Bhowmik *et al. BMC Bioinformatics* 2018, **19**(Suppl 18):484
https://doi.org/10.1186/s12859-018-2507-5

BMC Bioinformatics

RESEARCH Open Access

Deep clustering of protein folding simulations

Debsindhu Bhowmik, Shang Gao, Michael T. Young and Arvind Ramanathan*

From Computational Approaches for Cancer at SC17
Denver, CO, USA. 17 November 2017

Abstract

Background: We examine the problem of clustering biomolecular simulations using deep learning techniques. Since biomolecular simulation datasets are inherently high dimensional, it is often necessary to build low dimensional representations that can be used to extract quantitative insights into the atomistic mechanisms that underlie complex biological processes.

Results: We use a convolutional variational autoencoder (CVAE) to learn low dimensional, biophysically relevant latent features from long time-scale protein folding simulations in an unsupervised manner. We demonstrate our approach on three model protein folding systems, namely Fs-peptide (14 μs aggregate sampling), villin head piece (single trajectory of 125 μs) and β-β-α (BBA) protein (223 + 102 μs sampling across two independent trajectories). In these systems, we show that the CVAE latent features learned correspond to distinct conformational substates along the protein folding pathways. The CVAE model predicts, on average, nearly 89% of all contacts within the folding trajectories correctly, while being able to extract folded, unfolded and potentially misfolded states in an unsupervised manner. Further, the CVAE model can be used to learn latent features of protein folding that can be applied to other independent trajectories, making it particularly attractive for identifying intrinsic features that correspond to conformational substates that share similar structural features.

Conclusions: Together, we show that the CVAE model can quantitatively describe complex biophysical processes such as protein folding.

Keywords: Deep learning, Variational autoencoder, Protein folding, Conformational substates

Introduction

The phenomenal growth of computing capabilities have accelerated our ability to precisely model and understand complex bio-molecular events at the atomistic scale [1–5]. Several recent studies have demonstrated how long timescale molecular dynamics (MD) simulations can provide detailed insights into events driving several complex biological phenomena such as protein folding, ligand binding, and membrane transport, often complementing experimental results. MD simulations are governed by a potential energy function that includes both bonded and non-bonded terms whose gradient defines a force-field applied to every atom in the bio-molecular system [6]. These simulations integrate Newton's laws of motion for every atom in the system using time-steps that typically are of the order of a femtosecond (10^{-15}s). Even small simulation systems can potentially consist of thousands of atoms; given that bio-molecular events of interest typically occur at micro- and milli-second timescales, the increase in the size and complexity of these simulations is quickly becoming a limiting factor for extracting quantitative insights that are also biologically meaningful [7].

To overcome this challenge, a number of machine learning (ML) techniques are being developed to extract quantitative, biophysically relevant information from MD simulations. In particular, machine learning tools are able to quantify statistical insights into the time-dependent

*Correspondence: ramanathana@ornl.gov
Computational Science and Engineering Division, Oak Ridge National Laboratory, One Bethel Valley Road, MS6085, Oak Ridge, TN, USA

structural changes a biomolecule undergoes in simulations, identify events that characterize large-scale conformational changes at multiple timescales, build low-dimensional representations of simulation data capturing biophysical/biochemical/biological information, use these low-dimensional representations to infer kinetically and energetically coherent conformational substates, and obtain quantitative comparisons with experiments [8].

Since the dimensionality of MD simulations is large ($3 \times N$, where N is the number of atoms, or $2 \times (\phi, \psi, \chi)$ dihedral angles in the system of interest), ML techniques have focused on building low-dimensional representations of MD simulations. These dimensionality reduction techniques have used linear (e.g., principal component analysis [9], anharmonic conformational analysis [10–12]), non-linear (e.g., isometric mapping/ isomap [13], diffusion maps [14]) or hybrid approaches (e.g., locally linear embedding [15]) to characterize the conformational landscape sampled within simulations. Traditional ML approaches for analyzing long time-scale simulations typically require well-designed and often hand-crafted features. This in turn requires extensive prior knowledge about the system that is being simulated (for e.g., biophysically relevant reaction coordinates such as contacts between a ligand and its receptor). Often, use of certain ML techniques artificially restrict the simulation data being examined (for e.g., isolating only a subset of atoms from the simulations), or be prohibitively expensive to pre-/post-process the data. Finally, many of these approaches require pairwise comparison of individual conformers within the simulation with a similarity/dissimilarity measure that may be expensive to compute.

Deep structured learning approaches, on the other hand, overcome these challenges by automatically learning lower-level representations (or features) from the input data and successively aggregating them such that they can be used in a variety of supervised, semisupervised and unsupervised machine learning tasks [16]. Deep learning techniques have proven useful for a variety of structural bioinformatics applications, including protein structure prediction from biological sequences, and virtual screening/drug discovery applications [17–19]. Doerr and colleagues evaluated a variety of dimensionality reduction techniques for MD simulations and demonstrated that a shallow auto encoder could be used to visualize folding events within protein folding trajectories [20]. More recently, Pande and colleagues demonstrated how a reduced dimensionality representation from simulations built using tICA could be propagated using a time-dependent variational auto-encoder [21].

In this paper, we develop a convolutional variational auto-encoder (CVAE) that can automatically reduce the high dimensionality of protein folding trajectories and cluster conformations from MD simulations into a small number of conformational states that share similar structural, and energetic characteristics. Using equilibrium folding simulations of Fs peptide, villin headpiece, and BBA, all model systems for protein folding, our CVAE discovers latent features that captures folding intermediates, including misfolded states that can be challenging to characterize. We further demonstrate that the learned latent features from the CVAE can be 'transferred' across simulations, making it relevant for succinctly summarizing large-scale simulations and compare behaviors across trajectories. Together, we show that deep learning techniques can be used for unsupervised learning of biophysically relevant latent features from long timescale MD simulations.

Methods

Datasets and pre-processing

Given that deep learning techniques require large training data, we used three available datasets to demonstrate our approach. The first dataset consists of 28 separate MD simulations of the Fs peptide (Ace-$A_5(AAARA)_3$A-NME; 21 residues), a model system for protein folding, resulting in an aggregate sampling of 14 μs, consisting of 280,000 conformations. All simulations were performed at 300K using implicit solvent GBSA-OBC potentials and the AMBER-FF99SB-ILDN force field. This dataset was obtained from the MSMBuilder software [22]. The second dataset consists of (i) a single MD run of the Nle/Nle double variant of the C-terminal fragment of the villin head piece (referred to as VHP in this paper; Protein data bank (PDB) ID 2F4K [23]) of 125 μs simulated at 360 K, and third dataset is (ii) two long MD runs of the mixed β-β-α fold, namely BBA (PDB ID: 1FME [24]; 28 residues) for about 223 μs and 102 μs at 325 K [25], using Anton, a special purpose supercomputer for MD simulations [26] and the CHARMM22* force field and a modified TIP3P water model compatible with CHARMM force field [27].

We processed each trajectory using the `MDAnalysis` library [28, 29] to extract contact matrices between every pair of C^{α} atoms; we consider an atom to be in contact to another atom if it is separated by less than an 8 Å. Note that contact matrix representation is independent of rotation/translation (which is typically an artifact of MD simulations).

