties of molecular dynamics simulation. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 695−708.
(23) Laio, A.; Parrinello, M. Escaping free-energy minima. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99*, 12562−12566.
(24) Ayinde, B. O.; Zurada, J. M. Deep learning of constrained autoencoders for enhanced understanding of data. *arXiv preprint arXiv:1802.00003*, 2018.
(25) Neidigh, J. W.; Fesinmeyer, R. M.; Andersen, N. H. Designing a 20-residue protein. *Nat. Struct. Biol.* **2002**, *9*, 425.
(26) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* 2014,.
(27) Lemke, T.; Peter, C. Neural Network Based Prediction of Conformational Free EnergiesA New Route toward Coarse-Grained Simulation Models. *J. Chem. Theory Comput.* **2017**, *13*, 6213−6221.
(28) Lemke, T.; Peter, C.; Kukharenko, O. Efficient Sampling and Characterization of Free Energy Landscapes of Ion−Peptide Systems. *J. Chem. Theory Comput.* **2018**, *14*, 5476.
(29) Daura, X.; Gademann, K.; Jaun, B.; Seebach, D.; Van Gunsteren, W. F.; Mark, A. E. Peptide folding: when simulation meets experiment. *Angew. Chem., Int. Ed.* **1999**, *38*, 236−240.
(30) Qiu, L.; Pabit, S. A.; Roitberg, A. E.; Hagen, S. J. Smaller and faster: The 20-residue Trp-cage protein folds in 4 μs. *J. Am. Chem. Soc.* **2002**, *124*, 12952−12953.
(31) Juraszek, J.; Bolhuis, P. Sampling the multiple folding mechanisms of Trp-cage in explicit solvent. *Proc. Natl. Acad. Sci. U. S. A.* **2006**, *103*, 15859−15864.
(32) Kim, S. B.; Dsilva, C. J.; Kevrekidis, I. G.; Debenedetti, P. G. Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. *J. Chem. Phys.* **2015**, *142*, No. 085101.
(33) Paschek, D.; Hempel, S. Garćıa, A. E. Computing the stability diagram of the Trp-cage miniprotein. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 17754−17759.
(34) Deng, N.-j.; Dai, W.; Levy, R. M. How kinetics within the unfolded state affects protein folding: An analysis based on Markov state models and an ultra-long MD trajectory. *J. Phys. Chem. B* **2013**, *117*, 12787−12799.
(35) Berg, A.; Kukharenko, O.; Scheffner, M.; Peter, C. Towards a molecular basis of ubiquitin signaling: A dual-scale simulation study of ubiquitin dimers. *PLoS Comput. Biol.* **2018**, *14*, No. e1006589.
(36) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1*, 19−25.
(37) Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; van Gunsteren, W. F. Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur. Biophys. J.* **2011**, *40*, 843.
(38) Berendsen, H.; Grigera, J.; Straatsma, T. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.
(39) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, No. 014101.
(40) Berendsen, H. J.; Postma, J. v.; van Gunsteren, W. F.; DiNola, A.; Haak, J. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81*, 3684−3690.
(41) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(42) Hess, B.; Bekker, H.; Berendsen, H. J.; Fraaije, J. G. LINCS: a linear constraint solver for molecular simulations. *J. Comput. Chem.* **1997**, *18*, 1463−1472.
(43) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79*, 926−935.

DOI: 10.1021/acs.jctc.8b00975
J. Chem. Theory Comput. 2019, 15, 1209−1215

Supporting Information for "EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations"

Tobias Lemke and Christine Peter*

Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Germany

E-mail: Christine.Peter@uni-konstanz.de

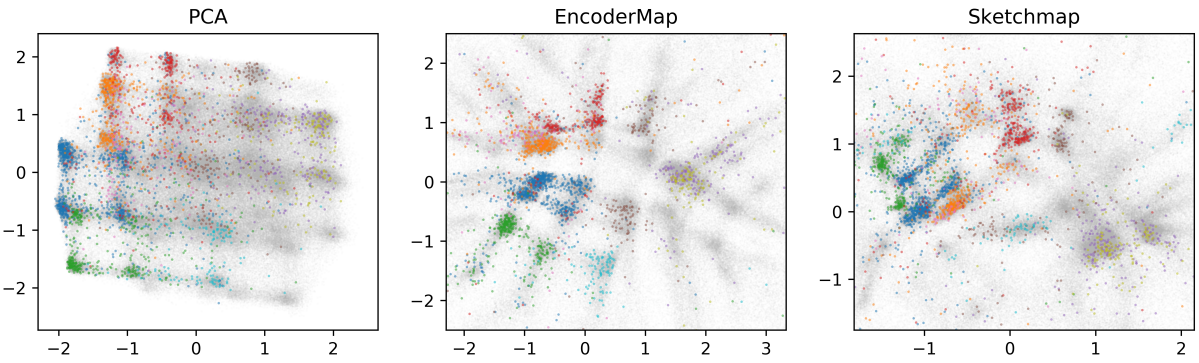


Figure S1: Comparison of different dimensionality reduction techniques for the asp-7 data set. In analogy to Figure 2, colored dots represent conformations from the rmsd-based clustering. In the left graph, a projection to the first two principal components is shown. To respect the periodicity of the dihedral data PCA was performed based on the sin and cos values of the angles. The different clusters are somewhat separated in the projection, however, they are not as clearly separated as in the EncoderMap, for example the blue cluster is not clearly separated from the green and the orange one. The graph in the middle shows the same EncoderMap as in Figure 2. It is interesting to note that from the spatial arrangement of the clusters one would conclude that the EncoderMap axes are related to the first two principal components. The sequence red-orange-blue-green is resolved in both the y-axis of PCA and EncoderMap but the different colors are clearly better separated in the EncoderMap. The graph on the right shows a sketch-map projection using the same sigmoid parameters as for the EncoderMap. Also here, the different clusters are well separated. A measurable comparison of the quality of the different projections is not straight forward and not the focus of this work. We see the main strength of EncoderMap in its usability and its features beyond the generation of a low-d map. It also gives a function to efficiently map high-d points to this map and vice versa and does this with great computational efficiency which allows to use millions of data points without the requirement to select landmarks.

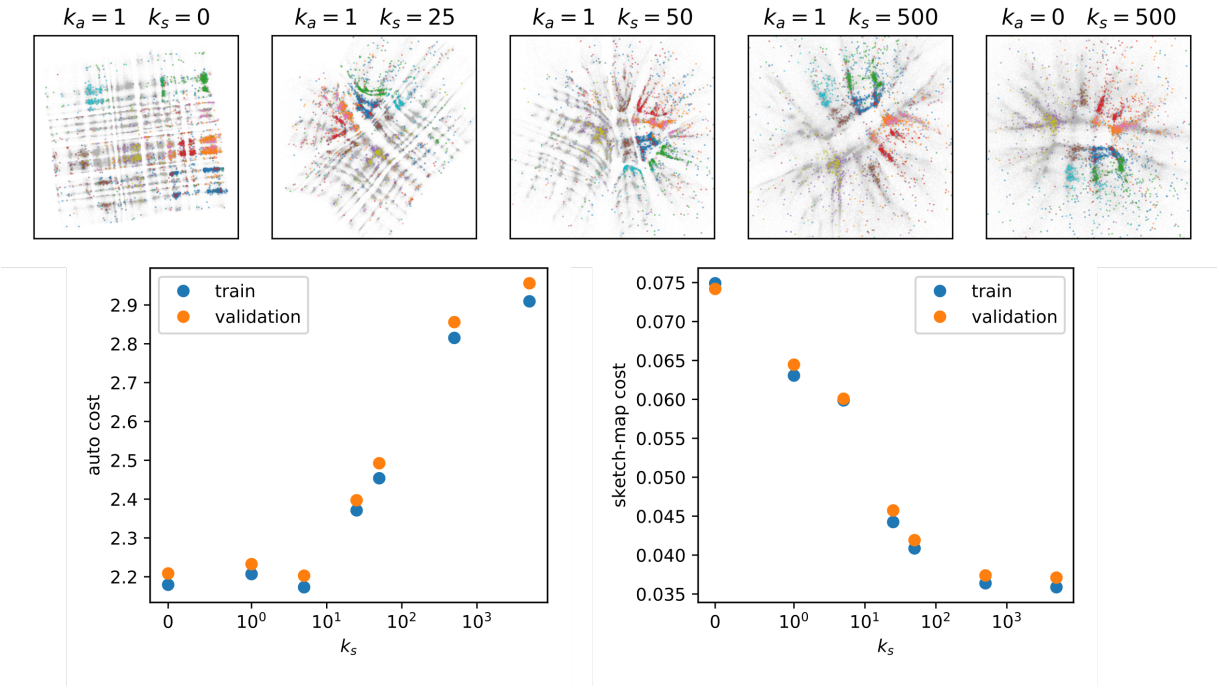
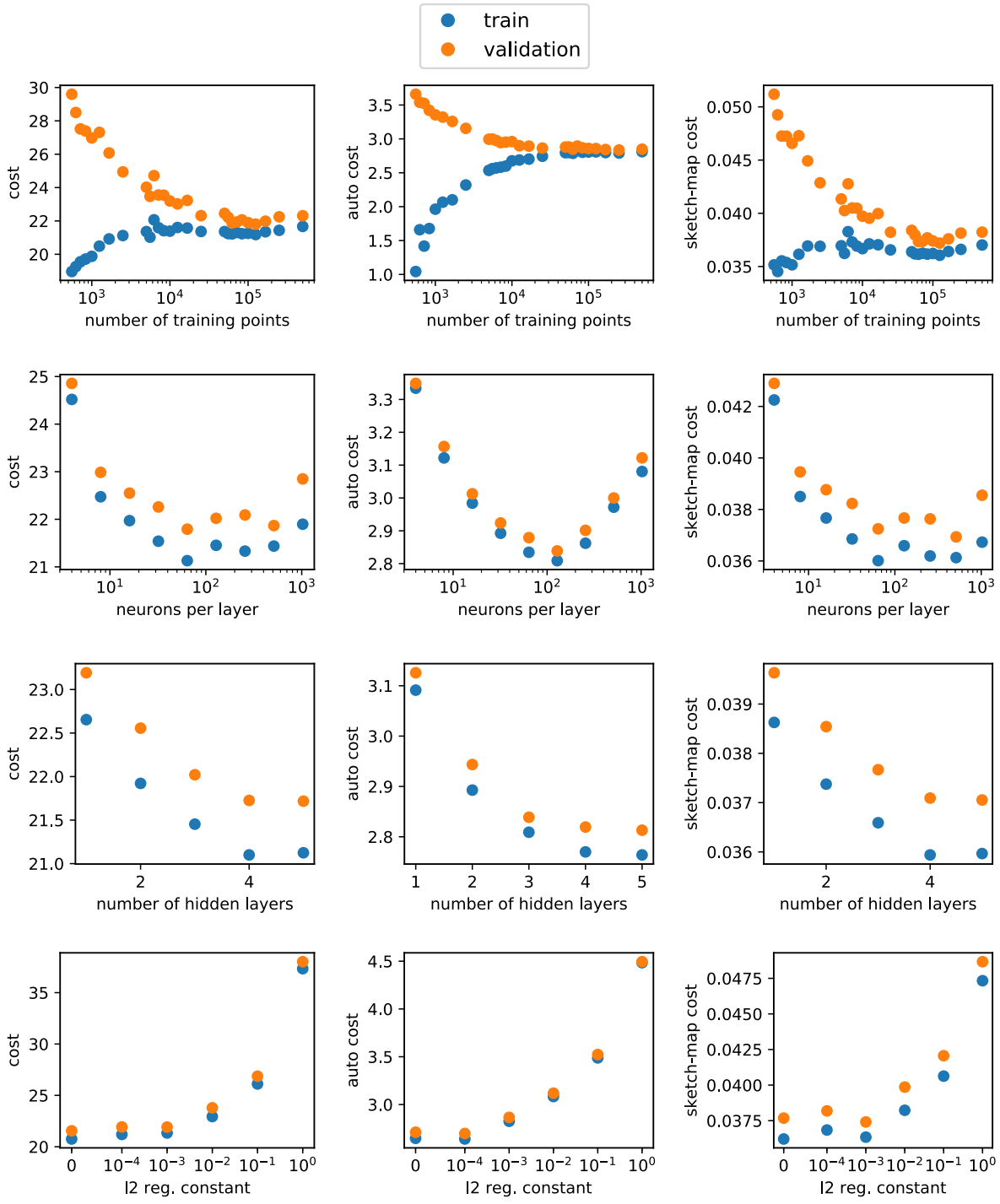


Figure S2: Comparison of different k_a and k_s settings. These two variables adjust the relative weight of the autoencoder and the sketch-map part of the cost function, respectively. The projections are from the asp-7 data set and colored in analogy to Figure 2 according to the rmsd-based clustering. Except for k_a and k_s all parameters were set as described in the parameter section of the main text. In the left graph in the top row k_s is set to zero. With this setting, EncoderMap is equivalent to a normal autoencoder where the projection is only optimized in such a way that the decoder part of the network is able to reconstruct the input as well as possible. In this map distances have little meaning. Clusters with similar conformations like for example the green and the dark blue cluster are not placed next to each other despite their conformational similarity (see Figure 2). Moreover, conformations within one cluster are not necessarily placed closely together. For example the dark blue cluster is split into different sub-clusters and it is not evident form the map that all these structures are similar. In the right graph k_a is set to zero. With this setting, the encoder part of the network is only optimized based on the sketch-map part of the cost function. Here, similar clusters are placed in proximity and transitions between these clusters become visible. It is, however, not possible to project back from low-d to high-d with this setting as the decoder is not trained in this case. The other graphs between these two extremes show maps for different ratios of k_a and k_s. The two panels in the lower row show the values for the two contributions to the cost function for different settings of k_s from 5-fold cross validation studies (see also Figure S3). With increasing k_s the auto cost increases since the optimization gets more focused on the sketch-map part of the cost function. Thus, the ability to reproduce the inputs is getting worse. At the same time the the values for the sketch-map part of the cost function go down. For $k_s > 500$ a plateau is reached. This is in agreement with the observation that the projection with $k_a = 1 \quad k_s = 500$ is already very similar to the projection only optimized based on the sketch-map part of the cost function.



caption on next page

Figure S3: Cross validation study for different EncoderMap settings for the asp-7 data set. To obtain the cross validation cost the simulation trajectory was split into 5 continuous parts of equal size. For each setting and each of the trajectory parts an EncoderMap was obtained where the respective part was used as validation set and the 4 other parts were used as training set. The values shown are the means of such 5-fold cross validation runs. The columns of plots contain cross validation results for: the overall cost, the autoencoder part of the cost function, and the sketch-map part of the cost function. Each row of plots contains results where one parameter was changed while all other parameters were kept constant at the values described in the parameter section of the main text. The first row contains results where the number of training points was reduced by using only every n^{th} training point. For small numbers of training points there is a large deviation between the costs evaluated based on the training and the validation set. This indicates strong over fitting. For lager numbers of training points this gap between training and validation cost gets smaller and over fitting appears to be not a big issue when the complete data is used. In the second row different numbers of neurons per hidden layer are evaluated. With too few neurons the network is to simple to perform its task. For more the 128 neurons per layer the auto cost is increasing again. This might be the case because other parameters like the regularization parameter were tuned to work with 128 neurons per layer. The third row of plots shows different numbers of hidden layers in comparison. Adding more layers appears to be beneficial up to a total of 4 layers. The bottom row compares different l2 regularization constants. Larger values of this constant force the network to have smaller weights which results in a smoother representation. Representations which are too smooth lack detail which results in an increased cost. We selected 10^{-3} as it is the largest value of the regularization constant where the cost is not considerably increased.

