Explore protein conformational space with variational autoencoder

Hao Tian,† Xi Jiang,‡ Francesco Trozzi,† Sian Xiao,† Eric C. Larson,¶ and Peng Tao*†

†Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas, USA
‡Department of Statistical Science, Southern Methodist University, Dallas, Texas, USA
¶Department of Computer Science, Southern Methodist University, Dallas, Texas, USA

E-mail: ptao@smu.edu

Abstract

Molecular dynamic (MD) simulations have been actively used in the study of protein structure and function. However, extensive sampling in the protein conformational space requires large computational resources and takes a prohibitive amount of time. In this study, we demonstrated that variational autoencoders (VAEs), a type of deep learning model, can be employed to explore the conformational space of a protein through MD simulations. VAEs are shown to be superior to autoencoders (AEs) through a benchmark study, with low deviation between the training and decoded conformations. Moreover, we show that the learned latent space in the VAE can be used to generate unsampled protein conformations. Additional simulations starting from these generated conformations accelerated the sampling process and explored hidden spaces in the conformational landscape.
1 Introduction

Molecular dynamics (MD) simulations have been applied extensively to understand protein structure, function and kinetics.\textsuperscript{1,2} Through the development of hardware and software, e.g., graphics processing unit (GPU)\textsuperscript{3} and OpenMM,\textsuperscript{4} the simulation time scale has climbed from nanoseconds to milliseconds. However, this time scale is still insufficient in the study of slow-motion molecular events such as large-scale conformational transitions.\textsuperscript{5} Moreover, the energy landscapes of proteins are discretized with many local energy minima separated by high energy barriers.\textsuperscript{6} This rough energy landscape limits the applications of MD simulations and hinders a complete sampling of protein movements.

In recent years, enhanced sampling methods have been developed to address this issue. One class of methods introduces biasing potentials, such as Gaussian-accelerated MD (GaMD),\textsuperscript{7} to expand the landscape. However, some domain knowledge is required to define the essential coordinates, e.g., collective variables (CVs).\textsuperscript{8} Another class iteratively conducts new simulations by selecting seed structures from less sampled regions. Those starting structures can be chosen from the results of Markov state models\textsuperscript{9} or dimensionality reduction methods.

The advancement of deep learning provides an alternative approach for protein sampling. Several studies have demonstrated the success of both autoencoders (AEs) and variational autoencoders (VAEs) in their applications to protein conformations and functions.\textsuperscript{10–14} These models are capable of learning a low-dimensional representation through the encoder model while predicting new protein conformations through the decoder model. Moreover, the learned latent space in one protein system is biologically meaningful and can be transferred to a similar system, with latent variables be treated as CVs.\textsuperscript{15}

In this study, we proved the success of variational autoencoders in protein sampling by using the enzyme adenosine kinase (ADK) as an example. The crystallized ADK is initially in its closed state and undergoes a series of conformational changes to its open state.\textsuperscript{16} MD simulations were conducted to sample this process and used for model training. A
benchmark study was conducted to compare the performance of both VAEs and AEs with regard to the encoder and decoder models. VAEs perform better than AEs and are selected for further analysis. Random points in the middle of the closed and open states in the latent space were selected and decoded into new protein conformations. Additional MD simulations starting from these predicted conformations, together with the training simulations, sampled a complete transition from the closed to the open states and explored hidden conformational spaces.

2 Methods

2.1 Molecular dynamics simulations

The initial structures of the closed and open states of ADK were taken from the Protein Data Bank (PDB)\textsuperscript{17} with the PDB IDs as 1ake and 4ake, respectively. Chain A was extracted with the removal of ligands and crystal waters as the starting structure in both states. The systems were added with hydrogen atoms and solvated in a periodic boundary box of TIP3P water molecules.\textsuperscript{18} Na\textsuperscript{+} and Cl\textsuperscript{-} ions were used to neutralize the system. Energy minimization was performed with the steep descent method for each system. 100 picoseconds of canonical ensemble (NVT) Langevin MD simulations were carried out, followed by 200 picoseconds of isothermal–isobaric ensemble (NPT) simulations at 1 atm and 300 K. Finally, the systems were switched back to NVT. Both the closed and open states conducted 5 nanoseconds simulations initially while the closed state simulations continued to 50 nanoseconds. Each MD simulation was repeated three times independently. The electrostatic interactions were calculated using the particle mesh Ewald (PME) algorithm.\textsuperscript{19} Bonds associated with hydrogen atoms were constrained using the SHAKE algorithm\textsuperscript{20} with 2 fs step size. All simulations were conducted with CHARMM27 force field\textsuperscript{21} and OpenMM 7.\textsuperscript{4}