Convolutional variational autoencoder (CVAE)

Autoencoders (AEs) are a deep learning architecture designed to capture key representational information for a dataset within a low-dimensional latent space in an unsupervised fashion [30]. Autoencoders typically have an hourglass shaped architecture in which data is compressed into a low-dimensional latent space in the early layers and then reconstructed back in later layers.

Therefore, the latent space learns to capture the most essential information required for reconstruction.

In variational autoencoders (VAEs), an additional optimization constraint is that we require the latent space to be normally distributed [31]. While regular autoencoders may effectively capture important information in a reduced dimension, the latent embeddings may be sparsely distributed; this typically means that key information is spread across several clusters in the latent space, and the empty space between clusters does not capture any useful information–sampling from this empty space typically creates nonsensical results. By forcing the latent space to be normally distributed, we force the network to fully utilize the latent space so that information is distributed more evenly; this allows us to sample from any point in the latent space to generate new results that reflect the patterns in the original dataset. We therefore chose to use a VAE instead of a regular AE, as one of our long-term goals is to generate new potential structures based on the information learned within our model.

Rather than using regular feedforward layers in our VAE, we apply convolutional layers because they utilize sliding filter maps that can better recognize local patterns independent of its position in the data. In contact map representations, the state of the protein depends on the local interactions between a few atoms rather than on the global position of all atoms in the protein. Because these local interactions do not always appear in the exact same place in the protein, convolutional layers are better suited to recognize these local patterns independent of their position compared to feedforward networks. The architecture for the convolutional autoencoder (CVAE) used in our experiments is illustrated in Fig. 1.

Each CVAE was trained for a fixed number of epochs that was determined by the convergence of loss and variance-bias trade-off. The batch size was selected to be relatively small (length of the training data/100) to ensure reduced data in latent space do not collapse. We divided each dataset into training/testing/validation (80%/10%/10% of the simulation trajectories). Although not a requirement for unsupervised learning techniques, we used the validation data to characterize both the clustering and reconstruction quality of the CVAE. For example for the BBA system [32], we used a total of 1.1 million conformations of which 0.88 million conformers were used for training, with the remaining 0.22 million conformers equally split for testing and validation of the CVAE on unseen data from the same trajectory. The second trajectory from the BBA simulations was used only for testing the CVAE based on the training from the first trajectory. The various hyperparameter settings are shown in Table 1.

Results

We posited that the CVAE (described in "Methods" section) encoding would result in a model that can automatically capture biophysically relevant features from the simulation datasets. We used three model protein folding systems, namely Fs-peptide, villin headpiece (VHP) and BBA to demonstrate that the CVAE can learn a biophysically relevant latent space that corresponds to folding reaction coordinates, including fraction of native contacts and root mean squared deviation (RMSD) to the native state. To calculate the fraction of native contacts we use a definition similar to Savol and Chennubhotla [33]. Native contacts are based on a distance cut off of 8 Å or less between between C^{α} atoms and at least 75% of conformations remain within an RMSD cut-off of 1.1 Å of the native structure. First, we evaluate the ability of CVAE to learn a reduced dimensional space given the MD simulation data. Second, we show that the CVAE latent space corresponds to biophysically relevant features for each of the folding simulations studied. Finally, we demonstrate that the CVAE latent features can be transferred to other simulations, making it generalizable to a particular protein type.

Reconstruction quality of CVAE on protein folding trajectories

In order to evaluate the CVAE reconstruction quality from the protein folding trajectories, we first examined the overall loss ($\mathcal{L}$) of the CVAE over the training epochs (Eq. 1).

$$\mathcal{L} = \mathcal{E}_r + \mathcal{E}_l \tag{1}$$

$$\mathcal{E}_r = -\frac{1}{n}\sum_{i=1}^{n} X_i \log(f(z_i)) \tag{2}$$

$$\mathcal{E}_l = \mathrm{KL}(z||Normal(0,1)) \tag{3}$$

$\mathcal{L}$ is characterized as the sum of the reconstruction loss $\mathcal{E}_r$ (Eq. 2) and the latent loss $\mathcal{E}_l$ (Eq. 3). The reconstruction loss measures how well the CVAE can reconstruct the original input contact matrices (consisting of n conformations from the trajectory) and is calculated as the cross entropy loss between $f(z)$, which indicates the reconstructed probability of contact between two C^{α} atoms, and the original X contact maps from the simulation, which indicate the existence of contact between two C^{α} atoms. The latent loss is a regularizing constraint that forces the latent embeddings z to conform to a Gaussian distribution; this is calculated as the Kullback-Leibler (KL) divergence between the latent embeddings z and a Normal distribution with mean 0 and standard deviation 1.

For the three protein folding trajectories in this study (Fig. 2) a–c, we observed that the overall loss, $\mathcal{L}$, stabilizes over the training epochs, showing that it converges over time. We observed that for each protein, the number of

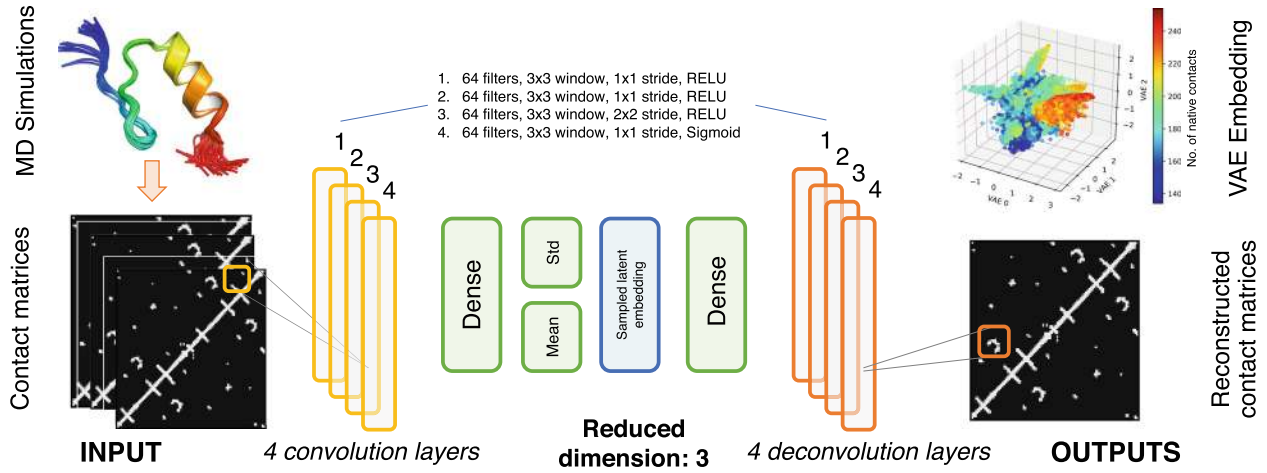

Fig. 1 Convolutional variational autoencoder architecture. The deep learning network processes MD simulation data into contact maps (2D images) that are then successively fed into 4 convolutional layers. The outputs from the final convolutional layer is then fed into a fully connected (dense) layer. This is then used to build the latent space in three dimensions, the output of which is the learned VAE embedding. In order to reconstruct the contact maps, we then use 4 successive de-convolutional layers, symmetric to the 4 input convolutional layers

training epochs needed to reach convergence is different; this is not surprising, given that the size of these proteins are different and the trajectories have unique folding pathways. Furthermore, we observed that the reconstruction loss (described in Eq. 1) is also different for each protein system – indicating that the quality of CVAE reconstruction is unique to each protein system.