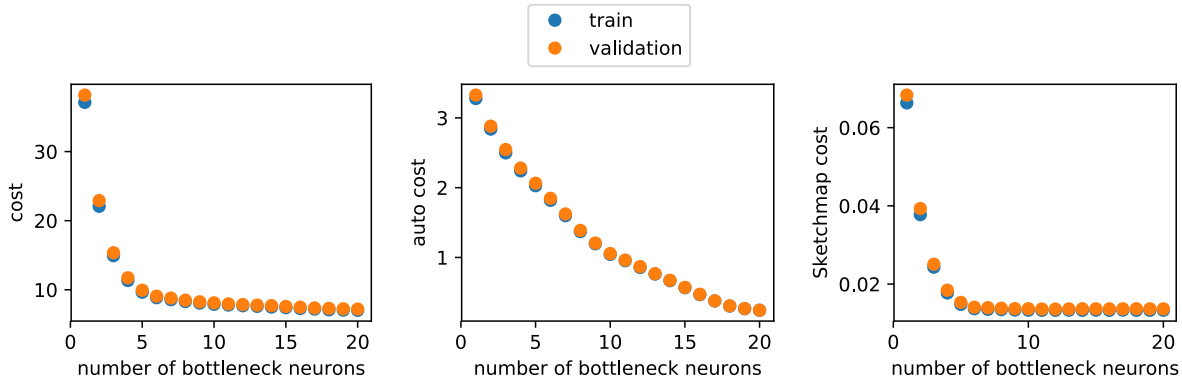


Figure S4: 5-fold cross validation cost (see also Figure S3) for different numbers of bottleneck neurons for EncoderMap based on the asp-7 data set. The number of bottleneck neurons is equivalent to the number of dimensions of the EncoderMap projection. The autoencoder part of the cost function measures the similarity between the high-d inputs and the reconstructed high-d outputs. This reconstruction becomes more accurate with a wider bottleneck. The distance based Skechmap cost also initially decreases with increasing number of bottleneck neurons, however, beyond $\sim$ 5 bottleneck neurons there is no further improvement. For clustering this arrangement according to the pairwise distances included in the sketch-map cost might be most important. This would lead to the conclusion that a 5 dimensional representation might be best suited for clustering analysis in this case. For visualization purposes, however, such higher dimensional projections are impractical.

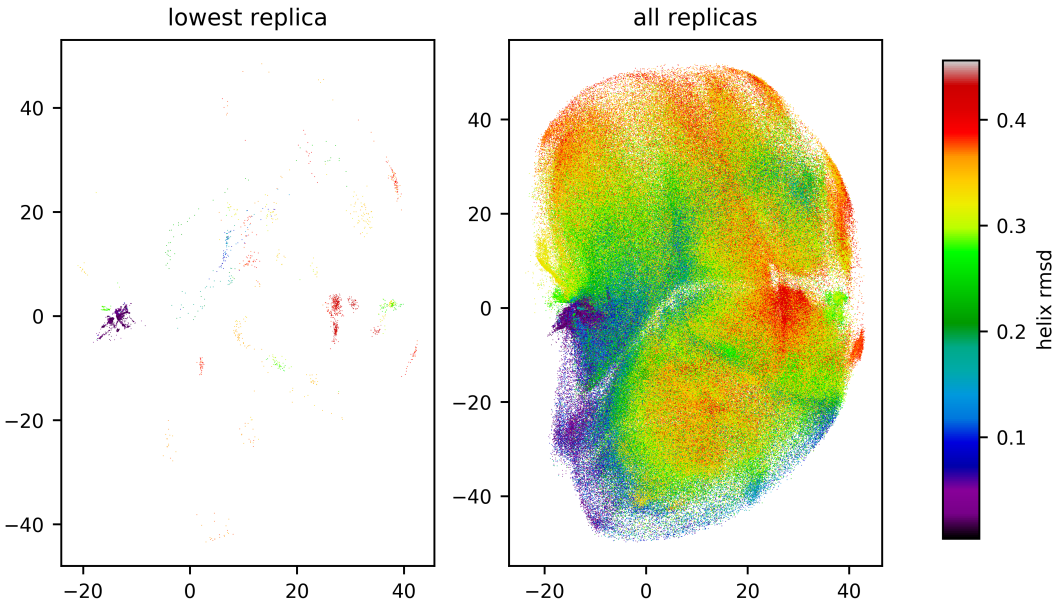


Figure S5: Comparison of the trp-cage encoder map for the lowest and for all temperature replicas. As expected, the lowest replica mainly contains conformations from the dense areas in the EncoderMap which correspond to the conformations with the lowest free energy.

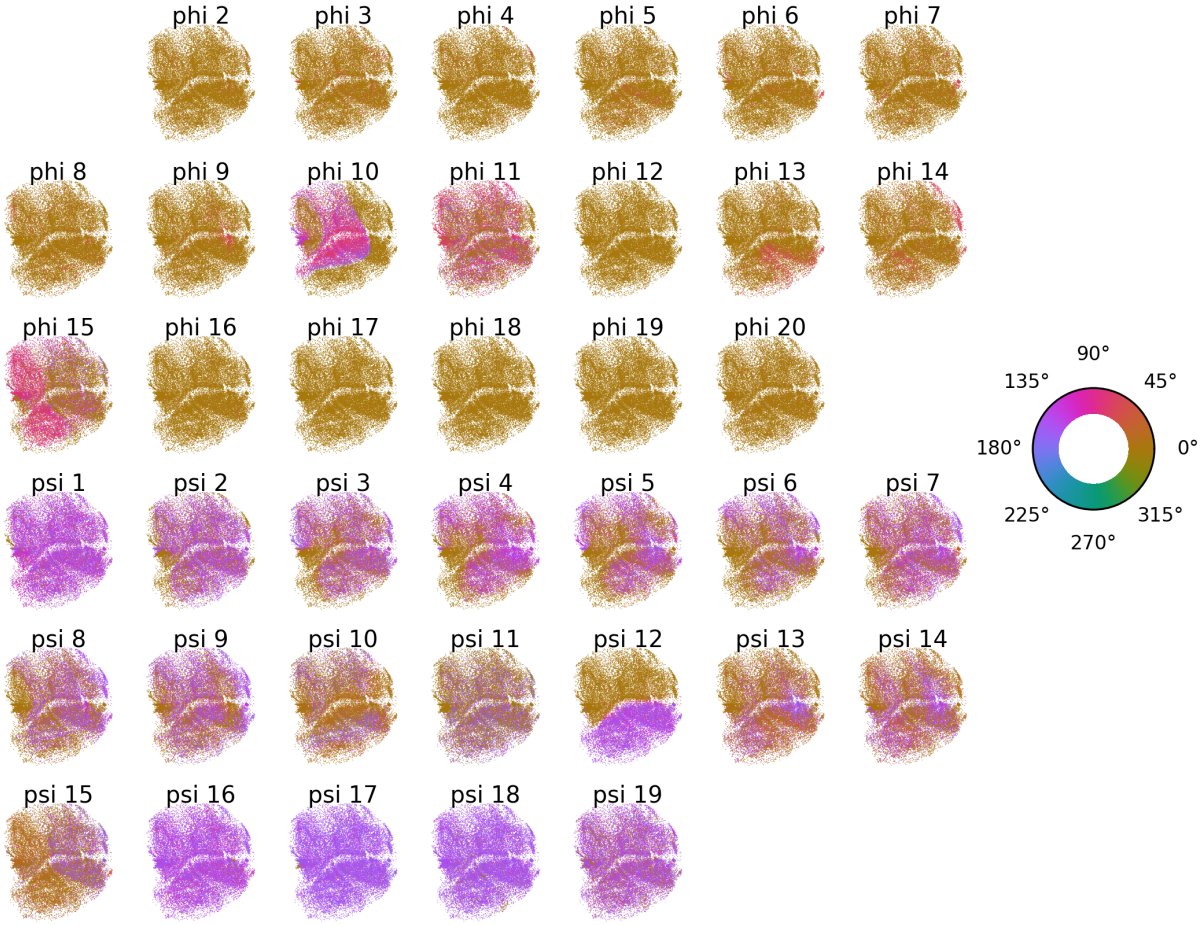


Figure S6: Trp-cage EncoderMap colored according to the different backbone dihedral angles. The angles are numbered starting at the N-terminus of the 20 residue protein. Note that the first and last amino acid do not have a phi and psi angle respectively. For some of the dihedrals, different values are clearly separated in the map. This is for example the case for psi 12 which appears to be the cause of the horizontal rift in the map. Psi 12 is located at a proline residue in the center of the protein. The clear separation in the map might indicate that rotation around this dihedral is very decisive for the conformation of the protein.

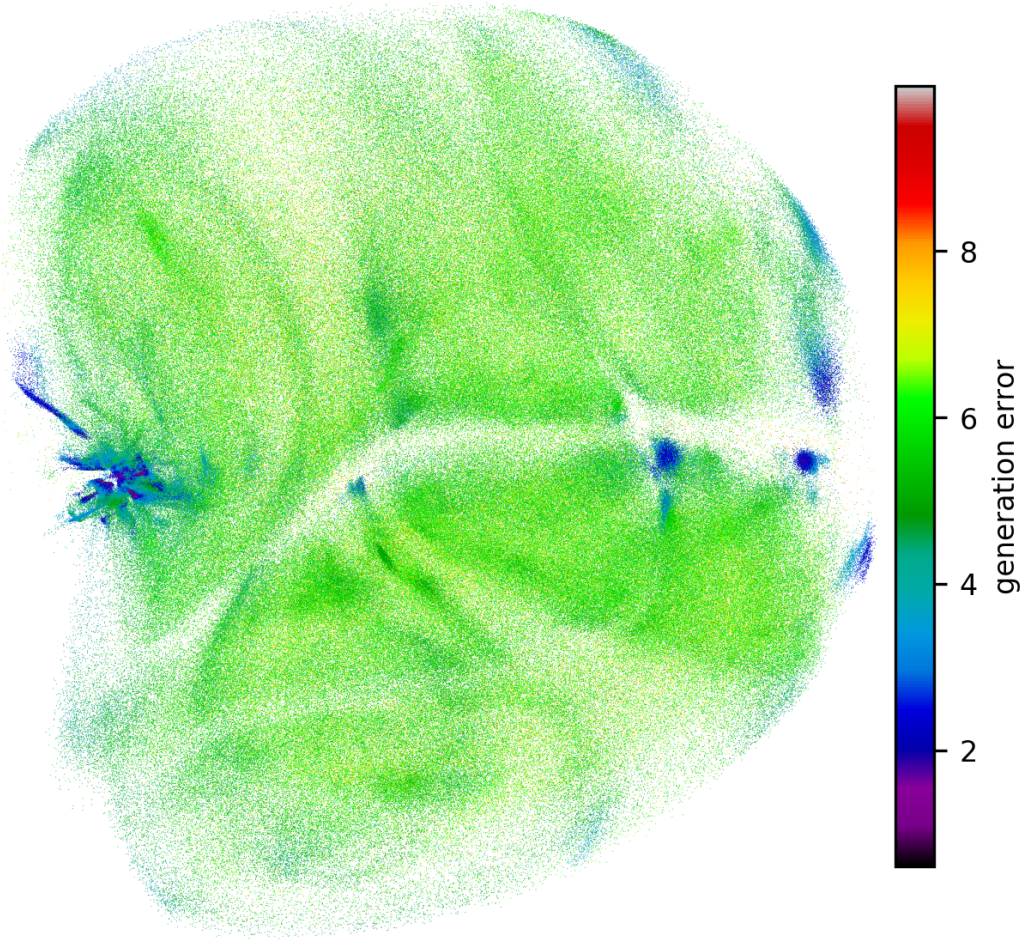


Figure S7: Trp-cage EncoderMap colored according to the deviation between conformations in the data set and conformations generated by the decoder part of the Network. The generation error is calculated as smallest euclidean distance in the the space of dihedrals taking periodicity into account. Deviations are especially small in densely populated areas with clearly defined structures. Outside of these areas where unfolded structures can be found, a broader variety of training structures can be expected. The generation error is presumably larger in these areas because the neural network has to interpolate between a broad variety of disordered conformations.

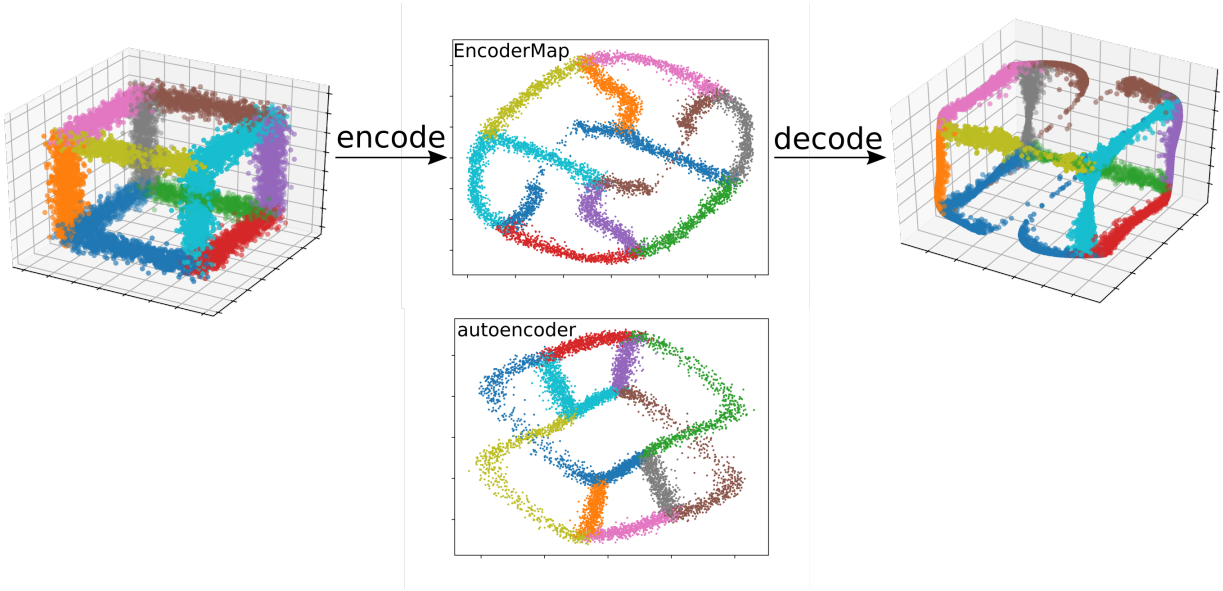


Figure S8: EncoderMap of a 3d toy example. The top left graph shows the data points of this toy data set which are distributed on the edges of a cube with gaussian noise. The top middle graph shows a projection of this data to a 2d plane created with EncoderMap. The top right graph shows the reconstructed 3d points based on the 2d map. The bottom graph shows a projection where the sketch-map part of the cost function was turned off. This resembles a "pure" autoencoder. Distances like the lengths of the cube edges are much better preserved in the EncoderMap projection which uses the distance based sketch-map cost function. Consequently also point densities are better preserved in the EncoderMap projection. The used sketch-map cost function parameters are $\sigma = 0.2$, $a = 3$, $b = 6$ for the sigmoid applied to the high-d distances, and $\sigma = 1$, $a = 2$, $b = 6$ for the low-d distances. The edge length of the cube is 1.

Movie S1: Generated conformations for asp-7. A black path is shown in the EncoderMap on the left. 450 points along this path were fed into the decoder part of the autoencoder. The sequence of generated conformations is shown on the right. The point from which each structure is generated is highlighted with the red ball on the left.

Movie S2: 3-dimensional EncoderMap for asp-7. The constellations of clusters in this 3-d map are very similar compared to the 2d map in Figure 2. From a subjective perspective this 3-d map does not appear to be advantageous compared to the 2d map. A more objective evaluation based on the values of the cost function for projections with different dimensionality is shown in Figure S4.

Movie S3: Trp-cage conformations generated along path A. 450 points along this path were fed into the decoder part of the autoencoder. The sequence of generated conformations is shown on the right. The point from which each structure is generated is highlighted with the red ball on the left. The generated conformations nicely illustrate that along this path the helix is generated last after the rest of the protein with the hydrophobic core is already in place.