Trajectories were aligned to the first frame and 1,660 heavy backbone atoms were selected. The Cartesian coordinates were extracted and further normalized as features using
Figure 1. Autoencoder architecture for ADK protein. The Cartesian coordinates from the closed and open states of ADK trajectories are extracted as inputs. The encoder module is designed with decreasing number of neurons in hidden layers to encode high-dimensional inputs to a low-dimensional latent space. The decoder module, with increasing number of neurons in hidden layers, aims to project latent space back to protein structures.

the MinMax scaling. Coordinate \( c, (c \in x, y, z) \) for atom \( i \) in structure \( k \) is normalized as:

\[
\text{normed} \ c_i^k = \frac{c_i^k - \min(c)}{\max(c) - \min(c)}
\]

(1)

2.2 Autoencoders and variational autoencoders

Autoencoders are a type of unsupervised deep learning models that are designed to encode an input to a low-dimensional latent space and decode it back.\(^{22}\) For this purpose, autoencoders normally have a hourglass shaped architecture, as shown in Figure 1. The first part of the hourglass is an encoder module for compression and the later part is a decoder module for reconstruction. The latent vectors are expected to capture the key representational information of the input space.

However, such classical autoencoders fail to learn a useful or well-constructed latent spaces and thus lead to unsatisfactory results in some applications.\(^{23,24}\) These shortcomings
limit the application of AEs for a wider range of problems. To address this, variational autoencoders are built upon autoencoders with an additional optimization constraint that latent space follows a certain distribution (like a normal distribution). Through this constraint, information is evenly distributed in the latent space that enables the model to sample any point for data reconstruction.

The encoder module, an inference model \( q_\phi(z|x) \), and the decoder module, a generative model \( p_\theta(x|z) \) are simultaneously trained with data \( x \) and the latent variable \( z \). Parameters \( \phi \) and \( \theta \) parameterize the encoder and decoder, respectively. VAEs model the joint distribution of the latent space and data as \( p(x,z) = p_\theta(x|z)p(z) \). The term \( p(z) \) is a prior over the latent variables which is typically chosen as a normal distribution for ease of sampling. The intractable posterior \( p_\theta(z|x) = p_\theta(x|z)p(z) / (\int p_\theta(x|z)p(z)dz) \) is approximated using the tractable variational Bayes approach which maximizes the Evidence Lower Bound (ELBO):

\[
\mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \leq \log p_\theta(x) \tag{2}
\]

where \( KL \) is the Kullback-Leibler divergence.

In our implementation, the autoencoders and variational autoencoders were developed in Python 3.7 using the Keras package with Tensorflow backend v2.4.1.

### 2.3 Performance assessment criteria

To quantify and compare the performance of autoencoders and variational autoencoders, four metrics are chosen: (1) Spearman correlation coefficient, (2) Pearson correlation coefficient (PCC), (3) root-mean-square deviation (RMSD), and (4) discrete optimized protein energy (DOPE). Each metric is introduced in turn. The correlation-based metrics have been widely applied in the comparison between dimensionality reduction methods for biomolecules. They are used here for the encoder module to measure how well the information is preserved in the latent space. The remaining metrics, RMSD and DOPE, are
used for the decoder module to compare the differences between the training and decoded structures.

Implied time scales were estimated through the construction of Markov state models (MSMs). Based on the coordinates of the protein NMP-LID angle plots, 100 cluster centers were chosen using $k$-means clustering method. Different lag times were set to calculate the transition matrix. The relaxation timescales were estimated using the corresponding second eigenvalue:

$$t(\tau) = -\frac{\tau}{\ln \lambda_1}$$

where $\lambda_1$ is the second eigenvalue and $\tau$ is the lag time. MSMs and implied timescales were calculated using the PyEMMA package$^{30}$ version 2.5.7.

3 Results

The enzyme adenosine kinase carries out large conformational transitions between the open and closed states in the adenosine triphosphate (ATP) to adenosine diphosphate (ADP) catalysis reaction. Among various structures of ADK, *E. Coli* ADK (abbreviated as ADK) was selected for this study, which is made up of a CORE domain, a LID domain and a NMP domain. According to the previous research,$^{31}$ the CORE domain is relatively rigid while the other two domains are flexible and are known to switch between open and closed conformations. To better characterize the protein conformation, the CORE-LID and CORE-NMP angles were calculated using four vectors. The protein structure, domains and vectors as illustrated in Figure 2.