We next examined whether the CVAE latent features can faithfully reconstruct the original data. To evaluate this, we used the reconstruction difference that simply measures the difference between the reconstructed and original contact matrices. Note that while the original contact matrix is typically binary (indicating presence or absence of a contact), the output from the CVAE is a value between 0 and 1, which may be interpreted as a likelihood of observing a contact between two C^{α} atoms. We plotted a histogram of reconstruction from the testing data, shown in Fig. 2d. The reconstruction difference varies between -1 and +1, which indicates whether the reconstructed data mispredicts the presence or absence of a contact respectively. We choose a nominal threshold of 10% of the original value to indicate misprediction. For the Fs-peptide simulations, the CVAE is able to faithfully reconstruct nearly 88% of all the observed contacts and mispredicts only 12% of contacts (Fig. 2d). We note that these contacts are at the interfaces of secondary structural elements, between α-helices, or between α-helices and β-strands. We can make similar observations for the other protein systems; the average reconstruction error for VHP is about 10.6% (Fig. 2e). For BBA (Fig. 2f), the CVAE reconstruction can recover nearly 88.5% of all contacts correctly in the folding simulations.

Table 1 Hyperparameter settings used for CVAE training

Hyperparameter	Lower bound	Upper bound	Optimal Fs-Peptide	Optimal BBA	Optimal VHP
Num convolutional layers	1	4	4	4	4
Num convolutional filters	16	125	100	100	99
Convolutional filter shape ($m \times m$)	2	7	5	5	5
Num dense neurons	32	100	64	64	58
Latent dimension	2	16	10	10	9
Mean reconstruction error			7.82	24.31	74.66
Mean reconstruction error (latent dim 3)			23.24	53.48	113.07

We evaluated the performance of CVAE as a function of several model hyperparameters using Bayesian optimization [34–36]. The search bounds and optimal results for the hyperparameters are summarized in Table 1. While the optimal settings for the latent dimension for each molecule was found to be near ten, we chose to use models with latent dimensions of size three. Since it is possible to verify visually that the autoencoder is meaningfully capturing the folding process without sacrificing much in terms of reconstruction error, we used a three dimensional latent space for each of the protein systems. To meaningfully visualize the CVAE latent representation, we chose the t-distributed stochastic neighborhood embedding (t-SNE) [37] method. There are many choices for visualizing the latent space, including techniques such as mixture of Gaussians, k-means clustering – however,

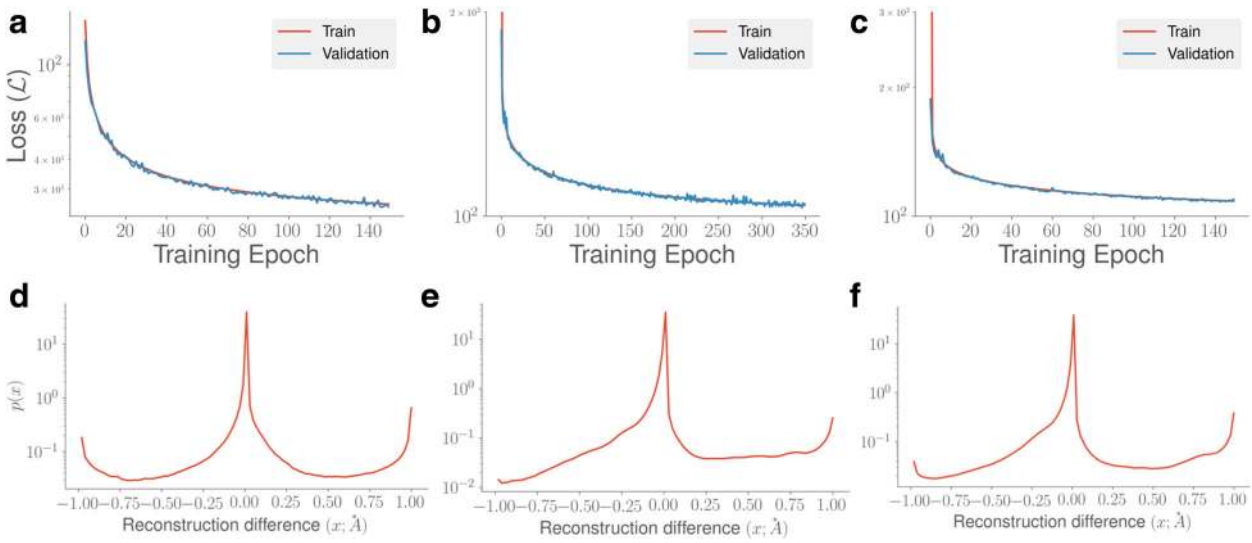

Fig. 2 Quantifying CVAE performance over MD simulations. We quantify the loss, $\mathcal{L}$, described in the "Results" section over the number of training epochs for each of the systems, namely **a** Fs-peptide, **b** VHP, and **c** BBA. As the number of epochs increases, the loss function decreases, indicating convergence in the training process. We also quantify the quality of reconstruction based on the CVAE's ability to predict contacts within the testing dataset. For each of the simulations, namely **d** Fs-peptide, **e** VHP, and **f** BBA, we show the reconstruction difference between the actual and predicted contact matrices from the testing data and histogram it to quantify how many contacts may be mispredicted by the model. See text for additional details

for this paper, t-SNE provided a practical way to visualize the CVAE latent space in a meaningful manner.

The mean reconstruction loss over various settings for the latent dimension for each of the protein folding trajectories can be seen in Fig. 3. When considering the latent dimension, there exists a trade-off between the model's ability to compress information and its ability to minimize reconstruction error. For example, we note that the choice of the optimization technique used for the training process affects the model performance. To illustrate this, we examined four different optimizers: namely, RMSProp, ADAM, ADAMax, and ADAgrad (Fig. 4). For each of these techniques, we tracked the reconstruction loss ($\mathcal{L}$) with both the training and testing data. As illustrated in Fig. 4, we found that the RMSprop optimizer (black line) performs the best compared to the other three optimizers with respect to the testing data. Further, we find that the model's performance can be affected by the interactions between the choice of latent dimension and other model hyperparameters.