Movie S4: Trp-cage conformations generated along path B. Generated conformations along this path illustrate how the helical part can be folded first before the rest of the protein with the hydrophobic region is folded.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING
Cite This: J. Chem. Inf. Model. XXXX, XXX, XXX−XXX
Article
pubs.acs.org/jcim

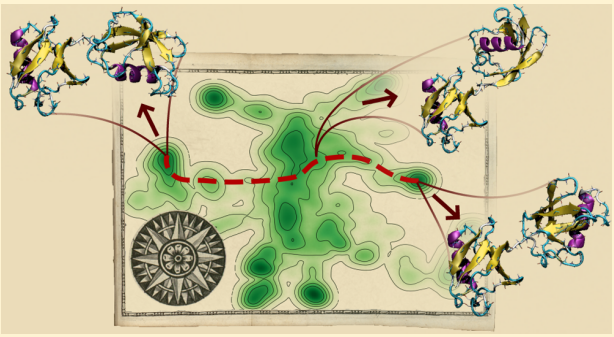
EncoderMap(II): Visualizing Important Molecular Motions with Improved Generation of Protein Conformations

Tobias Lemke,[†] Andrej Berg,[†] Alok Jain,[†,‡] and Christine Peter*[,†]

[†]Theoretical Chemistry, University of Konstanz, 78547 Konstanz, Baden-Württemberg, Germany
[‡]Department of Biotechnology, National Institute of Pharmaceutical Education and Research Ahmedabad, Gandhinagar, Gujarat 382355, India

Supporting Information

ABSTRACT: Dimensionality reduction can be used to project high-dimensional molecular data into a simplified, low-dimensional map. One feature of our recently introduced dimensionality reduction technique EncoderMap, which relies on the combination of an autoencoder with multidimensional scaling, is its ability to do the reverse. It is able to generate conformations for any selected points in the low-dimensional map. This transfers the simplified, low-dimensional map back into the high-dimensional conformational space. Although the output is again high-dimensional, certain aspects of the simplification are preserved. The generated conformations only mirror the most dominant conformational differences that determine the positions of conformational states in the low-dimensional map. This allows depicting such differences and—in consequence—visualizing molecular motions and gives a unique perspective on high-dimensional conformational data. In our previous work, protein conformations described in backbone dihedral angle space were used as the input for EncoderMap, and conformations were also generated in this space. For large proteins, however, the generation of conformations is inaccurate with this approach due to the local character of backbone dihedral angles. Here, we present an improved variant of EncoderMap which is able to generate large protein conformations that are accurate in short-range and long-range orders. This is achieved by differentiable reconstruction of Cartesian coordinates from the generated dihedrals, which allows adding a contribution to the cost function that monitors the accuracy of all pairwise distances between the C_α-atoms of the generated conformations. The improved capabilities to generate conformations of large, even multidomain, proteins are demonstrated for two examples: diubiquitin and a part of the Ssa1 Hsp70 yeast chaperone. We show that the improved variant of EncoderMap can nicely visualize motions of protein domains relative to each other but is also able to highlight important conformational changes within the individual domains.



INTRODUCTION

One of the major challenges in molecular simulation is to extract relevant information from the large amounts of high-dimensional data produced. Each frame in a simulation trajectory represents a point in a phase space with easily thousands or millions of dimensions. Each atom contributes in principle three dimensions to the configurational phase space. However, not all of them are equally important. A chain-like molecule like a protein can for example be described in a space of backbone dihedral angles. Information about the side chains and solvent is lost in this representation, but it still describes the conformation of the backbone well. Although the dimensionality of the dihedral space is much lower compared to the full phase space, it might still comprise tens or hundreds of dimensions in the case of a protein. The dimensionality needs to be further reduced to identify important conformational states and to characterize molecular motions like conformational transitions and folding.

Several dimensionality reduction techniques such as principle component analysis (PCA),[1,2] diffusion map,[3] time-lagged independent component analysis,[4] neural network autoencoders,[5−9] or variants of multidimensional scaling[10] such as sketch-map[11] are established in the simulation community.[12] Recently, we introduced a dimensionality reduction technique called EncoderMap,[13] which unites the advantages of autoencoders and multidimensional scaling. One of the major advantages of this method is that it is not only possible to project molecular conformations to a meaningful low-dimensional map but also possible to generate molecule conformations for any points in the low-dimensional map. This generation of conformations, for example, along paths selected in the low-dimensional map, is a unique tool to visualize the most important conformational motions. Unlike conformations selected from the original trajectory, which typically vary in

Received: August 13, 2019
Published: October 24, 2019

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

multiple degrees of freedom, the generated structures only reflect the major conformational changes identified during the dimensionality reduction process. These major conformational changes can thus be visualized without the distracting noisiness of a wiggling molecule.

EncoderMap combines a neural network autoencoder with the pairwise distance-based cost function of the multidimensional scaling variant sketch-map[11] as illustrated in Figure 1. In

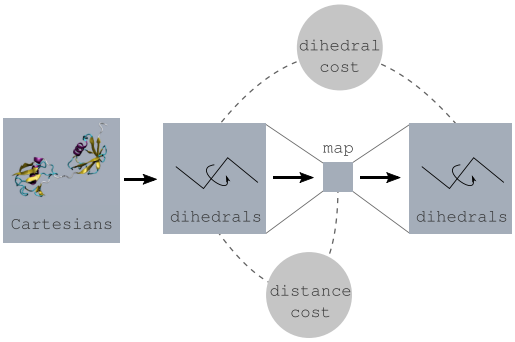


Figure 1. Schematic of the data flow and contributions to the cost function of EncoderMap based on dihedrals. The low-dimensional map is obtained from the bottleneck in a neural network autoencoder. The network is trained to reproduce the dihedral inputs (dihedral cost) and to match pairwise distances between data points in the input dihedral space and pairwise distances between the corresponding points in the map (distance cost).

multidimensional scaling, data points in the low-dimensional representation are arranged according to some distance metric, which we term "distance cost". This distance cost puts distances between pairs of data points in a high-dimensional space in comparison to distances between the corresponding points in the low-dimensional representation. This ensures that points that are close together in the high-dimensional space are also close together in the low-dimensional map and that points that are far apart in the high-dimensional space are also far apart in the low-dimensional map. The autoencoder part of EncoderMap enables an efficient projection to a low-dimensional map, as batchwise training circumvents the quadratic scaling of other multidimensional scaling approaches and allows generating conformations for given points in the map. More information about EncoderMap is available in the paper[13] or the introductory YouTube video.[14]

In this previously published EncoderMap setup, the Ramachandran dihedral angles of a protein were used as inputs for the autoencoder. Also, the conformations generated for given points in the low-dimensional map were obtained in this same dihedral space. In the case of proteins, one can then reconstruct backbone conformations in Cartesian space rather straightforwardly by successive rotation around the different dihedral axes of a starting conformation. All bond lengths and bond angles are assumed to be constant in this approach. For a small protein like trp-cage,[15] the generation of conformations with this approach yields accurate results.[13] For large molecules, however, this approach is problematic. Backbone dihedrals are local descriptors.[16] They accurately describe local motifs and secondary structure elements. Small deviations in the prediction of each dihedral angle, however, quickly add up along the chain and result in an inaccurate long-range order. Figure 2 compares conformations reconstructed from the dihedral output of EncoderMap with the original conformations from the simulation. Secondary structure elements are well recovered, but in the case of the larger diubiquitin molecule, which consists of two folded domains connected via a flexible linker, the spacial arrangement of these elements is inaccurate. This becomes most evident in the misaligned and partially overlapping β-strands. In the case of trp-cage, this is not observed simply because the backbone chain is comparatively short.

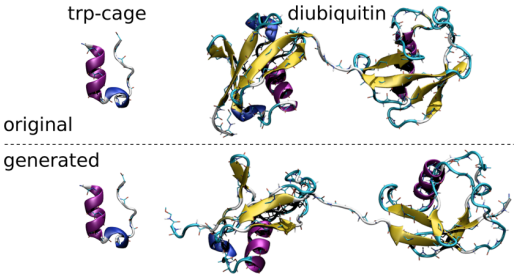


Figure 2. Comparison of an original trp-cage and a diubiquitin conformation (top) and the corresponding neural network generated conformations (bottom) where the network was trained only based on the backbone dihedrals. The correct secondary structure elements are present in the generated conformations, but the long-range order and spacial arrangement of these elements are inaccurate in the case of the large diubiquitin molecule.

In the following, we will address the question how to improve the generation of large molecule conformations and consequently EncoderMap's ability to visualize important conformational differences in the map.

The generation of molecule conformations is relevant in many different contexts. In molecular simulation, the generation of realistic structures at full atomistic resolution is relevant in various scale bridging approaches, for example, when a high-resolution structure needs to be reconstructed from a lower-resolution (coarse grained) model or when an advanced sampling algorithm starts new simulations deliberately in regions of phase space slightly extrapolated from where the simulation has already been.[17,18] Similar tasks are also relevant in the structure prediction community where the goal is, for example, to predict a protein fold from its amino acid sequence.[19,20] Often structure prediction methods, however, do not generate conformations from scratch but start from template conformations of similar proteins[21−23] and assemble new conformations from protein fragments.[24] Other methods generate conformations in dihedral space[25,26] but have problems with long-range order in a similar manner as EncoderMap. To make up for the local character of dihedrals, some methods rely on the prediction of contacts[27,28] or distances between amino acids.[29−31] Also AlphaFold,[32] which caught much attention with its exceptional performance in the CASP13 structure prediction competition, relies on the prediction of distances between amino acids. A drawback of pairwise distances between atoms or residues as model output to generate conformations is their ambiguity. The created set of distances is not necessarily geometrically conclusive. Instead, an additional optimization problem has to be solved to find a conformation that fits best to the given distances. In contrast to the pairwise distances, the backbone dihedrals are all geometrically independent from each other, and each set of dihedrals represents one unambiguous backbone conformation (ignoring bond length and bond angle degrees of freedom).

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

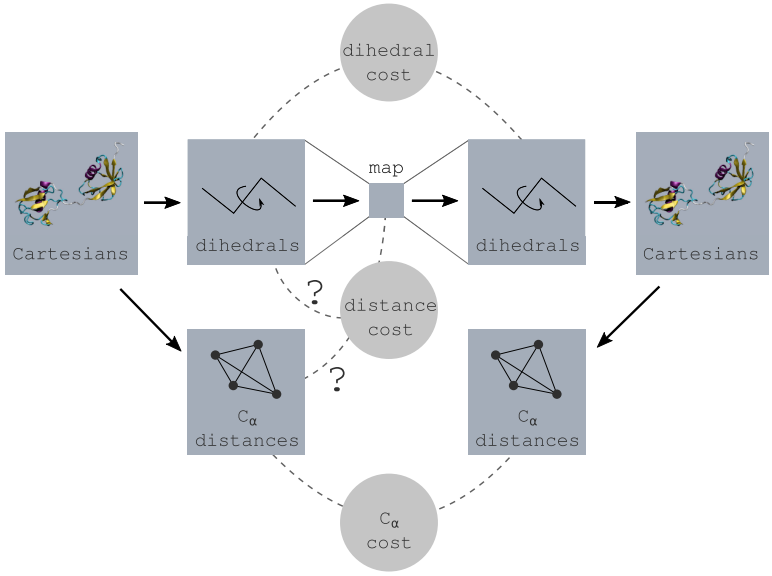


Figure 3. Schematic of the data flow and contributions to the cost function. The neural network autoencoder, which uses dihedrals as inputs and outputs, is supplemented with a reconstruction of Cartesian coordinates. This allows for an additional contribution to the cost function where pairwise distances between all C_α atoms are compared between the original and the generated conformations. The presence of C_α distances also allows choosing the high-dimensional reference space of the distance cost. Points in the map can be either arranged according to their distances in dihedral space or their distances in the C_α pairwise distance space.

Starting from these observations, to improve the quality of the structures generated by EncoderMap, we were looking for a way to generate protein conformations which are accurate in short- and long-range orders in an unambiguous way which does not require any subsequent optimization procedures or simulation steps. In the following section, we show how this can be achieved by reconstruction of Cartesian coordinates during the neural network training process and including this information as an additional contribution in the cost function. We also compare the influence of different deviation metrics in the cost function: that is, using mean square versus mean absolute deviation. Finally, we demonstrate the capabilities of the improved generation of conformations with two example systems. The first example is the M1-linked diubiquitin (two ubiquitin domains covalently linked via a peptide bond between their respective C- and N-terminal amino acids, see also Figure 2).[33] Diubiquitin is an ideal test system because, with 152 amino acids and its two-domain character, it is large enough that inaccuracies in the long-range order quickly show up. At the same time, with its two relatively rigid ubiquitin domains, its conformational landscape has a decent complexity where it is still reasonable to make a projection to a two-dimensional (2D) map. It is also an interesting test system because conformational changes can occur on very different scales. Conformations can vary on a global scale with different positions of the two domains relative to each, and conformations can vary on a local scale with differences inside a single domain. Capturing differences on such different scales poses an additional challenge for a dimensionality reduction technique. The second example is a part of the Ssa1 Hsp70 yeast chaperone.[34−36] With 210 residues, it is even larger compared to diubiquitin, and while it also consists of two distinct domains, a β-barrel substrate binding domain and a lid domain, which is constituted by several α-helices, the lid exhibits several hinge regions, which allow for rearrangements. Thus, it represents a further challenge to test EncoderMap's abilities to generate large protein conformations. Furthermore, this example consists of two separate simulations sampling different not overlapping regions of the conformational space. This allows us to analyze how EncoderMap deals with sparse data and how well it is able to generate reasonable conformations for void areas between sampled regions.

IMPROVED GENERATION OF CONFORMATIONS

Differentiable Reconstruction of Cartesian Coordinates. The fundamental idea of this approach is to still generate conformations in dihedral space like in the previous version of EncoderMap, but to reconstruct the Cartesian coordinates from these generated dihedrals during the training procedure, which allows adding a contribution to the cost function measuring the accuracy of all C_α pairwise distances in the generated conformations. This extended scheme is illustrated in Figure 3.

The extended scheme has three contributions to the cost function: the dihedral cost is the mean absolute deviation between the dihedrals of the input conformations and the generated conformations and ensures accurate short-range order. The C_α cost is the mean absolute deviation between all C_α-atom pairwise distances of the input conformations and the generated conformations and additionally ensures accurate long-range order. The distance cost compares distances between data points in the high-dimensional space with the corresponding distances of these points in the map. In our previous implementation of EncoderMap, the dihedral space was used as high-dimensional reference space to calculate the distance cost. As the C_α atom pairwise distances are calculated anyway for the C_α cost, it is now possible to choose between the dihedral space and C_α distance space as high-dimensional reference space for the distance cost. The three contributions to the cost function are described in more detail in the Details section at the end of the article.

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

In order to use the C_α cost to optimize the weights of the neural network with gradient descent, the reconstruction of the Cartesian coordinates from the generated dihedrals needs to be differentiable. We, therefore, implemented the reconstruction of the Cartesian coordinates with differentiable operations from the TensorFlow[37] library. To reconstruct the Cartesians, we start from a conformation where all dihedrals are zero and all bond lengths and bond angles are equal to the mean value observed in the simulation data. Then, we iteratively set all of the dihedral angles to the values obtained as the output of the neural network autoencoder. Therefore, we shift the molecule in such a way that the rotational axis of the given dihedral lies in the origin of the Cartesian coordinate system. Then, we update the Cartesian coordinates of all atoms on one side of the dihedral by multiplication with the rotation matrix for the given rotational axis and dihedral value. The Details section at the end of the article contains some pseudo code for this procedure. The full code is available in the EncoderMap repository (https://github.com/AG-Peter/EncoderMap). A similar fully differentiable approach to generate molecule conformations has recently been proposed by AlQuraishi[38] in the context of a structure prediction algorithm.

The three contributions to the cost function shown in Figure 3 are combined as a weighted sum

$$C = k_{\text{dih.}} C_{\text{dihedral}} + k_{C_\alpha} C_{C_\alpha} + k_{\text{dist.}} C_{\text{distance}} \quad (1)$$

The prefactors $k_{\text{dih.}}$, k_{C_α}, and $k_{\text{dist.}}$ can be used to balance the different contributions of the cost function. To make the balancing easier in this multiobjective optimization[39] problem, we normalize the contributions concerning the generation of conformations with a dummy model as baseline. The dummy model always returns the conformation where all dihedrals are set to their mean value. The mean dihedrals are evaluated using all available trajectory frames. Because of this normalization, a C_α cost or dihedral cost of 1 means equal performance to the dummy model, a value between 0 and 1 signifies better performance compared to the dummy model, and a value larger then 1 signifies worse performance compared to the dummy model.