CVAE reveals folding intermediates of Fs-peptide

Fs-peptide is often used as a model system to study protein folding processes; here the final state of the peptide

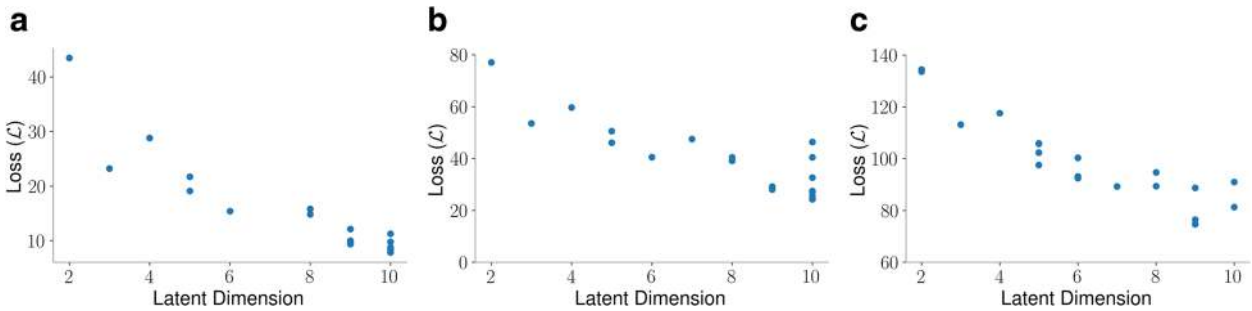

Fig. 3 Mean reconstruction error over various latent dimensions indicates as the latent dimension increases, the mean reconstruction error decreases. This shows a trade-off between compression and model accuracy. **a** Fs-Peptide exhibits a linear decrease in reconstruction error as the latent dimension increases. **b** VHP begins to show the limits of increasing the model's latent dimension; the lowest recorded mean reconstruction loss was found to be at latent dimension nine. **c** While the lowest reconstruction error for BBA was found to be with a latent dimension of size ten, there is a wide amount of variation in the reconstruction loss for this latent dimension size. Various values of the reconstruction error at each latent dimension shows there to be interactions between the various model hyperparameters

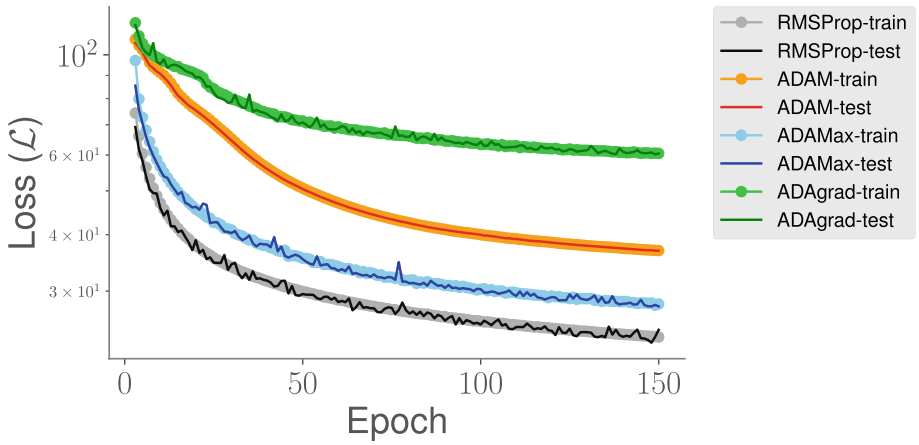

Fig. 4 RMSprop optimizer provides the best reconstruction of the CVAE latent space for Fs-peptide simulations. We used four popular optimizers, including ADAM, ADAMax, ADAgrad and RMSProp for understanding how well we can reconstruct the latent space representation of the Fs-peptide simulations. For each optimizer, we show the reconstruction error, defined as the loss, $\mathcal{L}$, for both the training/testing data over the course of 150 epochs. Notably, RMSProp provides the best reconstruction (lower is better) as indicated in the plot

is characterized by a fully α-helical state. In this paper, we examined whether our CVAE can recapitulate the diverse α-helical intermediate states in an unsupervised manner. Figure 5a shows the latent space learned by the CVAE. Each conformation from the training data is represented as a three-dimensional coordinate in the latent space. To understand whether the latent space captured by the CVAE describes the folding process, we colored each conformer with the corresponding RMSD to the native state. The RMSD to the native state is often used as a conformational coordinate to track protein folding trajectories [38]. We note that the input to the CVAE is only the raw contact maps; however, the model is able to distinguish between low and high RMSD conformers when projected onto the latent space.

Within the latent space, we note the presence of distinct pockets with high RMSD values to the native state (red colors), which converge eventually into folded state (blue colors). The gradation of the colors along the arms of the CVAE axes indicates that the latent space ($z_1 - z_3$) is able to describe the folding process. It indicates multiple pathways along which Fs-peptide folds into its final state. Although the CVAE-determined latent space can capture the presence of both folded/unfolded states (quantified by the total number of contacts), it is still challenging to interpret. Hence, we used t-SNE to visualize the results. We painted each conformation in the t-SNE with the RMSD values (Fig. 5b) and the fraction of native contacts (Fig. 5c). The t-SNE approach allows us to identify distinct conformational clusters observed from the simulations, labeled (i) to (viii), in the folding trajectories. In particular, we find the presence of partially folded α-helical bundles as well as a fully formed α-helix, which represents the folded state of the protein. Additionally, we also find that the different folded states are separated and connected via multiple intermediate states, all of which have relatively lower number of total contacts. This indicates that for the transitions between the folded microstates, the peptide must undergo several unfolding events.

Interestingly, our approach also reveals the presence of potentially misfolded states in these trajectories. In this work, we consider a misfolded state to be a set of conformations that share higher fraction of native contacts, but have a high RMSD from the native state ensemble of the protein. For example, state (viii) in Fig. 5c shows the presence of conformations that have higher fraction of native contacts (close to 0.95), however, its secondary structure content is significantly different from the native state, highlighted as (vii) in Fig. 5c. The intermediate states identified here have differences in their secondary structural content, i.e., the number of α-helical turns as depicted in states labeled (i), (ii), (iv), (vi) and (viii) along with differences in the extent to which the N- and C-terminal ends of the protein are folded (for e.g., state labeled (ii) folds from the N-terminal end versus state labeled (iii) folds from the C-terminal end).

We can also visualize the tSNE dimensions as the logarithm of the histograms as a simple estimate of the free energy surface as depicted in Fig. 5d, where by conformational states can be visualized. This representation is only for visual purposes and as such can be used for qualitative insights into the organization of the folding energy landscape of Fs-peptide. The native state of the protein, labeled (vii), consists of the fully folded peptide, while many of the partially folded states and their intermediates are distributed around the periphery of this landscape. It is interesting to note that the contours represent conformational states that correspond to

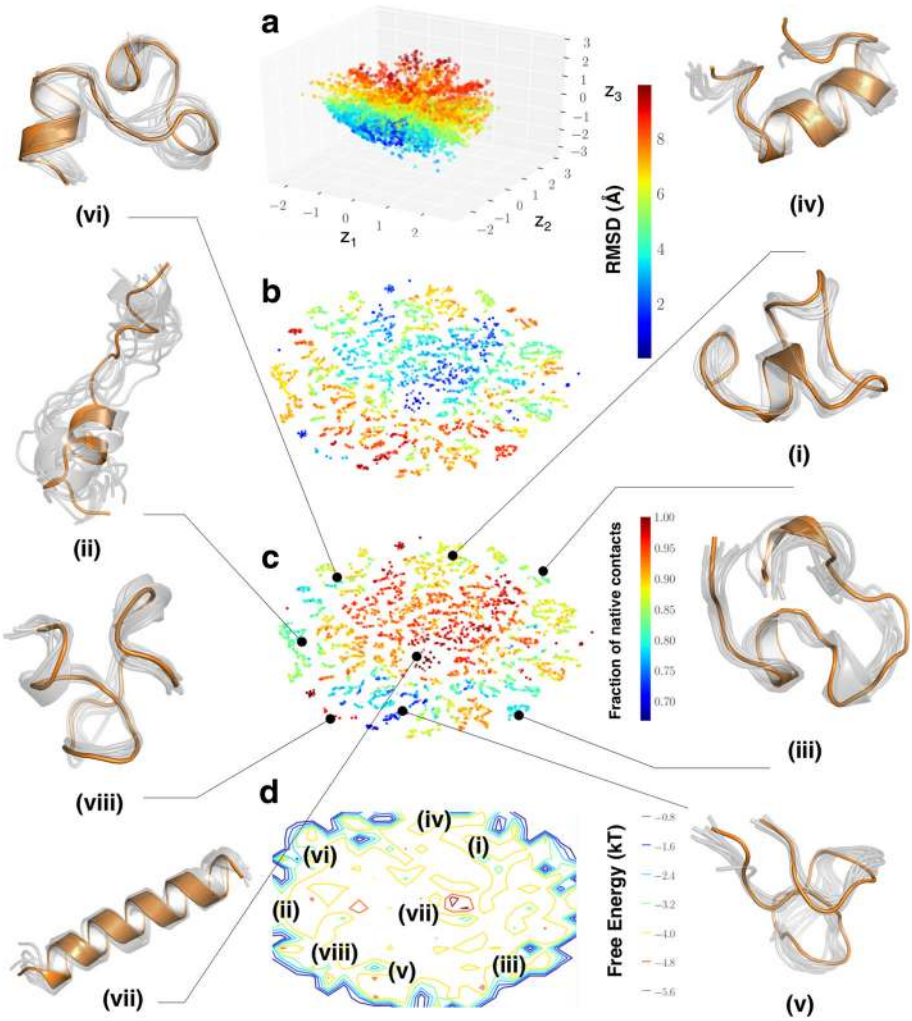

Fig. 5 CVAE can identify intermediate, partially folded states from Fs-peptide simulations. **a** We project the conformations from the simulations onto a latent space spanned by the three dimensions of the CVAE ($z_1 - z_3$). Each conformation is then painted with the corresponding RMSD to the native state. As observed from the figure, there is a gradation from unfolded states (i.e., higher RMSD values) towards more folded states (lower RMSD values). To allow easy interpretation, we use t-SNE to project the conformations. **b** t-SNE projection painted with RMSD values and **c** fraction of native contacts shows clear separation between folded conformations, labeled (vii) along with partially folded states, namely (ii), (iii), (iv), and (vi), along with unfolded states, labeled (v). It is notable that state (viii) has lower RMSD to the folded state (vii), where as shows very few of the native contacts that are found in the folded state, implying that this state may be a misfolded state in the ensemble simulations. **d** A simple histogram representation of the projected t-SNE space allows us to characterize the conformational states observed within the simulations. The labels provide a relative description of the location of the various conformational states

folding coordinates and each of the states are marked (using solid lines) as to where they belong on this landscape.

CVAE reveals conformational states in the VHP folding pathway

For the VHP simulations, we were able to identify a similar distribution of folded/unfolded conformations along its folding pathway (Fig. 6). Even though the reconstruction error plots from Fig. 3b indicate that the ideal number of latent dimensions is 9, we examined whether a low dimensional encoding with just three dimensions is able to capture folding events within this trajectory. Similar to the analysis of the Fs-peptide folding trajectories, the latent embedding of the CVAE reveals the presence of folded and unfolded conformations that are separated by a large number of intermediate states (Fig. 6a). Since these simulations were carried out at a higher temperature (360 K), these simulations indicate larger fluctuations in the secondary structures of VHP. Further,

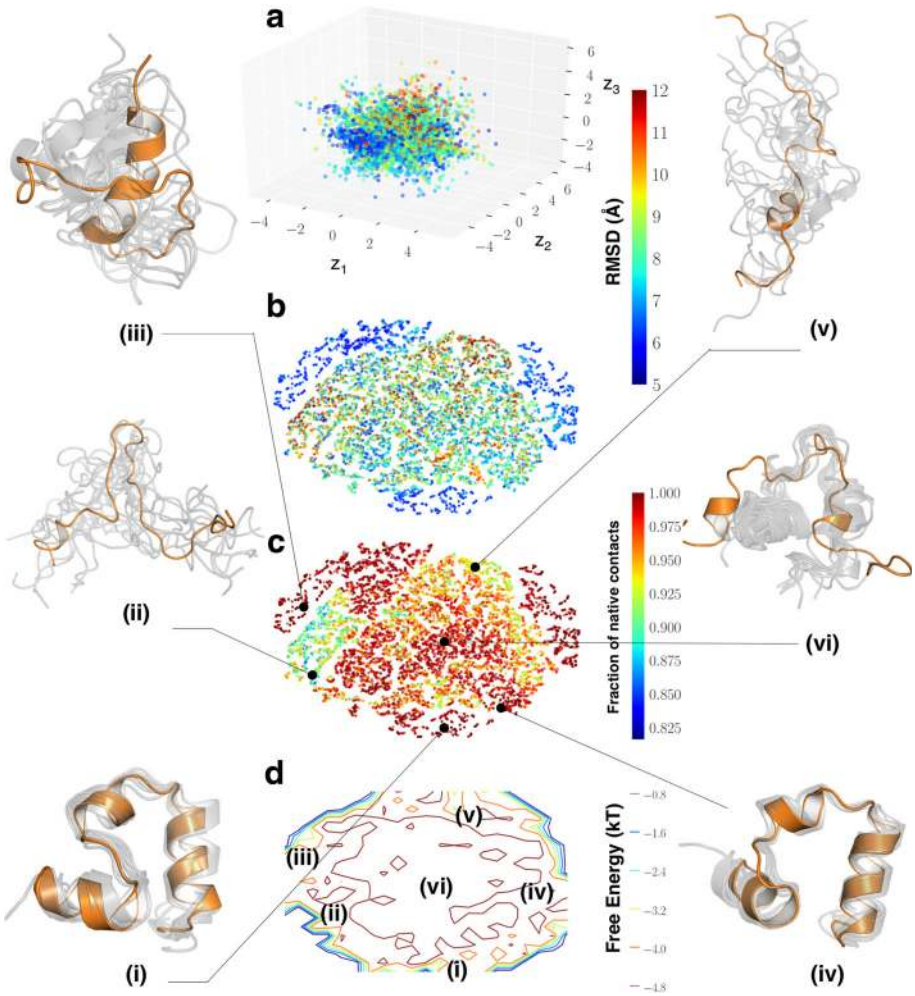

Fig. 6 CVAE identifies partially folded states within the VHP simulations. **a** For the VHP simulations, we project the conformations onto the CVAE space spanned by the three latent dimensions ($z_1 - z_3$). Each conformation is painted by the RMSD to the native state with the color scale depicting the range of RMSD values between the folded and unfolded states. For the VHP simulations we consider any state with an RMSD of less than 5 Å as essentially folded (since these simulations are carried out at a higher temperature). To enable a better characterization of the VHP folding landscape, we project the conformations using t-SNE to identify conformational substates. Each conformation is painted with **b** the RMSD to the native state, and **c** the fraction of native contacts. The separation between the folded, unfolded and intermediate states can be understood by examining the conformational states labeled (i) - (v). **d** A simple histogram representation from the projected t-SNE space allows us to characterize the locations of the conformational states observed from this simulation. The labels provide a relative location of the various conformational states

within the course of the simulations, a total of 34 folding events are summarized, which indicate a large number of conformational states actually correspond to folded conformations.