We evaluate the influence of the newly added C_α cost by applying the extended scheme described above to 60 000 diubiquitin conformations obtained from atomistic molecular dynamics simulations. Simulation details can be found in the Details section. Figure 4 shows learning curves for the two cost function contributions concerning the generation of conformations. We performed 10 neural network training runs for each of three different settings of k_{C_α}. The light colored lines represent these single runs while the saturated lines represent the mean of all 10 runs for one setting. In the first setting, k_{C_α} was set to 0 and $k_{\text{dih.}}$ to 1. With this setting, the C_α cost has no influence in the cost function. During the training process, the dihedral cost goes down quickly but the C_α cost stays on a comparatively high level. This mirrors the fact that the dihedral cost is mainly sensitive to the short-range order but not so much to the global structure of the protein. In the second setting, both k_{C_α} and $k_{\text{dih.}}$ were set to 1. With this setting, the C_α cost is drastically reduced compared to the previous setting. This means that pairwise distances between C_α atoms are much more accurately reproduced in the generated conformations. The dihedral cost, however, is elevated compared to the previous setting, which means that dihedrals are less

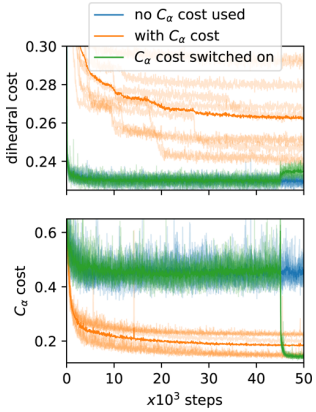


Figure 4. Influence of the C_α cost on the learning curves for diubiquitin. The C_α cost is either not used at all (blue) or used from the beginning (orange) or switched on after 45 000 steps. The dihedral cost is shown in the top graph. The C_α cost (evaluated regardless of its usage) is shown in the bottom graph.

accurate in the generated conformations. There is also a larger spread between the different runs with this setting, showing that the training process is less reproducible. When the generated conformations are only assessed based on their dihedrals, interpenetrating parts of the molecule and wrong long-range spatial arrangement of the protein do not cause barriers in the cost function. The correct secondary structure can easily be found without the need to cross such barriers. When the C_α cost is activated from the beginning, it is much more difficult to find the correct short-range order because the dihedrals can no longer be adjusted indifferent of the long-range spacial arrangement. In the third setting, we, therefore, first keep k_{C_α} set so 0, and then after 45 000 steps, we ramp k_{C_α} up linearly to 1 in 1000 steps. For the first 45 000 steps these runs are equivalent to the first setting and the dihedral cost is quickly reduced. As soon as the C_α cost is activated in the cost function, its values decrease drastically. With regard to the C_α cost, the end results are comparable to the best runs with the setting where the C_α cost was activated from the beginning. The training is, however, much more reliable when the C_α cost is turned on during the training process, as all 10 runs performed equally well with this setting. The dihedral cost is only slightly elevated as soon as the C_α cost is switched on, which demonstrates that the short-range order is mostly preserved when the conformations are in the end also optimized for the long-range order.

These results show that the accuracy of the C_α pairwise distances in the generated conformations can be substantially improved by adding the C_α cost to the cost function. The best results are achieved when the C_α cost is not active from the beginning but turned on during the training process. Besides leading to the best results, not using the C_α cost from the beginning also has computational advantages. The reconstruction of the Cartesian coordinates from the generated dihedrals causes significant computational overhead. Not using the C_α cost for most of the training process helps to avoid this overhead.

For the results above, the dihedral and C_α cost were calculated as the mean absolute deviation between the simulated and the generated conformations. In the following, we show why it is very important in this case to use the mean

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

absolute deviation and not the very common mean square deviation.

Choice of the Cost Metric: Mean Square Versus Mean Absolute Deviation. We find that it is very important how the deviation in the dihedrals and the C_α pairwise distances between the simulated and the generated dihedrals is calculated. We compare two popular metrics for deviation, that is, mean square deviation and mean absolute deviation. A comparison of these cost function variants requires a measure of quality of the generated conformations that is independent of the cost functions themselves. Here, we use the number of clashes, that is, atomic overlaps, in the generated diubiquitin conformations. We consider any distance between two atom centers shorter than 100 pm as a clash. The average number of clashes in the generated conformations after training with different cost function variants is shown in Figure 5. When no

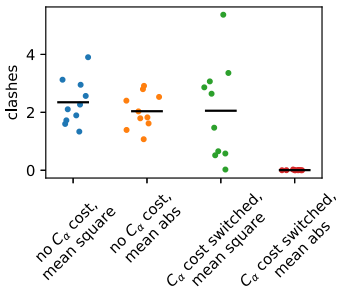


Figure 5. Average number of clashes in generated diubiquitin conformations after 50 000 steps of training with different cost function variants. Each dot represents the average result at the end of a single training run. The black line represents the mean of the 10 runs for each cost function variant.

C_α cost is used, generated conformations contain on average around two clashes. Using mean absolute instead of mean square deviation only results in slightly improved results in this case. However, when the C_α cost is switched on after 45 000 steps as described in the previous section, there is a massive difference between using mean absolute versus mean square deviation. When mean square deviation is used, there are on average still around 2 clashes in each conformation. When mean absolute deviation is used instead, hardly any clashes occur. How can such a drastic difference be explained?

Figure 6 illustrates the behavior of mean absolute cost and mean square cost for two one-dimensional example distributions for which a model should predict an x-value. In these examples, we further assume that the model is unable to distinguish all these points and can only make a prediction of a single x-value equally for all of the points. In the first example, 100 points are distributed according to a Gaussian distribution. The two lines show the respective cost as a function of the x-value predicted by the model. A model trained to convergence with any of these cost functions would predict an x-value corresponding to the minimum of the respective cost function. In the case of this unimodal distribution, the minimum of both cost functions is in the center of the distribution. The second example is a bimodal distribution where 40 points are distributed in a Gaussian on the left and 60 points are distributed in a Gaussian on the right. Let us again assume that the model is unable to distinguish all these points and can only make one prediction for all of the points. The minimum of the mean square cost is the mean of the data, which is in between

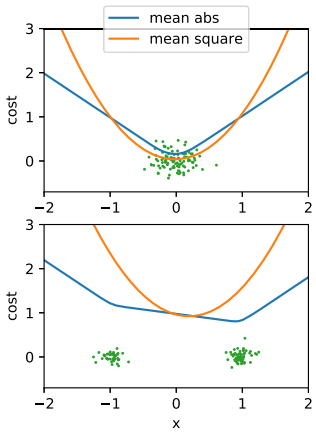


Figure 6. Mean abs and mean square cost for a unimodal and a bimodal distribution. The green points represent the example distributions. Only the x-component of the points is considered but they are scattered in y-direction for better visibility. The blue and orange line show the mean abs and mean square cost for a model returning a given x-value.

the two Gaussian distributions. A model that is trained with mean square cost would therefore predict an x-value that is in between the two Gaussians in an area where no data points have ever been observed. The minimum of the mean absolute cost is the median of the data. As both Gaussians of this bimodal distribution are not equally populated, the median lies in the more populated Gaussian. A model that is trained with mean absolute cost would therefore predict an x-value inside the more populated Gaussian.

Such a scenario, where the model cannot distinguish between different points and has to make one prediction for all of them, also frequently applies for autoencoders, especially, if the bottleneck is very narrow. Here, we force the network to project a very high-dimensional conformational space into a two-dimensional plane. It is clearly impossible to resolve all conformational variation in this two-dimensional map, and the choice of the cost metric determines how the network deals with this situation. Let us consider a flexible tail of a protein; the structural fluctuations of which are unimportant to describe major conformational changes of the protein and that are therefore not resolved in the two-dimensional map. Conformations with different orientations of this tail might be projected to the same area in the map and can therefore not be distinguished. Let us further assume that this tail can be found in two predominant orientations. A model trained with mean square cost would generate conformations where the orientation is in between the two predominant orientations: an orientation that might be nonphysical and lead to clashes. A model trained with mean absolute cost would generate conformations where the orientation of the tail corresponds to the more dominant orientation. It is therefore more likely a physically reasonable conformation without clashes. Note that this comes at the price of neglecting the second, less dominant conformation.

This behavior, where the network generates conformations more like the most likely conformations rather than generating conformations that have never occurred but are in between those that have occurred, is more useful for the visualization of relevant conformational states and transitions between them. Moreover, the generation of realistic conformations and the

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

avoidance of steric clashes is of immense importance if those structures are to be used to initiate new simulations. We demonstrate the capabilities of EncoderMap's improved conformation generation at the example of two large proteins in the next section.

VISUALIZATION OF IMPORTANT MOTIONS

In the previous sections, we introduced the reconstruction of Cartesian coordinates during the training process to add a cost contribution that improves the long-range order of generated conformations and showed why it is important to use mean absolute deviation instead of mean square deviation to calculate the cost for dihedrals and C_α pairwise distances. Here, we demonstrate the results of these improvements at the example of diubiquitin. We have created a few videos to be able to adequately show these results. A video icon in the top right corner of the following figures signifies that this content is available as video in the Supporting Information

Diubiquitin. Figure 7 and the accompanying video show a two-dimensional histogram of all 60 000 simulated diubiquitin

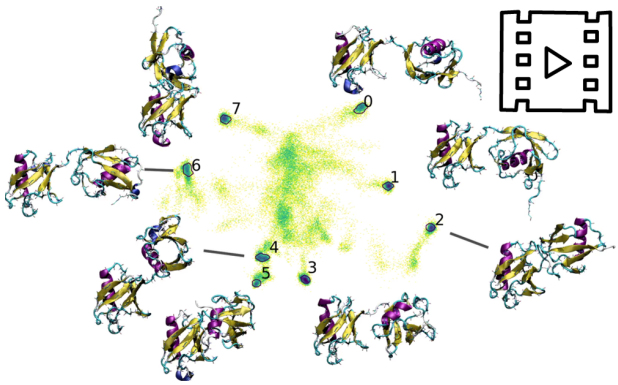


Figure 7. This figure and video show a histogram of all simulated conformations projected to the two-dimensional map. The color ranges from yellow to blue for low to high density. Empty bins are shown in white. Hand-selected areas are highlighted and random conformations that where projected to these areas are shown next to them. All conformations are aligned with their N-domains oriented to the left.

conformations projected to a two-dimensional map. The results are shown for one of the training runs also shown in Figure 4 where the C_α cost was turned on after 45 000 steps of training and mean absolute deviation was used as deviation metric. The map contains several spots with increased density. These spots were selected by hand with a lasso tool (available in the EncoderMap library), and original conformations from the simulation data that where projected to these spots are shown in the video. All conformations are aligned with their N-terminal ubiquitin domains (oriented to the left). This analysis shows that conformations are nicely grouped on this map according to the relative spacial arrangement of the two domains and demonstrates the meaningfulness of the map. In Figure 3, we introduced the possibility to choose either the dihedral space or the C_α distance space as basis for the distance cost which takes care that the two-dimensional map preserves relative distances between different conformations. For the map in Figure 7 and the following figures, we chose to use the C_α distance space as reference space for the distance cost. This space is very sensitive to global conformational differences of the two-domain protein, and this aspect is therefore nicely captured in the map. In the (local) dihedral space instead, conformations with different arrangements of the two domains would only differ in few dihedrals in the linking part between the two domains. If the distances between structures were calculated in the dihedral space instead, conformations that differ only in the relative arrangement of the two domains would therefore not be separated as clearly.

Such an analysis where the original conformations are shown that are projected to certain areas in a low-dimensional map is possible with any dimensionality reduction technique. A feature of EncoderMap is to additionally transfer this simplified low-dimensional map back into the high-dimensional space. This high-dimensional yet simplified representation contains only the major conformational variations that determine their location in the map. We can explore this representation by selecting paths in the low-dimensional map. Paths can be selected with a tool included in the EncoderMap library. Equally spaced (two-dimensional) points along such paths are then fed into the decoder part of the autoencoder to generate the (full Cartesian backbone) conformations corresponding to these points. The generated conformations from different paths indicated in the map are shown in the video accompanying Figure 8. The low C_α cost and small number

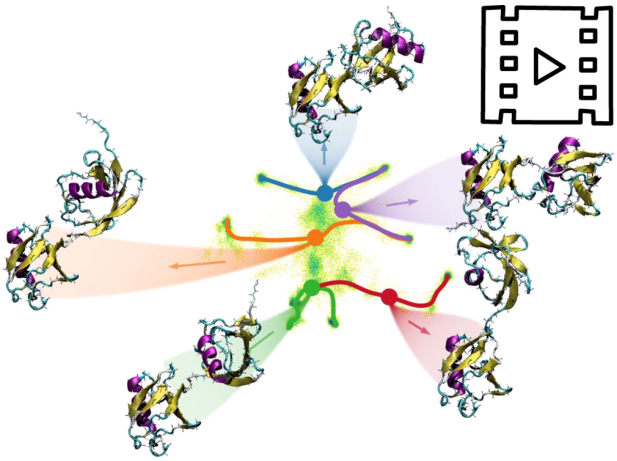


Figure 8. This figure and video show the same map as in Figure 7. Next to the map, generated conformations are shown for points along different paths in the map. The dots represent the points for which the generated conformations are shown at that time.

of steric clashes that had been found in the analysis described in the previous sections had already indicated that these generated conformations should be much more realistic and accurate in their long-range order compared to conformations generated only based on dihedrals. The generated conformations shown in Figure 8 indeed look very realistic. The β-strands of the ubiquitin domains are well aligned, and no clashes are apparent. This in strong contrast to the generated conformation solely based on dihedrals shown in Figure 2. With the naked eye, it is hard to distinguish individual original conformations (Figure 7) and generated conformations (Figure 8). In contrast to the bundles of original conformations that were projected to certain areas in the map (Figure 7), the generated conformations along a path, however, do not show distracting structural fluctuations of the molecule but only the dominant conformational changes. This

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

nicely illustrates the motions it takes to transform one of the dominant diubiquitin conformations into another. Mainly, motions of the two ubiquitin units relative to each other are visible in Figure 8, but the single ubiquitin units are also not completely rigid entities. The model has the freedom to also represent relevant (backbone) conformational changes within a single ubiquitin unit. Figure 9 and the accompanying video

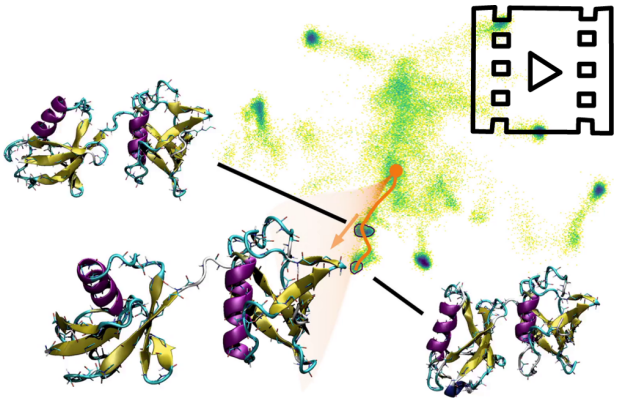


Figure 9. This figure and video show a comparison of original conformations projected to selected areas in the map and generated conformations along a path crossing these areas. The chosen example also shows that the generated conformations do not only represent motions of the two ubiquitin domains relative to each other but also internal motions of one domain. The lower part of the helix of the C-terminal domain unfolds to increase the number of interdomain contacts.

highlight such an example. In the bottom left, conformations generated along the orange path are shown. We see how the molecule changes from an "open" conformation, where the two subunits are relatively far apart, to a closed one, where the long α-helix of the C-terminal domain (right) gets in contact with the β-sheet of the N-terminal domain (left). Parallel to this global motion, there is also a more hidden conformational transition within the C-terminal domain when the N-terminal domain comes in contact. The lower part of the helix opens up to increase the number of interdomain contacts. In principle, the same can be seen in the original data from the simulation as indicated by the structure bundles from the two selected areas that are also shown in the video/Figure 9. However, it is much easier to spot this change in the generated conformations where random fluctuations of the molecule are not present. Figure 9 also shows good agreement between the original conformations selected in the map and the generated conformations. This demonstrates that EncoderMap indeed generates good representative conformations for the underlying sampling.

Ssa1 Hsp70 Chaperone. This second example is different in two major ways. First, it is even larger compared to diubiquitin. This allows us to further verify the ability of EncoderMap's improved version to deal with large proteins. Second, this example does not consist of a connected set of simulations. Instead it consists of two simulations sampling different, not overlapping regions of the conformational space. This allows analyzing how EncoderMap deals with such a situation where no information is available on how two areas of the conformational space are linked.

The example we chose is a part of the Ssa1 Hsp70 chaperone of the yeast species *Saccharomyces cerevisiae*.[34−36] The 210 residue part of the protein under investigation consists of a substrate binding domain in form of a β-barrel and a C-terminal domain consisting of multiple α-helical parts. The C-terminal domain is assumed to act like a lid that covers or uncovers the substrate binding domain.[35] The two simulations in this example are one simulation that started with a closed "lid" and a second simulation that started with an open "lid". In the first simulation, the lid stayed closed. In the second simulation, the lid closed but resulted in a very different closed conformation that does not correspond to the experimentally known structure. The data of both simulations were used as input for EncoderMap in its improved variant in analogy to the diubiquitin example. Figure 10 shows the traces of both

Figure 10. This figure shows the traces of two Ssa1 simulations in the two-dimensional EncoderMap. Chronologically subsequent conformations are connected with a line. The blue trace comes from a simulation that started in a conformation with a closed lid (corresponding to the experimentally known structure). The orange trace comes from a simulation that started from an open conformation and moved into a different closed conformation.

simulations projected to the two-dimensional map. Each point represents one projected conformation and chronologically subsequent conformations are connected with lines. The trace of one simulation (blue) only covers a comparatively narrow area. This is the simulation that started with the closed conformation. The small conformational changes that can happen inside this closed state result in a narrow distribution on the map. The second simulation (orange) covers a much wider area of the map. Interestingly, the trace resembles a

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

worm-like structure with a wider tail and a more narrow tail. This nicely reflects what one might expect for such a simulation starting with an open conformation with lots of flexibility and ending in a closed conformation allowing less conformational variety.

As described for the diubiquitin example already, we can now generate conformations for selected paths in the map to visualize important molecular motions. Figure 11 and the

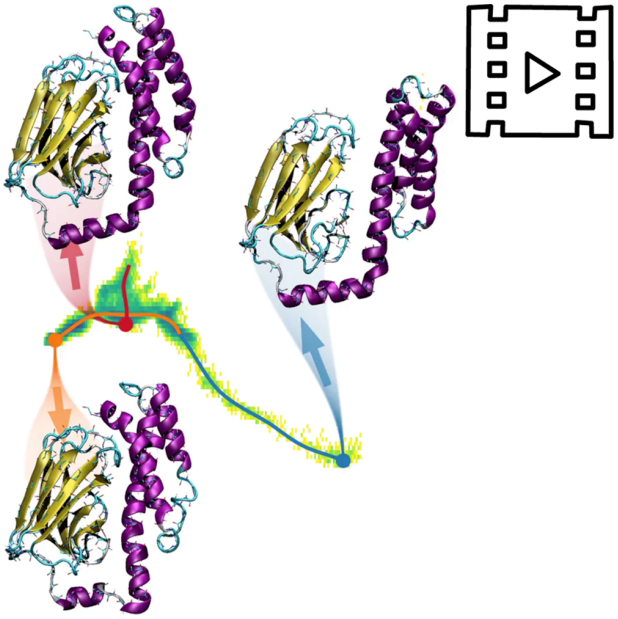


Figure 11. This figure and video show a 2D histogram of the zoomed in region of the "closed" simulation (blue area in Figure 10). The color ranges from yellow to blue for low to high density and empty bins are shown in white. Conformations generated for points on different paths are shown next to the map.

accompanying video show this for the "closed" simulation. They show the same map as Figure 10 but for a zoom into the region covering the simulation that started with the closed conformation. Conformations generated along three selected paths are shown. Irrespective of the path, we can clearly see the β-barrel of the binding domain and the helical parts of the "lid" domain. The β-strands in the β-barrel are accurately aligned. This is a tricky thing to achieve in dihedral space and is again a sign of the improved accuracy due to the introduced reconstruction of Cartesian coordinates during the training procedure. The different paths show different molecular motions that were sampled in the simulation: a rotational motion of the helical part (blue path) where the terminal helix moves closer to (or further away from) the β-barrel, a slight shifting motion (red path) of the whole lid around the β-barrel, and a displacement of the bend (orange path) in the helix next to the β-barrel in combination with a reorientation in the loop region of the nearest β-strands.

Figure 12 shows again the complete map with both simulations. Here. we were especially interested to see what happens in areas between the two simulations, that is, in white spaces in the map where no simulation data is available. Additionally to conformations generated on selected paths, we also generated conformations for points on a 200 by 200 grid. Whenever the generated conformation for a point on the grid

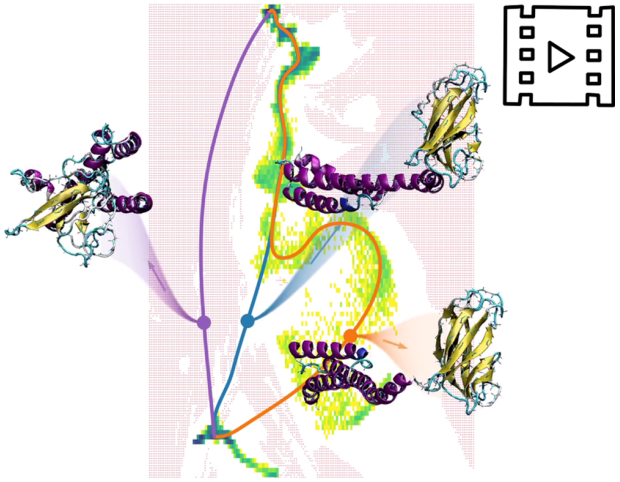


Figure 12. This figure and video show a 2D histogram of the complete map with both Ssa1 simulations. The one started form a closed conformation and the one started from an open conformation. It is the same map as shown in Figure 10. Tiny red dots indicate that conformations generated for these points contain at least one clash (two atom centers closer than 100 pm). The paths for which generated conformations are shown connect the two sampled regions that are not connected in the underlying simulation data.

contained at least one clash (i.e., atomic overlap), we mark this point with a red dot on the map. The areas of the map where simulation data are available hardly contain any red dots. This is in good agreement with the observation that for conformations generated along paths through the sampled area we hardly observe clashes. Also, conformations generated for points that are in close proximity to the sampled area are often free of clashes. Further away from the sampled regions we see a dense grid of red points indicating that all generated conformations for points in these regions contain clashes. To further analyze how EncoderMap deals with the not sampled space between the two simulations, we have generated conformations along three paths connecting the two disconnected regions from the two simulations (blue and orange areas in Figure 10). With the blue path, we tried to avoid "red-dot-territory" as good as possible. Indeed the generated conformations along this path look reasonable and no obvious clashes are observable. The orange path passes through a narrow area with clashes. In the generated conformations, we can see that these clashes are caused by the N-terminal β-strand. Apart from that, the conformations generated along this path also look reasonable. This path also nicely shows what happened in the simulation that started from the open conformation. The helical "lid" part attached to the opposite side of the β-barrel compared to where it attaches in the "correct" closed conformation. The purple path was deliberately chosen to go through areas far away from the sampled regions. Some of the conformations generated along this path are not at all plausible. Major parts of the protein travel trough each other and the β-barrel breaks apart.

Visualizing the sampled conformational space in a simplified way is the main goal behind EncoderMap. The above analysis however shows that it can potentially also be used to interpolate between or to slightly extrapolate away from sampled regions. In any case, the conformations generated for not sampled areas should be considered with caution, as there is no physical justification for these conformations. There is

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

nothing promoting the generation of reasonable conformations in these areas other than the regularization which prevents unnecessary complexity.

CONCLUSIONS

Previously, we had shown that a combination of a neural network autoencoder with multidimensional scaling results in a very advantageous dimensionality reduction technique, we termed EncoderMap. The multidimensional scaling aspect ensures that data points are arranged according to their distance in the high-dimensional space, which provides for a meaningful low-dimensional map. The neural network autoencoder constitutes a differentiable relation between the representations of different dimensionality. This can be done in a computationally efficient manner as the batchwise training of the network circumvents the quadratic scaling of other multidimensional scaling variants. With the autoencoder one does not only obtain a differentiable function mapping from the high-dimensional to the low-dimensional representation, which can be useful in combination with enhanced sampling methods that rely on a biasing potential defined in a low-dimensional space; one does also obtain a function mapping from the low-dimensional back to the high-dimensional space, which, for example, can be used to generate protein conformations for given points on the low-dimensional map. Now, we showed how this generation of protein conformations can be improved over the previously proposed basic generation in dihedral space. Generating conformations in dihedral space is problematic due to the short-range character of dihedrals. At the same time it is advantageous due to the unambiguousness of dihedrals and the lacking need to solve subsequent optimization problems to find the corresponding conformations in Cartesian space. Using dihedral angle output to reconstruct Cartesian coordinates during the training process allows preserving this advantage while solving the problem of inaccurate long-range order. The best results are achieved when the C_α pairwise distance cost, calculated from the reconstructed Cartesian coordinates, is not used from the beginning but turned on during the training process. This way, the network can freely learn the correct secondary structure without the need to cross barriers in the cost function caused by the C_α cost. These prefolded conformations are then corrected in their long-range order once the C_α cost is turned on. This strategy involving subsequent optimization according to local and then global criteria, which turned out to be very beneficial in this case, might also be useful for other optimization problems. It is also important how the deviations in the dihedrals and the C_α pairwise distances are calculated in the cost function. Using mean square deviation encourages the network to return nonphysical "mean conformations" with lots of clashes. Using mean absolute deviation instead results in distinctly fewer clashes. With these improvements, EncoderMap is now also able to accurately generate conformations of large proteins like diubiquitin or other multidomain proteins. The obtained low-dimensional map in combination with this ability to generate conformations for any point in the map allows for a new perspective on high-dimensional molecular data. Important molecular motions, which else might be hidden in the noisy wiggling of a molecule, can nicely be visualized.

DETAILS

EncoderMap. The used neural network autoencoder is made out of 7 fully connected layers: an input layer, 2 hidden layers with 128 neurons each, a bottleneck layer with 2 neurons, again 2 hidden layers with 128 neurons each, and an output layer. The hidden layers use tanh as activation function and all other layers use the identity function instead. The number of neurons of the input and output layers are equal to twice the number of dihedrals as sin and cos values of the dihedrals are used to circumvent periodicity issues. For example in the case of Diubiquitin with 152 amino acids there are 453 backbone dihedrals (Φ, Ψ, and Ω combined). The input and output layers therefore contain 906 neurons. The 906 sin and cos values obtained from the output layer are then converted back to 453 dihedral angle values. The pseudo code shown in Figure 13 describes how these dihedral angle values are used to generate Cartesian coordinates of a chain. The network is then trained using the cost function given in eq 1. The detailed contributions are

```
get the target dihedrals

get the Cartesian coordinates of all
 central backbone atoms
 (N-Cα-C-N-Cα-C ... ) of a conformation
 with mean bond lengths, mean bond
 angles, and all dihedral angles set to
 zero

for each dihedral in dihedrals do
    calculate the normed vector pointing
     from the current atom to the next
     (The current atom is the second of
     the four atoms defining the current
     dihedral angle. E.g. the Cα atom in
     case of the dihedral defined by the
     N-Cα-C-N atoms)

    use the vector and the dihedral value
     to assemble the rotation matrix

    shift the coordinates so that the
     current atom gets placed in the
     origin

    matrix multiply the coordinates of all
     following atoms with the rotation
     matrix
end
```

Figure 13. Pseudo code for the reconstruction of Cartesian coordinates from dihedral angles. The full code is available in the EncoderMap repository (https://github.com/AG-Peter/EncoderMap).

$$C_{\text{dihedral}} = \frac{\frac{1}{n_b n_d}\sum_{b=1}^{n_b}\sum_{i=1}^{n_d}\min(|d_{b,i}-\tilde{d}_{b,i}|, 2\pi-|d_{b,i}-\tilde{d}_{b,i}|)}{\frac{1}{n_a n_d}\sum_{a=1}^{n_a}\sum_{i=1}^{n_d}\min(|d_{a,i}-\bar{d}_i|, 2\pi-|d_{a,i}-\bar{d}_i|)} \quad (2)$$

where $d_{b,i}$ is the dihedral of the bth frame of the training batch and the ith position along the backbone, $\tilde{d}_{b,i}$ is the dihedral output of the network, $\bar{d}_i$ is the mean value for the dihedral at the ith position, n_b is the number of points in a training batch,

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

n_d is the number of dihedrals, and n_a is the number of points in the complete data set. The denominator of eq 2 represents the normalization with the dummy model that always returns the conformation with mean dihedrals. The $\min(x, 2\pi - x)$ part takes care of the periodicity of dihedrals.

$$C_{C_\alpha} = \frac{\frac{1}{n_b n_p}\sum_{b=1}^{n_b}\sum_{j=1}^{n_p}|p_{b,j} - \tilde{p}_{b,j}|}{\frac{1}{n_a n_p}\sum_{a=1}^{n_a}\sum_{j=1}^{n_p}|p_{a,j} - \bar{p}_j|} \tag{3}$$

where $p_{b,j}$ is the distance between the jth pair of C_α atoms in bth conformation of a training batch, $\tilde{p}_{b,j}$ is the equivalent distance in the conformation reconstructed from the dihedral output of the network, $\bar{p}_j$ is the equivalent distance in the conformation with mean dihedrals, and n_p is the number of pairwise distances between all C_α atoms. The third contribution to the cost function is the distance cost. Here, we use the cost function of the multidimensional scaling variant sketch-map[11]

$$C_{\text{distance}} = \frac{1}{\frac{n_b}{2}(n_b - 1)}\sum_{b=1}^{n_b}\sum_{k=b+1}^{n_b}[\text{SIG}_h(R_{bk}) - \text{SIG}_l(r_{bk})]^2 \tag{4}$$

where R_{bk} is the euclidean distance in the high-dimensional space (in this case the C_α pairwise distance space) between the bth and the kth point in the training batch and r_{bk} is the equivalent distance in the low-dimensional space (map). SIG_h and SIG_l are sigmoid functions defined as follows

$$\text{SIG}_{\sigma,a,b}(r) = 1 - (1 + (2^{a/b} - 1)(r/\sigma)^a)^{-b/a} \tag{5}$$

The sigmoid parameters used in the diubiquitin example are $\sigma = 400$, $a = 10$, $b = 5$ for the sigmoid applied to the high-d distances and $\sigma = 1$, $a = 2$, $b = 5$ for the low-d distances. For the chaperone example, we use the same sigmoid parameters except for the σ value of the sigmoid applied to the high-d distances which was set to $\sigma = 300$. The sketch-map literature[11] provides detailed information how to select these parameters. The scaling factors $k_{\text{dih.}}$ and k_{C_α} of eq 1 are set to 1 or 0 depending on whether the contribution is turned on or off. $k_{\text{dist.}}$ was set to 100.

The cost function was used to optimize the network with batches of 256 points using the Adam optimizer[40] with a learning rate of 0.001 and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as implemented in TensorFlow 1.9.[37] Weights were regularized using l2-regularization with a regularization constant of 0.001 and 0.0001 for the diubiquitin and the chaperone examples, respectively.

Diubiquitin Simulation Data. The diubiquitin data consist of 60 000 conformations from 12 atomistic 50 ns simulations. All simulations were started from an extended conformation where the two ubiquitin domains were (apart from the linker) not in contact. The GROMACS simulation package v5[41] with the GROMOS96 54a7 force field[42] was used to run the simulations. Further details about the simulations can be found in Berg et al. 2018.[33]

Chaperone Simulation Data. The example data set of Hsp70 Ssa1 consists of two simulations. The simulation that started in the closed conformation is 266.8 ns long, and 26 680 frames were included in the data set. The simulation that started in the open conformation is 543.5 ns long, and 54 350 frames were included in the data set. Both Simulations were performed using the Gromacs-4.6.5 package,[43] the Gromos54a7 force field,[42] and the SPC/E water model.[44] Further simulation details can be found in Hanebuth et al. 2016.[36]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.9b00675.

Histogram of all simulated conformations projected to the two-dimensional map (MP4)

Histogram of all simulated conformations projected to the two-dimensional map with generated conformations (MP4)

Comparison of original conformations, projected to selected areas in the map, and generated conformations (MP4)

2D histogram of the zoomed in region of the "closed" simulation (MP4)

2D histogram of the complete map with both Ssa1 simulations (MP4)

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: Christine.Peter@uni-konstanz.de.

ORCID

Tobias Lemke: 0000-0002-0593-2304

Andrej Berg: 0000-0002-5232-1995

Christine Peter: 0000-0002-1471-5440

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We gratefully acknowledge funding by the DFG through SFB969. We are also very grateful for computational resources of the bwHPC project (DFG grants INST 35/1134-1 FUGG, INST 37/935-1 FUGG and the state of Baden-Württemberg).