To enable interpretation of the VHP folding landscape, we projected the CVAE latent dimensions using t-SNE and observed that the folded states of VHP are separated into three distinct 'wells' that correspond to the folding events along this trajectory. The evidence for the folding events emerges from painting the t-SNE landscape with the fraction of native contacts (Fig. 6c). A large portion of the trajectory is either unfolded (e.g., states labeled (ii), (v) indicated as conformational ensembles along the trajectory) or partially folded, i.e., showing the presence of all the three helices, but with different number of helical turns (e.g., states labeled (iii) and (vi) in Fig. 6). Finally, the folded states labelled (i) and (iv) capture distinct orientations of α-helices as observed from the figure. It is interesting to note that the transition from one folded state to the other involves partial unfolding (similar to Fs-peptide). Further, we also note that the partially folded state (iii) consists of many native contacts; however, this state does not have all the three helices and may represent an unfolded intermediate state through which the

transition to either of the two folded states may occur. The simple histogram representation of the t-SNE coordinates (Fig. 6d) provides an easy way to interpret the different conformational states with respect to the folded states in the trajectory (states (i) and (iv)).

CVAE analysis of BBA folding simulations can be transferred to learn folding patterns across trajectories

We next examined whether the CVAE learned features could be used to predict conformational states from a completely different trajectory. To facilitate this analysis, we used the BBA simulations (see "Methods" section) as a prototype example. Our experimental set up included training the CVAE on the first long time-scale trajectory (223 μs) and predicting if it captures the folding events from the second trajectory (102 μs). As depicted in Fig. 7a, the three latent CVAE dimensions capture the presence of multiple folded conformational states (labeled (i), (iii) and (iv) in Fig. 7b using t-SNE). These states are separated by an intermediate state labeled (ii) and an unfolded

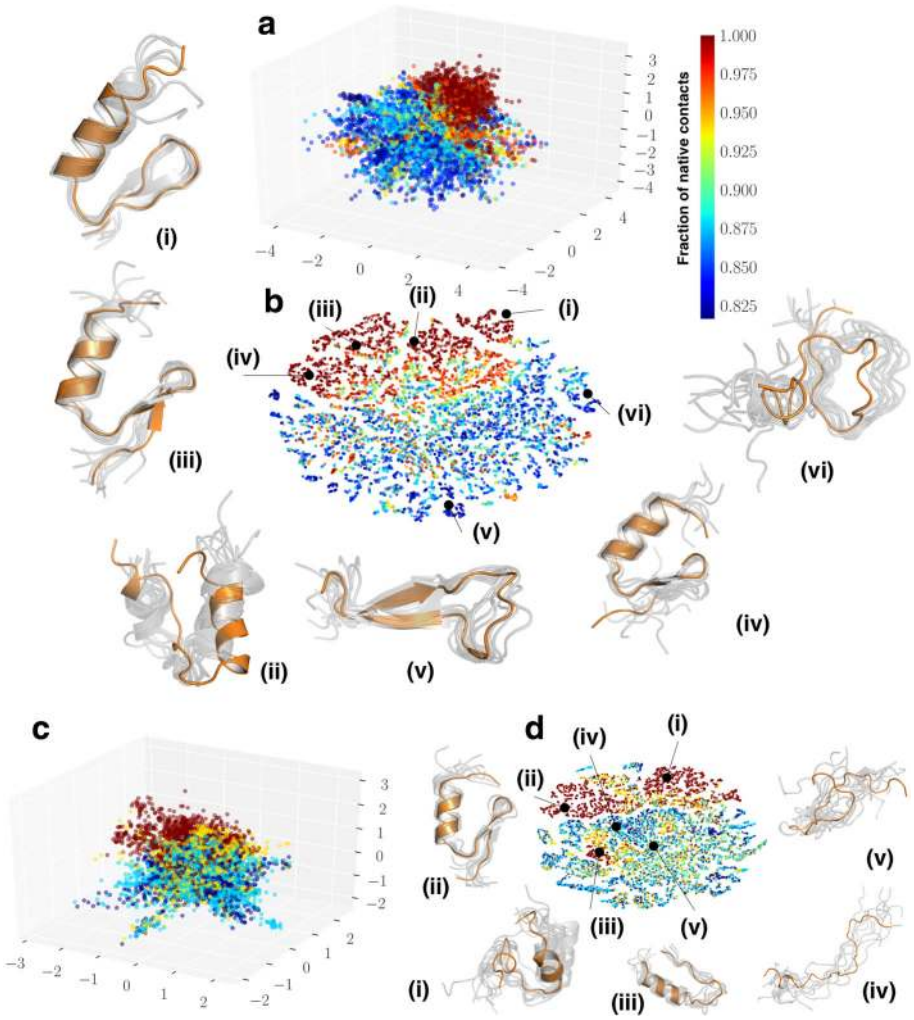

Fig. 7 CVAE learned features predict folding intermediates across two independent folding simulations of BBA protein. **a** depicts the CVAE latent space embedding on one 223 μs trajectory of BBA folding. Each conformation from the simulation is projected onto a three dimensional embedding and painted with the fraction of native contacts. One can observe that folded and unfolded states are separated into distinct clusters. **b** To elucidate the embedding from the CVAE, we project the conformations from the trajectory using t-SNE and identify conformational states captured by the CVAE. States are captured as described in Fig. 5 and shown as a cartoon representation based on the cluster from which they belong to. These conformational states depict various levels of BBA folding, labeled (i) through (vi). **c** Using the model learned on the longer trajectory, we project the conformers from the second independent simulation of length 102 μs onto the same latent space. It is notable that the folded states are similarly clustered together while the unfolded states are captured separately. **d** The latent representation from the CVAE can be applied across different trajectories to summarize folded states. As shown in the cartoon representation, the conformational ensembles are separated into distinct folded (labeled (ii)), partially folded (labeled (i) and (iii)) and fully unfolded states (labeled (iv) and (v))

state labeled (vi). Finally, it is interesting to note that latent space also characterizes a misfolded state, labeled (v), which shows the presence of an extended β-strand.

Using the same model that was trained on the first trajectory, we can project the conformations from the second, shorter simulation onto the latent space learned to test if the folded/unfolded states are separated. As shown in Figs. 7c and 7d, the latent space from the second trajectory clearly shows a separation between the folded states labeled (ii) in Fig. 7d, partially folded states labeled (i) and (iii) in Fig. 7d, and unfolded states labeled (iv) and (v) in Fig. 7d. We also observed that the latent space reconstruction difference is on par with the original model, implying that the features learned by the CVAE can indeed be transferred.