■ REFERENCES

(1) Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **1901**, *2*, 559−572.

(2) Mu, Y.; Nguyen, P. H.; Stock, G. Energy landscape of a small peptide revealed by dihedral angle principal component analysis. *Proteins: Struct., Funct., Bioinf.* **2004**, *58*, 45−52.

(3) Coifman, R. R.; Lafon, S.; Lee, A. B.; Maggioni, M.; Nadler, B.; Warner, F.; Zucker, S. W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 7426−7431.

(4) Pérez-Hernández, G.; Paul, F.; Giorgino, T.; De Fabritiis, G.; Noé, F. Identification of slow molecular order parameters for Markov model construction. *J. Chem. Phys.* **2013**, *139*, 015102.

(5) Wehmeyer, C.; Noé, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **2018**, *148*, 241703.

(6) Chen, W.; Tan, A. R.; Ferguson, A. L. Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.* **2018**, *149*, 072312.

(7) Sultan, M. M.; Wayment-Steele, H. K.; Pande, V. S. Transferable neural networks for enhanced sampling of protein dynamics. *J. Chem. Theory Comput.* **2018**, *14*, 1887−1894.

(8) Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, 072301.

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(9) Hernández, C. X.; Wayment-Steele, H. K.; Sultan, M. M.; Husic, B. E.; Pande, V. S. Variational encoding of complex dynamics. *Phys. Rev. E* **2018**, *97*, 062412.
(10) Cox, T. F.; Cox, M. A. *Multidimensional Scaling*; Chapman and Hall/CRC, 2000.
(11) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108*, 13023−13028.
(12) Tribello, G. A.; Gasparotto, P. Using dimensionality reduction to analyze protein trajectories. *Front. Mol. Biosci.* **2019**, *6*, 46.
(13) Lemke, T.; Peter, C. EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations. *J. Chem. Theory Comput.* **2019**, *15*, 1209−1215.
(14) Lemke, T. Dimensionality Reduction with EncoderMap. 2019, https://www.youtube.com/watch?v=JV59OABhNTY (accessed Oct 17, 2019).
(15) Neidigh, J. W.; Fesinmeyer, R. M.; Andersen, N. H. Designing a 20-residue protein. *Nat. Struct. Mol. Biol.* **2002**, *9*, 425.
(16) Sittel, F.; Stock, G. Perspective: Identification of collective variables and metastable states of protein dynamics. *J. Chem. Phys.* **2018**, *149*, 150901.
(17) Chiavazzo, E.; Covino, R.; Coifman, R. R.; Gear, C. W.; Georgiou, A. S.; Hummer, G.; Kevrekidis, I. G. Intrinsic map dynamics exploration for uncharted effective free-energy landscapes. *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114*, E5494−E5503.
(18) Kukharenko, O.; Sawade, K.; Steuer, J.; Peter, C. Using dimensionality reduction to systematically expand conformational sampling of intrinsically disordered peptides. *J. Chem. Theory Comput.* **2016**, *12*, 4726−4734.
(19) Lee, J.; Freddolino, P. L.; Zhang, Y. *From Protein Structure to Function with Bioinformatics*; Springer, 2017; pp 3−35.
(20) Moult, J.; Fidelis, K.; Kryshtafovych, A.; Schwede, T.; Tramontano, A. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins: Struct., Funct., Bioinf.* **2018**, *86*, 7−15.
(21) Šali, A.; Blundell, T. L. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* **1993**, *234*, 779−815.
(22) Karplus, K.; Barrett, C.; Hughey, R. Hidden Markov models for detecting remote protein homologies. *Bioinformatics* **1998**, *14*, 846−856.
(23) Yang, Y.; Faraggi, E.; Zhao, H.; Zhou, Y. Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics* **2011**, *27*, 2076−2082.
(24) Rohl, C. A.; Strauss, C. E.; Misura, K. M.; Baker, D. *Methods Enzymology*; Elsevier, 2004; Vol. *383*; pp 66−93.
(25) Boomsma, W.; Mardia, K. V.; Taylor, C. C.; Ferkinghoff-Borg, J.; Krogh, A.; Hamelryck, T. A generative, probabilistic model of local protein structure. *Proc. Natl. Acad. Sci. U.S.A.* **2008**, *105*, 8932−8937.
(26) Heffernan, R.; Paliwal, K.; Lyons, J.; Dehzangi, A.; Sharma, A.; Wang, J.; Sattar, A.; Yang, Y.; Zhou, Y. Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Sci. Rep.* **2015**, *5*, 11476.
(27) Wu, S.; Zhang, Y. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics* **2008**, *24*, 924−931.
(28) Kosciolek, T.; Jones, D. T. De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PloS one* **2014**, *9*, No. e92197.
(29) Aszódi, A.; Gradwell, M. J.; Taylor, W. R. Global fold determination from a small number of distance restraints. *J. Mol. Biol.* **1995**, *251*, 308−326.
(30) Lund, O.; Frimand, K.; Gorodkin, J.; Bohr, H.; Bohr, J.; Hansen, J.; Brunak, S. Protein distance constraints predicted by neural networks and probability density functions. *Protein Eng.* **1997**, *10*, 1241−1248.
(31) Kukic, P.; Mirabello, C.; Tradigo, G.; Walsh, I.; Veltri, P.; Pollastri, G. Toward an accurate prediction of inter-residue distances in proteins using 2D recursive neural networks. *BMC Bioinf.* **2014**, *15*, 6.
(32) Evans, R.; Jumper, J.; Kirkpatrick, J.; Sifre, L.; Green, T.; Qin, C.; Zidek, A.; Nelson, A.; Bridgland, A.; Penedones, H.; Petersen, S.; Simonyan, K.; Crossan, S.; Jones, D.; Silver, D.; Kavukcuoglu, K.; Hassabis, D.; Senior, A. De novo structure prediction with deep-learning based scoring. In *Thirteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstracts)*, 2018.
(33) Berg, A.; Kukharenko, O.; Scheffner, M.; Peter, C. Towards a molecular basis of ubiquitin signaling: A dual-scale simulation study of ubiquitin dimers. *PLoS Comput. Biol.* **2018**, *14*, No. e1006589.
(34) James, P.; Pfund, C.; Craig, E. A. Functional specificity among Hsp70 molecular chaperones. *Science* **1997**, *275*, 387−389.
(35) Hartl, F. U.; Bracher, A.; Hayer-Hartl, M. Molecular chaperones in protein folding and proteostasis. *Nature* **2011**, *475*, 324.
(36) Hanebuth, M. A.; Kityk, R.; Fries, S. J.; Jain, A.; Kriel, A.; Albanese, V.; Frickey, T.; Peter, C.; Mayer, M. P.; Frydman, J.; Deuerling, E. Multivalent contacts of the Hsp70 Ssb contribute to its architecture on ribosomes and nascent chain interaction. *Nat. Commun.* **2016**, *7*, 13695.
(37) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; https://www.tensorflow.org/ (accessed Oct 17, 2019).
(38) AlQuraishi, M. End-to-end differentiable learning of protein structure. *Cell Systems* **2019**, *8*, 292−301.
(39) Miettinen, K. *Nonlinear Multiobjective Optimization*; Springer Science & Business Media, 2012; Vol. *12*.
(40) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. arXiv:1412.6980. arXiv preprint **2014**.
(41) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1−2*, 19−25.
(42) Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; van Gunsteren, W. F. Definition and testing of the GROMOS force-field versions 54A7 and 54B7. *Eur. Biophys. J.* **2011**, *40*, 843.
(43) Hess, B.; Kutzner, C.; Van Der Spoel, D.; Lindahl, E. GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435−447.
(44) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The missing term in effective pair potentials. *J. Phys. Chem.* **1987**, *91*, 6269−6271.

DOI: 10.1021/acs.jcim.9b00675
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Summary and Outlook

4.1 Summary

Simplified representations of molecular systems are essential for solving two central problems of molecular simulations: simplified representations are valuable to reduce the computational cost of simulations, and simplified representations are indispensable to analyse the outcome of simulations.

In the publications included in this thesis, we demonstrated that machine learning approaches can be applied very successfully to obtaining simplified representations of molecular systems.

We showed that artificial neural networks can be used to extract high-dimensional free energy surfaces for coarse-grained simulation models. A unique contribution of our proposed method is the conversion of the problem of finding the free energy surface into a classification problem between conformations sampled in an atomistic reference simulation and "fake" conformations drawn from a known distribution. With this approach it is possible to create exceptionally coarse models where the molecule is represented with very few beads only. To get a substantial performance advantage compared to atomistic simulations, such exceptionally coarse models are especially important.

With EncoderMap, we introduced a new dimensionality reduction algorithm that combines advantages of multidimensional scaling and neural network autoencoders. Our approach circumvents the unfavorable quadratic scaling of multidimensional scaling and drastically reduces the time required to perform

the dimensionality reduction. With the trained neural network, we obtain a differentiable function linking the representations of different dimensionality, which allows to combine EncoderMap with enhanced sampling approaches that involve biasing potentials defined in a low-dimensional space. At the same time, the multidimensional scaling aspect ensures that data is arranged in an informative way where distances in the low-dimensional map become meaningful. Besides projecting high-dimensional data into a low-dimensional map, it is also possible to project points in the map back into the high-dimensional space. With this tool, for example, molecule conformations can be generated for given points in a two-dimensional conformational map. The generated conformations only reflect the most important conformational changes identified during the dimensionality reduction process, which allows for a unique, new perspective on conformational data.

The combined usage of atomistic simulation together with our proposed coarse-graining approach and our developed dimensionality reduction scheme allows to benefit from all these levels of resolution. We can profit from a detailed sampling in the atomistic space, a fast sampling in the coarse-grained space, and we can monitor and guide everything with maps obtained by dimensionality reduction.

4.2 Outlook

Among many other studies, our work demonstrates the great potential of machine learning approaches in the field of molecular simulations. In the next years, such methods will continue to become integral parts in the simulation toolbox, but a great deal of work is still required to advance this process. Here, I describe what I believe are the most pressing challenges or most promising approaches to building on the methodology provided in the context of this thesis.

In the years of applying machine learning, I more and more came to the conclusion that we should only use machine learning for the things that we do not know already. This sounds trivial, but it is not always that simple. In our neural network coarse-graining approach, for example, we made the network learn all the interactions from scratch. This worked surprisingly well, which demonstrates the potential of our approach, however, it would be even more promising to make the network only learn the difference between some classical coarse-grained model and the atomistic reference. This way, we could profit from all the progress made in classical coarse-graining and the resources of the network would be focused on learning the missing pieces.

We incorporated this idea, of not learning what we anyway know, in EncoderMaps conformation generation already. In principle, we could train the network to generate molecule conformations in Cartesian coordinates directly, but then the network would spend most of its resources to learn the constitution of the molecule. It would learn how long bonds are and which atoms are connected to each other. These are all things that we know perfectly well when we set up a simulation, and there is no need for the neural network to learn these things. Instead, we trained the network only to learn the backbone dihedral angles and reconstructed the Cartesian coordinates using our knowledge about the molecule. It is important to stick to this strategy, however, it needs to be extended. Currently only chain-like molecules, like proteins or other polymers, are supported. The reconstruction of Cartesian coordinates needs to be extended to branched and ring-like systems to make EncoderMaps conformation generation tool universally applicable.

With these changes, both the neural network coarse graining approach and EncoderMap have a good chance to become well established in the simulation community.

Zusammenfassung

Vereinfachte Repräsentationen molekularer Systeme sind für die Lösung zweier zentraler Probleme molekularer Simulationen unerlässlich: Zum einen könne sie den Rechenaufwand einer Simulation enorm senken und zum anderen ermöglichen sie die Analyse der erhaltenen Ergebnisse.

In den in dieser Arbeit enthaltenen Veröffentlichungen haben wir gezeigt, dass maschinelles Lernen sehr erfolgreich angewendet werden kann, um vereinfachte Repräsentationen molekularer Systeme zu erhalten.

Wir haben gezeigt, dass künstliche neuronale Netzwerke verwendet werden können, um hochdimensionale freie Energielandschaften für grobskalige Simulationsmodelle zu extrahieren. Eine Besonderheit unserer vorgeschlagenen Methode ist, dass das Problem die freien Energielandschaft zu finden in ein Klassifizierungsproblem umgewandelt wird. Die Klassifizierung erfolgt zwischen Konformationen, die in einer atomistischen Referenzsimulation gefunden wurden, und "fake" Konformationen, die aus einer bekannten Verteilung gezogen wurden. Mit diesem Ansatz ist es möglich, außergewöhnlich grobe Modelle zu erstellen, bei denen das Molekül nur mit sehr wenigen Einheiten dargestellt wird. Solche außergewöhnlich grobskalige Modelle sind besonders wichtig, um einen wesentlichen Effizienzvorteil gegenüber atomistischen Simulationen zu erzielen.

Mit EncoderMap haben wir einen neuen Algorithmus zur Reduzierung der Dimensionalität eingeführt, der die Vorteile von multidimensionaler Skalierung und neuronalen Netzwerk-Autoencodern kombiniert. Unser Ansatz umgeht die ungünstige quadratische Skalierung der mehrdimensionalen Skalierung und

reduziert die für die Durchführung der Dimensionsreduzierung erforderliche Zeit drastisch. Mit dem trainierten neuronalen Netzwerk erhalten wir eine differenzierbare Funktion, die die Repräsentationen unterschiedlicher Dimensionalität verbindet und es ermöglicht, EncoderMap mit enhanced sampling-Ansätzen zu kombinieren, bei denen in einem niedrigdimensionalen Raum Bias-Potentiale definiert werden. Gleichzeitig sorgt der multidimensionale Skalierungsaspekt dafür, dass Daten auf informative Weise angeordnet werden, wobei Entfernungen in der niedrigdimensionalen Karte aussagekräftig werden. Neben der Projektion hochdimensionaler Daten in eine niedrigdimensionale Karte ist es auch möglich, Punkte in der Karte zurück in den hochdimensionalen Raum zu projizieren. Mit diesem Werkzeug können beispielsweise Molekülkonformationen für bestimmte Punkte in einer zweidimensionalen Konformationskarte erzeugt werden. Die generierten Konformationen spiegeln nur die wichtigsten Konformationsänderungen wider, die während des Dimensionsreduktionsprozesses identifiziert wurden, was eine einzigartige, neue Perspektive auf Konformationsdaten ermöglicht.

Die Kombination von atomistischen Simulationen und unserem vorgeschlagenen Ansatz um grobskalige Modelle zu erzeugen, sowie unserem entwickelten Dimensionsreduktionsschema ermöglicht es, von all diesen Auflösungsstufen zu profitieren. Dadurch können wir die Vorteile einer detaillierten Erkundung im atomistischen Raum mit der Schnelligkeit von grobeskaligen Simulationsmodellen vereinigen und zusätzlich den Verlauf mit Hilfe von Karten, die durch Dimensionsreduktion erhalten wurden, beobachten und steuern.

Bibliography

[Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[Abraham et al., 2015] Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. (2015). Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25.

[Allegre et al., 1995] Allegre, C. J., Manhes, G., and Göpel, C. (1995). The age of the earth. *Geochim. Cosmochim. Acta*, 59(8):1445–1456.

[Allen et al., 2009] Allen, R. J., Valeriani, C., and ten Wolde, P. R. (2009). Forward flux sampling for rare event simulations. *J. Phys.: Condens. Matter*, 21(46):463102.

[AlQuraishi, 2019] AlQuraishi, M. (2019). End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301.

[Aschauer et al., 2010] Aschauer, U., Spagnoli, D., Bowen, P., and Parker, S. C. (2010). Growth modification of seeded calcite using carboxylic acids: Atomistic simulations. *J. Colloid Interface Sci.*, 346(1):226 – 231.

[Aszodi et al., 1995] Aszodi, A., Gradwell, M., and Taylor, W. (1995). Global fold determination from a small number of distance restraints. *J. Mol. Biol.*, 251(2):308–326.

[Ayinde and Zurada, 2018] Ayinde, B. O. and Zurada, J. M. (2018). Deep learning of constrained autoencoders for enhanced understanding of data. *arXiv preprint arXiv:1802.00003*.

[Babin et al., 2013] Babin, V., Leforestier, C., and Paesani, F. (2013). Development of a "first principles" water potential with flexible monomers: Dimer potential energy surface, vrt spectrum, and second virial coefficient. *J. Chem. Theory Comput.*, 9(12):5395–5403.