Discussion and conclusions

We have demonstrated how deep learning algorithms can be used to analyze and interpret protein folding simulations. We designed a CVAE that can encode the inherent high dimensionality of the folding trajectories into a low dimensional embedding that is biophysically relevant. We demonstrated our approach on three prototypical systems, namely Fs-peptide, VHP and BBA, all of which have been extensively characterized in previous studies. In all the cases, we note that the learned CVAE embeddings captured the distinction between potentially folded, partially folded, and misfolded states.

We used contact matrices determined form the simulations as inputs to the CVAE. Contact matrices are a practical approach to represent simulation datasets, which have been widely used to characterize protein folding pathways [39, 40]. However, the resolution of information captured using contact maps is fairly low and may not be specific. Although the CVAE identified the presence of folded/unfolded and misfolded states in the simulations, there is significant scope for directly using coordinate information (or other physical quantities such as dihedral angles) from simulations for characterizing these pathways. However, using coordinate information requires alignment of the trajectories to a reference conformation, which can be often challenging when the simulations are running. This is not true with internal coordinate representations such as dihedral angles, and we will use these techniques in the near future.

Complementary to the approaches taken by Doerr and colleagues [20], we build an autoencoder; however augmenting it with a variational formulation allows us to obtain interpretable features from the latent space. As demonstrated in the three systems, the CVAE latent spaces capture a succinct model of protein folding with the ability to distinguish conformational substates that share similar structural features. We have yet to evaluate whether these substates share similar energetic profiles. Further, our CVAE can be used to potentially augment propagators in time [21] such that temporal correlations are captured within these trajectories.

The selection of the hyperparameters, such as the size/stride of convolutional filters and the dimensions of the latent space to embed the simulations were based on empirical evaluations. Ideally, the choice of the latent space representation should be a parameter that can be learned from the simulation data itself (instead of being specified by the user). Further, these latent dimensions should correspond to directions in the landscape that enable the bio-molecular system to sample folded/misfolded states, which has been previously demonstrated by pursuing higher order statistical dependencies in atomic fluctuations in the simulations [7, 10]. We plan to extend our CVAE to automatically learn and infer this latent dimensional space.

Further studies are essential in associating the biophysical relevance of the learned CVAE embeddings. Specifically, we have not evaluated whether the CVAE embeddings for these folding trajectories correspond to biophysical reaction coordinates, i.e., whether the unique directions proposed by the CVAE can 'fold' a protein system. Temporal correlations are known to significantly influence bio-molecular events [41]. Although we trained our model to include temporal information (i.e., frames for the training was based on successive conformations in the trajectory), the embeddings learned do not necessarily correspond to detectable bio-molecular events. For e.g., in a protein folding trajectory, a typical event corresponds to 'whether a β-strand was formed' – our CVAE is currently unable to identify timepoints where significant structural or dynamical changes have occurred within trajectories. Leveraging our previous experience in developing techniques for event detection [12, 42], we will explore deep learning models for bio-molecular event detection in the near future.

Acknowledgements

The authors would like to thank D. E. Shaw Research for providing access to the protein folding simulation trajectories of BBA and VHP. The authors also thank the MSMBuilder team for making their Fs-Peptide simulations available online. This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This work has been supported in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health. This work was performed under the auspices of the U.S. Department of Energy by Argonne National Laboratory under Contract DE-AC02-06-CH11357, Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, Los Alamos National Laboratory under Contract DE-AC5206NA25396, Oak Ridge National Laboratory under Contract

DE-AC05-00OR22725, and Frederick National Laboratory for Cancer Research under Contract HHSN261200800001E.
This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05- 00OR22725.

Funding
Publications costs were funded in part by the Joint Design of Advanced Computing Solutions for Cancer (JDACS4C) program established by the U.S. Department of Energy (DOE) and the National Cancer Institute (NCI) of the National Institutes of Health and the Laboratory Director's Research and Development Fund.

Availability of data and materials
The individual simulation datasets of protein folding trajectories can be requested from D.E. Shaw Research. The Fs-peptide simulations were downloaded using the MSMBuilder software.The code developed is available upon request from Arvind Ramanathan (ramanathana@ornl.gov).

About this supplement
This article has been published as part of *BMC Bioinformatics Volume 19 Supplement 18, 2018: Selected Articles from the Computational Approaches for Cancer at SC17 workshop*. The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-19-supplement-18.

Authors' contributions
AR and DB conceived and designed the study. DB and SG implemented the CVAE model. MTY contributed to developing tools for analyzing hyperparameters and their impact on CVAE performance. All authors contributed to the writing, editing and reviewing the article. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Authors' information
Computational Science and Engineering Division, Oak Ridge National Laboratory, MS6085, Oak Ridge, TN, 37830-6085, USA.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Published: 21 December 2018