[Barducci et al., 2008] Barducci, A., Bussi, G., and Parrinello, M. (2008). Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Phys. Rev. Lett.*, 100(2):020603.

[Bartók et al., 2010] Bartók, A. P., Payne, M. C., Kondor, R., and Csányi, G. (2010). Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104(13):136403.

[Behler, 2011] Behler, J. (2011). Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.*, 13(40):17930–17955.

[Behler and Parrinello, 2007] Behler, J. and Parrinello, M. (2007). Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401.

[Bejagam et al., 2018] Bejagam, K. K., Singh, S., An, Y., and Deshmukh, S. A. (2018). Machine-learned coarse-grained models. *J. Phys. Chem. Lett.*, 9(16):4667–4672.

[Bereau and Swendsen, 2009] Bereau, T. and Swendsen, R. H. (2009). Optimized convergence for multiple histogram analysis. *J. Comput. Phys.*, 228(17):6119 – 6129.

[Berendsen et al., 1987] Berendsen, H., Grigera, J., and Straatsma, T. (1987). The missing term in effective pair potentials. *J. Phys. Chem.*, 91(24):6269–6271.

[Berendsen et al., 1984] Berendsen, H. J., Postma, J. v., van Gunsteren, W. F., DiNola, A., and Haak, J. (1984). Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81(8):3684–3690.

[Berendsen et al., 1981] Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F., and Hermans, J. (1981). Interaction models for water in relation to protein hydration. *Intermol. Forces*, pages 331–342.

[Berg et al., 2018] Berg, A., Kukharenko, O., Scheffner, M., and Peter, C. (2018). Towards a molecular basis of ubiquitin signaling: A dual-scale simulation study of ubiquitin dimers. *PLoS Comput. Biol.*, 14(11):e1006589.

[Bernardi et al., 2015] Bernardi, R. C., Melo, M. C., and Schulten, K. (2015). Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochim. Biophys. Acta, Gen. Subj.*, 1850(5):872 – 877.

[Bewernitz et al., 2012] Bewernitz, M. A., Gebauer, D., Long, J., Cölfen, H., and Gower, L. B. (2012). A metastable liquid precursor phase of calcium carbonate and its interactions with polyaspartate. *Faraday Discuss.*, 159:291–312.

[Bishop, 1995] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

[Blank et al., 1995] Blank, T. B., Brown, S. D., Calhoun, A. W., and Doren, D. J. (1995). Neural network models of potential energy surfaces. *J. Chem. Phys.*, 103(10):4129–4137.

[Boomsma et al., 2008] Boomsma, W., Mardia, K. V., Taylor, C. C., Ferkinghoff-Borg, J., Krogh, A., and Hamelryck, T. (2008). A generative, probabilistic model of local protein structure. *Proc. Natl. Acad. Sci. U. S. A.*, 105(26):8932–8937.

[Bowman et al., 2014] Bowman, G. R., Pande, V. S., and Noé, F., editors (2014). *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, volume 797. Springer Netherlands, Dordrecht.

[Brand et al., 2015] Brand, A., Allen, L., Altman, M., Hlava, M., and Scott, J. (2015). Beyond authorship: attribution, contribution, collaboration, and credit. *Learned Publishing*, 28(2):151–155.

[Bulo et al., 2007] Bulo, R. E., Donadio, D., Laio, A., Molnar, F., Rieger, J., and Parrinello, M. (2007). "site binding" of ca2+ ions to polyacrylates in water: a molecular dynamics study of coiling and aggregation. *Macromolecules*, 40(9):3437–3442.

[Burges et al., 2005] Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. N. (2005). Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96.

[Bussi, 2013] Bussi, G. (2013). Hamiltonian replica exchange in GROMACS: a flexible implementation. *Mol. Phys.*, 112(3-4):379–384.

[Bussi et al., 2007] Bussi, G., Donadio, D., and Parrinello, M. (2007). Canonical sampling through velocity rescaling. *J. Chem. Phys.*, 126(1):014101.

[Ceriotti et al., 2011] Ceriotti, M., Tribello, G. A., and Parrinello, M. (2011). Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U. S. A.*, 108(32):13023–13028.

[Ceriotti et al., 2013] Ceriotti, M., Tribello, G. A., and Parrinello, M. (2013). Demonstrating the transferability and the descriptive power of sketch-map. *J. Chem. Theory Comput.*, 9(3):1521–1532.

[Chen and Trout, 2008] Chen, J. and Trout, B. L. (2008). Computational study of solvent effects on the molecular self-assembly of tetrolic acid in solution and implications for the polymorph formed from crystallization. *J. Phys. Chem. B*, 112(26):7794–7802.

[Chen et al., 2018] Chen, W., Tan, A. R., and Ferguson, A. L. (2018). Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.*, 149(7):072312.

[Chiavazzo et al., 2017] Chiavazzo, E., Covino, R., Coifman, R. R., Gear, C. W., Georgiou, A. S., Hummer, G., and Kevrekidis, I. G. (2017). Intrinsic map dynamics exploration for uncharted effective free-energy landscapes. *Proc. Natl. Acad. Sci. U. S. A.*, 114(28):E5494–E5503.

[Coifman et al., 2005] Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc. Natl. Acad. Sci. U. S. A.*, 102(21):7426–7431.

[Collobert et al., 2002] Collobert, R., Bengio, S., and Marithoz, J. (2002). Torch: A modular machine learning software library.

[Cox and Cox, 2000] Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.

[Cybenko, 1988] Cybenko, G. (1988). *Continuous valued neural networks with two hidden layers are sufficient*. University of Illinois at Urbana-Champaign. Center for Supercomputing Research and Development.

[Darden et al., 1993] Darden, T., York, D., and Pedersen, L. (1993). Particle mesh ewald: An n□log(n) method for ewald sums in large systems. *J. Chem. Phys.*, 98(12):10089–10092.

[Das et al., 2011] Das, P., Frewen, T. A., Kevrekidis, I. G., and Clementi, C. (2011). Think globally, move locally: Coarse graining of effective free energy surfaces. In *Coping with Complexity: Model Reduction and Data Analysis*, pages 113–131. Springer.

[Das et al., 2006] Das, P., Moll, M., Stamati, H., Kavraki, L. E., and Clementi, C. (2006). Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc. Natl. Acad. Sci. U. S. A.*, 103(26):9885–9890.

[Daura et al., 1999] Daura, X., Gademann, K., Jaun, B., Seebach, D., Van Gunsteren, W. F., and Mark, A. E. (1999). Peptide folding: when simulation meets experiment. *Angew. Chem., Int. Ed.*, 38(1-2):236–240.

[Demichelis et al., 2011] Demichelis, R., Raiteri, P., Gale, J. D., Quigley, D., and Gebauer, D. (2011). Stable prenucleation mineral clusters are liquid-like ionic polymers. *Nat. Commun.*, 2:590.

[Deng et al., 2013] Deng, N.-j., Dai, W., and Levy, R. M. (2013). How kinetics within the unfolded state affects protein folding: An analysis based on markov state models and an ultra-long md trajectory. *J. Phys. Chem. B*, 117(42):12787–12799.

[Ding and He, 2004] Ding, C. and He, X. (2004). K-means clustering via principal component analysis. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 29–36, New York, NY, USA. ACM.

[Dror et al., 2012] Dror, R. O., Dirks, R. M., Grossman, J., Xu, H., and Shaw, D. E. (2012). Biomolecular simulation: a computational microscope for molecular biology. *Annu. Rev. Biophys.*, 41:429–452.

[Duan et al., 2019] Duan, K., He, Y., Li, Y., Liu, J., Zhang, J., Hu, Y., Lin, R., Wang, X., Deng, W., and Li, L. (2019). Machine-learning assisted coarse-grained model for epoxies over wide ranges of temperatures and cross-linking degrees. *Materials & Design*, 183:108130.

[Durrant and McCammon, 2011] Durrant, J. D. and McCammon, J. A. (2011). Molecular dynamics simulations and drug discovery. *BMC Biology*, 9(1):1–9.

[Essmann et al., 1995] Essmann, U., Perera, L., Berkowitz, M. L., Darden, T., Lee, H., and Pedersen, L. G. (1995). A smooth particle mesh ewald method. *J. Chem. Phys.*, 103(19):8577–8593.

[Evans et al., 2018] Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, A., Nelson, A., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Jones, D., Silver, D., Kavukcuoglu, K., Hassabis, D., and Senior, A. (2018). De novo structure prediction with deep-learning based scoring. *Thirteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstracts)*.

[Frenkel and Smit, 2001] Frenkel, D. and Smit, B. (2001). *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier.

[Fukunishi et al., 2002] Fukunishi, H., Watanabe, O., and Takada, S. (2002). On the hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *J. Chem. Phys.*, 116(20):9058–9067.

[Galvelis and Sugita, 2017] Galvelis, R. and Sugita, Y. (2017). Neural network and nearest neighbor algorithms for enhancing sampling of molecular dynamics. *J. Chem. Theory Comput*, 13(6):2489–2500.

[Garrido and Juste, 1998] Garrido, L. and Juste, A. (1998). On the determination of probability density functions by using neural networks. *Comput. Phys. Commun.*, 115(1):25–31.

[Gebauer and Cölfen, 2011] Gebauer, D. and Cölfen, H. (2011). Prenucleation clusters and non-classical nucleation. *Nano Today*, 6(6):564–584.

[Grubmüller, 1995] Grubmüller, H. (1995). Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E*, 52:2893–2906.

[Grubmüller, 1995] Grubmüller, H. (1995). Predicting slow structural transitions in macromolecular systems: Conformational flooding. *Phys. Rev. E*, 52(3):2893.

[Gutmann and Hyvärinen, 2010] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

[Hamelberg et al., 2004] Hamelberg, D., Mongan, J., and McCammon, J. A. (2004). Accelerated molecular dynamics: A promising and efficient simulation method for biomolecules. *J. Chem. Phys.*, 120(24):11919–11929.

[Handley and Popelier, 2010] Handley, C. M. and Popelier, P. L. (2010). Potential energy surfaces fitted by artificial neural networks. *J. Phys. Chem. A*, 114(10):3371–3383.

[Hanebuth et al., 2016] Hanebuth, M. A., Kityk, R., Fries, S. J., Jain, A., Kriel, A., Albanese, V., Frickey, T., Peter, C., Mayer, M. P., Frydman, J., and Deuerling, E. (2016). Multivalent contacts of the hsp70 ssb contribute to its architecture on ribosomes and nascent chain interaction. *Nat. Commun.*, 7:13695.

[Harada and Kataoka, 1995] Harada, A. and Kataoka, K. (1995). Formation of polyion complex micelles in an aqueous milieu from a pair of oppositely-charged block copolymers with poly (ethylene glycol) segments. *Macromolecules*, 28(15):5294–5299.

[Harris, 2018] Harris, S. (2018). Mind the gap: Bridging between atomistic models and the continuum limit with fluctuating finite element analysis. Computational Biophysics on Your Desktop: Is That Possible?

[Hartl et al., 2011] Hartl, F. U., Bracher, A., and Hayer-Hartl, M. (2011). Molecular chaperones in protein folding and proteostasis. *Nature*, 475(7356):324.

[Heffernan et al., 2015] Heffernan, R., Paliwal, K., Lyons, J., Dehzangi, A., Sharma, A., Wang, J., Sattar, A., Yang, Y., and Zhou, Y. (2015). Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning. *Scientific reports*, 5:11476.

[Hernández et al., 2018] Hernández, C. X., Wayment-Steele, H. K., Sultan, M. M., Husic, B. E., and Pande, V. S. (2018). Variational encoding of complex dynamics. *Phys. Rev. E*, 97(6):062412.

[Hess et al., 1997] Hess, B., Bekker, H., Berendsen, H. J., and Fraaije, J. G. (1997). Lincs: a linear constraint solver for molecular simulations. *J. Comput. Chem.*, 18(12):1463–1472.

[Hess et al., 2008] Hess, B., Kutzner, C., Van Der Spoel, D., and Lindahl, E. (2008). Gromacs 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation. *J. Chem. Theory Comput.*, 4(3):435–447.

[Hinton et al., 2012] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

[Huber et al., 1994] Huber, T., Torda, A. E., and Van Gunsteren, W. F. (1994). Local elevation: a method for improving the searching properties of molecular dynamics simulation. *J. Comput.-Aided Mol. Des.*, 8(6):695–708.

[Hukushima and Nemoto, 1996] Hukushima, K. and Nemoto, K. (1996). Exchange monte carlo method and application to spin glass simulations. *J. Phys. Soc. Jpn.*, 65(6):1604–1608.

[Hunkler et al., 2019] Hunkler, S., Lemke, T., Peter, C., and Kukharenko, O. (2019). Back-mapping based sampling: Coarse grained free energy landscapes as a guideline for atomistic exploration. *J. Chem. Phys.*, 151(15):154102.

[Ischtwan and Collins, 1994] Ischtwan, J. and Collins, M. A. (1994). Molecular potential energy surfaces by interpolation. *J. Chem. Phys.*, 100(11):8080–8088.

[Jain et al., 2014] Jain, A., Jochum, M., and Peter, C. (2014). Molecular dynamics simulations of peptides at the air–water interface: Influencing factors on peptide-templated mineralization. *Langmuir*, 30(51):15486–15495.

[James et al., 1997] James, P., Pfund, C., and Craig, E. A. (1997). Functional specificity among hsp70 molecular chaperones. *Science*, 275(5298):387–389.

[John and Csányi, 2017] John, S. and Csányi, G. (2017). Many-body coarse-grained interactions using gaussian approximation potentials. *The Journal of Physical Chemistry B*, 121(48):10934–10949.

[Jorgensen et al., 1983] Jorgensen, W. L., Chandrasekhar, J., Madura, J. D., Impey, R. W., and Klein, M. L. (1983). Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79(2):926–935.

[Juraszek and Bolhuis, 2006] Juraszek, J. and Bolhuis, P. (2006). Sampling the multiple folding mechanisms of trp-cage in explicit solvent. *Proc. Natl. Acad. Sci. U. S. A.*, 103(43):15859–15864.

[Kahlen et al., 2015] Kahlen, J., Peter, C., and Donadio, D. (2015). Molecular simulation of oligo-glutamates in a calcium-rich aqueous solution : insights into peptide-induced polymorph selection. *CrystEngComm*, 17(36):6863–6867.

[Karplus et al., 1998] Karplus, K., Barrett, C., and Hughey, R. (1998). Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856.

[Karplus and Kuriyan, 2005] Karplus, M. and Kuriyan, J. (2005). Molecular dynamics and protein function. *Proc. Natl. Acad. Sci. U. S. A. of the United States of America*, 102(19):6679–6685.

[Kim et al., 2015] Kim, S. B., Dsilva, C. J., Kevrekidis, I. G., and Debenedetti, P. G. (2015). Systematic characterization of protein folding pathways using diffusion maps: Application to trp-cage miniprotein. *J. Chem. Phys.*, 142(8):02B613_1.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kobrak, 2003] Kobrak, M. N. (2003). Systematic and statistical error in histogram-based free energy calculations. *J. Comput. Chem.*, 24(12):1437–1446.

[Kolmogorov, 1933] Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, 4:83–91.

[Kosciolek and Jones, 2014] Kosciolek, T. and Jones, D. T. (2014). De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PloS one*, 9(3):e92197.

[Kubelka et al., 2004] Kubelka, J., Hofrichter, J., and Eaton, W. A. (2004). The protein folding 'speed limit'. *Curr. Opin. Struct. Biol.*, 14(1):76–88.

[Kukharenko et al., 2016] Kukharenko, O., Sawade, K., Steuer, J., and Peter, C. (2016). Using dimensionality reduction to systematically expand conformational sampling of intrinsically disordered peptides. *J. Chem. Theory Comput.*, 12(10):4726–4734.

[Kukic et al., 2014] Kukic, P., Mirabello, C., Tradigo, G., Walsh, I., Veltri, P., and Pollastri, G. (2014). Toward an accurate prediction of inter-residue distances in proteins using 2d recursive neural networks. *BMC Bioinf.*, 15(1):6.

[Kumar et al., 1992] Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., and Kollman, P. A. (1992). The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *J. Comput. Chem.*, 13(8):1011–1021.

[Laio and Parrinello, 2002] Laio, A. and Parrinello, M. (2002). Escaping free-energy minima. *Proc. Natl. Acad. Sci. U. S. A.*, 99(20):12562–12566.

[Le Roux et al., 2011] Le Roux, N., Bengio, Y., and Fitzgibbon, A. (2011). 15 improving first and second-order methods by modeling uncertainty. *Optimization for Machine Learning*, page 403.

[LeCun et al., 1990] LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lee et al., 2017] Lee, J., Freddolino, P. L., and Zhang, Y. (2017). Ab initio protein structure prediction. In *From protein structure to function with bioinformatics*, pages 3–35. Springer.

[Lemke, 2019] Lemke, T. (2019). Dimensionality reduction with encodermap. https://www.youtube.com/watch?v=JV59OABhNTY (accessed Jan 27, 2020).

[Lemke et al., 2019] Lemke, T., Berg, A., Jain, A., and Peter, C. (2019). Encodermap (ii): Visualizing important molecular motions with improved generation of protein conformations. *J. Chem. Inf. Model.*

[Lemke and Peter, 2017] Lemke, T. and Peter, C. (2017). Neural network based prediction of conformational free energies-a new route toward coarse-grained simulation models. *J. Chem. Theory Comput.*

[Lemke and Peter, 2019] Lemke, T. and Peter, C. (2019). Encodermap: Dimensionality reduction and generation of molecule conformations. *J. Chem. Theory Comput.*

[Lemke et al., 2018] Lemke, T., Peter, C., and Kukharenko, O. (2018). Efficient sampling and characterization of free energy landscapes of ion–peptide systems. *J. Chem. Theory Comput.*

[Lindorff-Larsen and Ferkinghoff-Borg, 2009] Lindorff-Larsen, K. and Ferkinghoff-Borg, J. (2009). Similarity measures for protein ensembles. *PLoS One*, 4(1):1–13.

[Liu et al., 2005] Liu, P., Kim, B., Friesner, R. A., and Berne, B. J. (2005). Replica exchange with solute tempering: A method for sampling biological systems in explicit water. *Proc. Natl. Acad. Sci. U. S. A.*, 102(39):13749–13754.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Trans. Inf. Theory*, 28(2):129–137.

[Lorenz et al., 2004] Lorenz, S., Groß, A., and Scheffler, M. (2004). Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.*, 395(4):210–215.

[Lowenstam and Weiner, 1989] Lowenstam, H. A. and Weiner, S. (1989). *On biomineralization*. Oxford University Press, USA.

[Lund et al., 1997] Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen, J., and Brunak, S. (1997). Protein distance constraints predicted by neural networks and probability density functions. *Protein Eng.*, 10(11):1241–1248.

[Májek and Ron, 2010] Májek, P. and Ron, E. (2010). Milestoning without a reaction coordinate. *J. Chem. Theory Comput.*, 6(6):1805–1817.

[Marinari and Parisi, 1992] Marinari, E. and Parisi, G. (1992). Simulated tempering: A new monte carlo scheme. *Europhys. Lett.*, 19(6):451.

[Meldrum and Cölfen, 2008] Meldrum, F. C. and Cölfen, H. (2008). Controlling mineral morphologies and structures in biological and synthetic systems. *Chem. Rev.*, 108(11):4332–4432.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092.

[Mey et al., 2014] Mey, A. S. J. S., Wu, H., and Noé, F. (2014). xtram: Estimating equilibrium expectations from time-correlated simulation data at multiple thermodynamic states. *Phys. Rev. X*, 4:041018.

[Miettinen, 2012] Miettinen, K. (2012). *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine learning*. McGraw hill.

[Modha and Fainman, 1994] Modha, D. S. and Fainman, Y. (1994). A learning law for density estimation. *IEEE Transactions on Neural Networks*, 5(3):519–523.

[Moult et al., 2018] Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2018). Critical assessment of methods of protein structure prediction (casp)—round xii. *Proteins: Struct., Funct., Bioinf.*, 86:7–15.

[Mu et al., 2005] Mu, Y., Nguyen, P. H., and Stock, G. (2005). Energy landscape of a small peptide revealed by dihedral angle principal component analysis. *Proteins: Struct., Funct., Bioinf.*, 58(1):45–52.

[Neidigh et al., 2002] Neidigh, J. W., Fesinmeyer, R. M., and Andersen, N. H. (2002). Designing a 20-residue protein. *Nat. Struct. Mol. Biol.*, 9(6):425.

[Ng, 2004] Ng, A. Y. (2004). Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

[Noid, 2013] Noid, W. (2013). Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.*, 139(9):09B201_1.

[Noid et al., 2008] Noid, W., Chu, J.-W., Ayton, G. S., Krishna, V., Izvekov, S., Voth, G. A., Das, A., and Andersen, H. C. (2008). The multiscale coarse-graining method. i. a rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.*, 128(24):244114.

[Oleinikovas et al., 2016] Oleinikovas, V., Saladino, G., Cossins, B. P., and Gervasio, F. L. (2016). Understanding cryptic pocket formation in protein targets by enhanced sampling simulations. *J. Am. Chem. Soc.*, 138(43):14257–14263.

[Paschek et al., 2008] Paschek, D., Hempel, S., and García, A. E. (2008). Computing the stability diagram of the trp-cage miniprotein. *Proc. Natl. Acad. Sci. U. S. A.*, 105(46):17754–17759.

[Pearson, 1901] Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[Pérez-Hernández et al., 2013] Pérez-Hernández, G., Paul, F., Giorgino, T., De Fabritiis, G., and Noé, F. (2013). Identification of slow molecular order parameters for markov model construction. *J. Chem. Phys.*, 139(1):07B604_1.

[Peter and Kremer, 2009] Peter, C. and Kremer, K. (2009). Multiscale simulation of soft matter systems–from the atomistic to the coarse-grained level and back. *Soft Matter*, 5(22):4357–4366.

[Piana et al., 2005] Piana, S., Reyhani, M., and Gale, J. D. (2005). Simulating micrometre-scale crystal growth from solution. *Nature*, 438(7064):70–73.

[Picker et al., 2012] Picker, A., Kellermeier, M., Seto, J., Gebauer, D., and Cölfen, H. (2012). The multiple effects of amino acids on the early stages of calcium carbonate crystallization. *Zeitschrift für Kristallographie - Crystalline Materials*, 227:744–757.

[Potestio et al., 2014] Potestio, R., Peter, C., and Kremer, K. (2014). Computer simulations of soft matter: Linking the scales. *Entropy*, 16(8):4199–4245.

[Pound and Mer, 1952] Pound, G. M. and Mer, V. K. L. (1952). Kinetics of crystalline nucleus formation in supercooled liquid tin1, 2. *J. Am. Chem. Soc.*, 74(9):2323–2332.

[Qian, 1999] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151.

[Qiu et al., 2002] Qiu, L., Pabit, S. A., Roitberg, A. E., and Hagen, S. J. (2002). Smaller and faster: The 20-residue trp-cage protein folds in 4 μs. *J. Am. Chem. Soc.*, 124(44):12952–12953.

[Raiteri et al., 2012] Raiteri, P., Demichelis, R., Gale, J. D., Kellermeier, M., Gebauer, D., Quigley, D., Wright, L. B., and Walsh, T. R. (2012). Exploring the influence of organic species on pre-and post-nucleation calcium carbonate. *Faraday Discuss.*, 159(1):61–85.

[Reith et al., 2003] Reith, D., Pütz, M., and Müller-Plathe, F. (2003). Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.*, 24(13):1624–1636.

[Ribeiro et al., 2018] Ribeiro, J. M. L., Bravo, P., Wang, Y., and Tiwary, P. (2018). Reweighted autoencoded variational bayes for enhanced sampling (rave). *J. Chem. Phys.*, 149(7):072301.

[Ripley, 2007] Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

[Rodriguez and Laio, 2014] Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496.

[Rohl et al., 2004] Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. (2004). Protein structure prediction using rosetta. In *Methods Enzymol.*, volume 383, pages 66–93. Elsevier.

[Rohrdanz et al., 2013] Rohrdanz, M. A., Zheng, W., and Clementi, C. (2013). Discovering mountain passes via torchlight: Methods for the definition of reaction coordinates and pathways in complex macromolecular reactions. *Annu. Rev. Phys. Chem.*, 64(1):295–316.

[Rosta and Hummer, 2015] Rosta, E. and Hummer, G. (2015). Free energies from dynamic weighted histogram analysis using unbiased markov state model. *J. Chem. Theory Comput.*, 11(1):276–285.

[Rothwell et al., 2004] Rothwell, A. C., Jagger, L. D., Dennis, W. R., and Clarke, D. R. (2004). Intelligent spam detection system using an updateable neural analysis engine. US Patent 6,769,016.

[Šali and Blundell, 1993] Šali, A. and Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.*, 234(3):779–815.

[Salvalaglio et al., 2012] Salvalaglio, M., Vetter, T., Giberti, F., Mazzotti, M., and Parrinello, M. (2012). Uncovering molecular details of urea crystal growth in the presence of additives. *J. Am. Chem. Soc.*, 134(41):17221–17233.

[Scherer et al., 2015] Scherer, M. K., Trendelkamp-Schroer, B., Paul, F., Pèrez-Hernández, G., Hoffmann, M., Plattner, N., Wehmeyer, C., Prinz, J.-H., and Noé, F. (2015). PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *J. Chem. Theory Comput.*, 11:5525–5542.

[Schmid et al., 2011] Schmid, N., Eichenberger, A. P., Choutko, A., Riniker, S., Winger, M., Mark, A. E., and van Gunsteren, W. F. (2011). Definition and testing of the gromos force-field versions 54a7 and 54b7. *Eur. Biophys. J.*, 40(7):843.

[Shell, 2008] Shell, M. S. (2008). The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.*, 129(14):144108.

[Shell, 2015] Shell, M. S. (2015). Coarse-graining with the relative entropy. *Adv. Chem. Phys.*, pages 395–441.

[Shen et al., 2013] Shen, J.-W., Li, C., van der Vegt, N. F. A., and Peter, C. (2013). Understanding the control of mineralization by polyelectrolyte additives:

Simulation of preferential binding to calcite surfaces. *J. Phys. Chem. C*, 117(13):6904–6913.

[Shirts and Chodera, 2008] Shirts, M. R. and Chodera, J. D. (2008). Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.*, 129(12).

[Sittel and Stock, 2018] Sittel, F. and Stock, G. (2018). Perspective: Identification of collective variables and metastable states of protein dynamics. *J. Chem. Phys.*, 149(15):150901.

[Souaille and Roux, 2001] Souaille, M. and Roux, B. (2001). Extension to the weighted histogram analysis method: combining umbrella sampling with free energy calculations. *Comput. Phys. Commun.*, 135:40–57.

[Stack et al., 2012] Stack, A. G., Raiteri, P., and Gale, J. D. (2012). Accurate rates of the complex mechanisms for growth and dissolution of minerals using a combination of rare-event theories. *J. Am. Chem. Soc.*, 134(1):11–14.

[Sugita and Okamoto, 1999] Sugita, Y. and Okamoto, Y. (1999). Replica-exchange molecular dynamics method for protein folding. *Chem. Phys. Lett.*, 314(1–2):141 – 151.

[Sultan et al., 2018] Sultan, M. M., Wayment-Steele, H. K., and Pande, V. S. (2018). Transferable neural networks for enhanced sampling of protein dynamics. *J. Chem. Theory Comput.*, 14(4):1887–1894.

[Tan et al., 2012] Tan, Z., Gallicchio, E., Lapelosa, M., and Levy, R. M. (2012). Theory of binless multi-state free energy estimation with applications to protein-ligand binding. *J. Chem. Phys.*, 136(14):144102.

[Theano Development Team, 2016] Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

[Torrie and Valleau, 1977] Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.*, 23(2):187–199.

[Tozzini, 2010] Tozzini, V. (2010). Minimalist models for proteins: a comparative analysis. *Q. Rev. Biophys.*, 43(3):333–371.

[Tribello et al., 2014] Tribello, G. A., Bonomi, M., Branduardi, D., Camilloni, C., and Bussi, G. (2014). PLUMED 2: New feathers for an old bird. *Comput. Phys. Commun.*, 185(2):604–613.

[Tribello et al., 2010] Tribello, G. A., Ceriotti, M., and Parrinello, M. (2010). A self-learning algorithm for biased molecular dynamics. *Proc. Natl. Acad. Sci. U. S. A.*, 107(41):17509–17514.

[Tribello et al., 2012] Tribello, G. A., Ceriotti, M., and Parrinello, M. (2012). Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proc. Natl. Acad. Sci. U. S. A.*

[Tribello and Gasparotto, 2019] Tribello, G. A. and Gasparotto, P. (2019). Using dimensionality reduction to analyze protein trajectories. *Front. Mol. Biosci.*, 6:46.

[van der Spoel et al., 2005] van der Spoel, D., Lindahl, E., Hess, B., Groenhof, G., Mark, A. E., and Berendsen, H. J. C. (2005). Gromacs: Fast, flexible, and free. *J. Comput. Chem.*, 26(16):1701–1718.

[van Gunsteren et al., 2002] van Gunsteren, W., Daura, X., and Mark, A. (2002). Computation of free energy. *Helv. Chim. Acta*, 85(10):3113–3129.

[Wagner et al., 2016] Wagner, J. W., Dama, J. F., Durumeric, A. E., and Voth, G. A. (2016). On the representability problem and the physical meaning of coarse-grained models. *J. Chem. Phys.*, 145(4):044108.

[Wallace et al., 2013] Wallace, A. F., Hedges, L. O., Fernandez-Martinez, A., Raiteri, P., Gale, J. D., Waychunas, G. A., Whitelam, S., Banfield, J. F., and De Yoreo, J. J. (2013). Microscopic evidence for liquid-liquid separation in supersaturated caco3 solutions. *Science*, 341(6148):885–889.

[Wang et al., 2019] Wang, J., Olsson, S., Wehmeyer, C., Pérez, A., Charron, N. E., De Fabritiis, G., Noé, F., and Clementi, C. (2019). Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.*

[Wang et al., 2011] Wang, L., Friesner, R. A., and Berne, B. J. (2011). Replica exchange with solute scaling: A more efficient version of replica exchange with solute tempering (rest2). *J. Phys. Chem. B*, 115(30):9431–9438.

[Wehmeyer and Noé, 2018] Wehmeyer, C. and Noé, F. (2018). Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.*, 148(24):241703.

[Weiner and Addadi, 1997] Weiner, S. and Addadi, L. (1997). Design strategies in mineralized biological materials. *J. Mater. Chem.*, 7:689–702.

[Wolf et al., 2015] Wolf, S. L., Jähme, K., and Gebauer, D. (2015). Synergy of mg 2+ and poly (aspartic acid) in additive-controlled calcium carbonate precipitation. *CrystEngComm*, 17(36):6857–6862.

[Wu et al., 2014] Wu, H., Mey, A. S. J. S., Rosta, E., and Noé, F. (2014). Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. *J. Chem. Phys.*, 141(21):214106.

[Wu et al., 2016] Wu, H., Paul, F., Wehmeyer, C., and Noé, F. (2016). Multiensemble markov models of molecular thermodynamics and kinetics. *Proc. Natl. Acad. Sci. U. S. A.*, 113(23):E3221–E3230.

[Wu and Zhang, 2008] Wu, S. and Zhang, Y. (2008). A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics*, 24(7):924–931.

[Yang et al., 2010] Yang, Y., Cui, Q., and Sahai, N. (2010). How does bone sialoprotein promote the nucleation of hydroxyapatite? a molecular dynamics study using model peptides of different conformations. *Langmuir*, 26(12):9848–9859.

[Yang et al., 2011] Yang, Y., Faraggi, E., Zhao, H., and Zhou, Y. (2011). Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates. *Bioinformatics*, 27(15):2076–2082.

[Zhang et al., 2018] Zhang, L., Han, J., Wang, H., Car, R., and E, W. (2018). Deepcg: Constructing coarse-grained models via deep neural networks. *J. Chem Phys.*, 149(3):034101.

[Zou et al., 2017] Zou, Z., Bertinetti, L., Politi, Y., Fratzl, P., and Habraken, W. J. E. M. (2017). Control of polymorph selection in amorphous calcium carbonate crystallization by poly(aspartic acid): Two different mechanisms. *Small*, 13(21):1603100.

[Zuckerman, 2011] Zuckerman, D. M. (2011). Equilibrium sampling in biomolecular simulations. *Annu. Rev. Biophys.*, 40:41–62.