References

1. Ingólfsson HI, Arnarez C, Periole X, Marrink SJ. Computational 'microscopy' of cellular membranes. J Cell Sci. 2016. https://doi.org/10.1242/jcs.176040.
2. Dror RO, Dirks RM, Grossman JP, Xu H, Shaw DE. Biomolecular simulation: a computational microscope for molecular biology. Annu Rev Biophys. 2012;41(1):429–52.
3. Lee EH, Hsin J, Sotomayor M, Comellas G, Schulten K. Discovery through the computational microscope. Structure. 2009;17(10):1295–306. https://doi.org/10.1016/j.str.2009.09.001.
4. Dhindsa GK, Bhowmik D, Goswami M, O'Neill H, Mamontov E, Sumpter BG, Hong L, Ganesh P, Chu X.-q. Enhanced dynamics of hydrated trna on nanodiamond surfaces: A combined neutron scattering and md simulation study. J Phys Chem B. 2016;120(38):10059–68. PMID: 27584158. https://doi.org/10.1021/acs.jpcb.6b07511.
5. Lynch VE, Borreguero JM, Bhowmik D, Ganesh P, Sumpter BG, Proffen TE, Goswami M. An automated analysis workflow for optimization of force-field parameters using neutron scattering data. J Comput Phys. 2017;340:128–37. https://doi.org/10.1016/j.jcp.2017.03.045.
6. Adcock SA, McCammon JA. Molecular dynamics: Survey of methods for simulating the activity of proteins. Chem Rev. 2006;106(5):1589–615.
7. Ramanathan A, Savol A, Burger V, Chennubhotla C, Agarwal PK. Protein Conformational Populations and Functionally Relevant Substates. Acc Chem Res. 2014;47(1):149–156. https://doi.org/10.1021/ar400084s.
8. Ramanathan A, Savol A, Burger V, Quinn S, Agarwal PK, Chennubhotla C. Statistical inference for big data problems in molecular biophysics. In: Neural Information Processing Systems: Workshop on Big Learning. 2012. https://www.osti.gov/biblio/1055187.
9. Maisuradze GG, Liwo A, Scheraga HA. Principal component analysis for protein folding dynamics. J Mol Biol. 2009;385(1):312–29. https://doi.org/10.1016/j.jmb.2008.10.018.
10. Ramanathan A, Savol AJ, Langmead CJ, Agarwal PK, Chennubhotla C. Discovering conformational sub-states relevant to protein function. PLoS ONE. 2011;6(1):15827.
11. Burger VM, Ramanathan A, Savol AJ, Stanley CB, Agarwal PK, Chennubhotla C. Quasi-anharmonic analysis reveals intermediate States in the nuclear co-activator receptor binding domain ensemble. Pac Symp Biocomput. 2012;70–81. https://www.worldscientific.com/doi/abs/10.1142/9789814366496_0008.
12. Ramanathan A, Savol AJ, Agarwal PK, Chennubhotla C. Event detection and sub-state discovery from biomolecular simulations using higher-order statistics: application to enzyme adenylate kinase. Proteins Struct Func Bioinform. 2012;80(11):2536–51.
13. Das P, Moll M, Stamati H, Kavraki LE, Clementi C. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. Proc Natl Acad Sci. 2006;103(26):9885–90. https://doi.org/10.1073/pnas.0603553103. http://www.pnas.org/content/103/26/9885.full.pdf.
14. Kim SB, Dsilva CJ, Kevrekidis IG, Debenedetti PG. Systematic characterization of protein folding pathways using diffusion maps: Application to trp-cage miniprotein. J Chem Phys. 2015;142(8):085101. https://doi.org/10.1063/1.4913322.
15. Duan M, Fan J, Li M, Han L, Huo S. Evaluation of dimensionality-reduction methods from peptide folding–unfolding simulations. J Chem Theory Comput. 2013;9(5):2490–7. https://doi.org/10.1021/ct400052y.
16. Goodfellow I, Bengio Y, Courville A. Deep Learning; 2016. http://www.deeplearningbook.org. Accessed 28 Nov 2018.
17. Gawehn E, Hiss JA, Schneider G. Deep learning in drug discovery. Mol Inform. 2016;35(1):3–14.
18. Ramsundar B, Kearnes S, Riley P, Webster D, Konerding D, Pande V. Massively multitask networks for drug discovery. 2015. arXiv preprint arXiv:1502.02072. https://arxiv.org/abs/1502.02072.
19. Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein-ligand binding affinity. 2017. arXiv preprint arXiv:1703.10603. https://arxiv.org/abs/1703.10603.
20. Doerr S, Ariz-Extreme I, Harvey MJ, De Fabritiis G. Dimensionality reduction methods for molecular simulations. 2017. ArXiv e-prints. https://arxiv.org/abs/1710.10629.
21. Hernández CX, Wayment-Steele HK, Sultan MM, Husic BE, Pande VS. Variational Encoding of Complex Dynamics. 2017. ArXiv e-prints. https://journals.aps.org/pre/abstract/10.1103/PhysRevE.97.062412.
22. Beauchamp KA, Bowman GR, Lane TJ, Maibaum L, Haque IS, Pande VS. Msmbuilder2: Modeling conformational dynamics on the picosecond to millisecond scale. J Chemical Theory Computat. 2011;7(10):3412–9.
23. Kubelka J, Chiu TK, Davies DR, Eaton WA, Hofrichter J. Sub-microsecond protein folding. J Mol Biol. 2006;359(3):546–53. https://doi.org/10.1016/j.jmb.2006.03.034.
24. Sarisky CA, Mayo SL. The bba-fold: explorations in sequence space. J Mol Biol. 2001;307(5):1411–8. https://doi.org/10.1006/jmbi.2000.4345.
25. Lindorff-Larsen K, Piana S, Dror RO, Shaw DE. How fast-folding proteins fold. Science. 2011;334(6055):517–20.
26. Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK, Young C, Batson B, Bowers KJ, Chao JC, Eastwood MP, Gagliardo J, Grossman JP, Ho CR, Ierardi DJ, Kolossváry I, Klepeis JL, Layman T, McLeavey C, Moraes MA, Mueller R, Priest EC, Shan Y, Spengler J, Theobald M, Towles B, Wang SC. Anton a special-purpose machine for molecular dynamics simulation. Commun ACM. 2008;51(7):91–7. https://doi.org/10.1145/1364782.1364802.
27. Lindorff-Larsen K, Maragakis P, Piana S, Eastwood MP, Dror RO, Shaw DE. Systematic validation of protein force fields against experimental data. PLOS ONE. 2012;7(2):32131.

28. Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O. Mdanalysis: A toolkit for the analysis of molecular dynamics simulations. J Comput Chem. 2011;32(10):2319–27. https://doi.org/10.1002/jcc.21787.
29. Gowers RJ, Linke M, Barnoud J, Reddy TJE, Melo MN, Seyler SL, Domanski J, Dotson DL, Buchoux S, Kenney IM, Beckstein O. MDAnalysis: A python package for the rapid analysis of molecular dynamics simulations. In: Benthall S, Rostrup S, editors. Proceedings of the 15th Python in Science Conference. 2016. p. 98–105. http://conference.scipy.org/proceedings/scipy2016/pdfs/oliver_beckstein.pdf.
30. Baldi P. Autoencoders, unsupervised learning, and deep architectures. In: Proceedings of ICML Workshop on Unsupervised and Transfer Learning. 2012. p. 37–49. http://proceedings.mlr.press/v27/baldi12a/baldi12a.pdf.
31. Doersch C. Tutorial on variational autoencoders. 2016. arXiv preprint arXiv:1606.05908.
32. Lindorff-Larsen K, Piana S, Palmo K, Maragakis P, Klepeis JL, Dror RO, Shaw DE. Improved side-chain torsion potentials for the amber ff99sb protein force field. Proteins Struct Func Bioinform. 2010;78(8):1950–58.
33. Savol AJ, Chennubhotla C. Quantifying the sources of kinetic frustration in folding simulations of small proteins. J Chem Theory Comput. 2014;10(8):2964–74. PMID: 25136267. https://doi.org/10.1021/ct500361w.
34. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim. 1998;13(4):455–92. https://doi.org/10.1023/A:1008306431147.
35. Bergstra JS, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. In: Advances in Neural Information Processing Systems. 2011. p. 2546–54. https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf.
36. Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2. NIPS'12. Curran Associates Inc.; 2012. p. 2951–2959. http://dl.acm.org/citation.cfm?id=2999325.2999464.
37. Maaten LVD, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9(Nov):2579–605.
38. Gsponer J, Caflisch A. Molecular dynamics simulations of protein folding from the transition state. Proc Natl Acad Sci. 2002;99(10):6719–24. https://doi.org/10.1073/pnas.092686399. http://www.pnas.org/content/99/10/6719.full.pdf.
39. Vendruscolo M, Kussell E, Domany E. Recovery of protein structure from contact maps. Fold Des. 1997;2(5):295–306.
40. Domany E. Protein folding in contact map space. Physica A Stat Mech App. 2000;288(1):1–9. Dynamics Days Asia-Pacific: First International Conference on NonLinear Science.
41. Bowman GR, Beauchamp KA, Boxer G, Pande VS. Progress and challenges in the automated construction of markov state models for full protein systems. J Chem Phys. 2009;131(12):124101. https://doi.org/10.1063/1.3216567.
42. Ramanathan A, Yoo JO, Langmead CJ. On-the-fly identification of conformational substates from molecular dynamics simulations. J Chem Theory Comput. 2011;7(3):778–89. PMID: 26596308.