There are four available crystal structures for ADK: a fully closed state (PDB id: 1AKE), a fully open state (PDB id: 4AKE), a LID-open state (PDB id: 2AK3) and a NMP-open state (PDB id: 1DVR). The fully closed and open states were used for simulations while the other two were used as references. 5 ns MD simulations were conducted for both the
open and closed states. The RMSDs were plotted in Figure 3A-B. The four characterizing vectors were also calculated and plotted as a 2D angle map in Figure 3C. Each point in this angle map corresponds to a protein conformation. It is shown that: (1) the open state simulations explored larger conformational spaces compared to the closed state ones; (2) the opening of the LID and NMP domains in the closed state is observed within short simulation time. These suggest that the transition occurs in a short time scale, which aligns with the past findings. However, given the limited simulation time, a complete transition path connecting the closed and the open state was not observed. Moreover, there is almost no overlap between the conformational spaces covered by these two states.

The Cartesian coordinates in these 5 ns simulations were scaled and used as the data set for model training. Simulations with an interval of 4 (e.g., 4, 8, ...) were extracted as the testing set and the remaining intervals are used as the training set. Therefore, the
Table 1. Residue numbers in the centers of mass of heads and tails in the four vectors as shown in figure 2.

<table>
<thead>
<tr>
<th></th>
<th>1AKE-4AKE</th>
<th>1DVR</th>
<th>2AK3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tail</td>
<td>Head</td>
<td>Tail</td>
</tr>
<tr>
<td>V3</td>
<td>115-125</td>
<td>179-185</td>
<td>124-134</td>
</tr>
</tbody>
</table>

Figure 3. MD simulations of the open and closed states. RMSDs in each trajectory are calculated with regard to the first simulation frame. The open and closed states RMSDs are plotted in (A) and (B). NMP and LID angles were calculated and shown in (C) with the closed state conformations shown in cyan and the open state conformations in pink.

The overall data set was split into 75% for training and 25% for testing. Autoencoders and variational autoencoders with different number of hidden layers were trained using this data set. Detailed model architectures are listed in Table 2.

Based on the number of layers in encoder and decoder modules, the number of neurons is adjusted to keep the same compression factor (ratio of sizes in adjacent layers) between layers. We refer to the model with \( n \) number of layers in the encoder as \( n \)-layer model (e.g., AE with 3 layers in the encoder as 3-layer AE). A total of 8 models (4 different layer numbers with 2 models) were trained using the training data and tested with the testing data. Each model was trained three times independently and the mean value of each metric
Table 2. Architectures of autoencoders and variational autoencoders. The number of neurons in the input and output layers is a fixed number of 4980 while the number of neurons in the encoder and decoder varies with the number of hidden layers. The dimension of the latent space is set to 2.

<table>
<thead>
<tr>
<th>Input</th>
<th>Hidden layers</th>
<th>Encoder size</th>
<th>Latent space size</th>
<th>Decoder size</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4980</td>
<td>1</td>
<td>128</td>
<td></td>
<td>128</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>512, 32</td>
<td></td>
<td>32, 512</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>1024, 128, 16</td>
<td>2</td>
<td>16, 128, 1024</td>
<td>4980</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1024, 256, 64, 16</td>
<td>16, 64, 256, 1024</td>
<td>4980</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2048, 512, 128, 32, 8</td>
<td>8, 32, 128, 512, 2048</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

was calculated. The results of performance assessment metrics are plotted in Figure 4.

In terms of the encoder (Figure 4A-B), a larger number of layers lead to a more complicated network that fail to keep enough biological information in the latent space. This is particularly evident in the autoencoders, in which both metrics drop sharply with increasing number of layers. In contrast, variational autoencoders kept a relatively flat curve. For the performance of the decoder (Figure 4C-D), variational autoencoders lead to a lower deviation between the training and decoded protein structures. It is worth noting that the decoded protein structures from both AEs and VAEs reached a more stable state with lower DOPE scores. Based on the elbow method, 4-layer VAE was selected as the final model with good performance and short training time.

Two decoded ADK structures in the open and the closed states through the selected 4-layer VAE are illustrated in Figure 5. The mean RMSD between the training and decoded structures is 1.03 Å. The learned latent space is plotted in Figure 6A. It is shown that the regions of the open and closed states are well separated. Also, there are blank spaces within each region. Four data points were manually selected and their decoded structures are illustrated in Figure 6C-F with the NMP-CORE and LID-CORE angles plotted in Figure 6B.

It should be noted that the latent space learned the nature of the characterizing angles as they shared similar trends. Points 1 and 2, originally selected from the open and closed states regions in the latent space, also lie in the regions of the open and closed states in the
Figure 4. Performance assessment results in (A) Spearman correlation coefficient, (B) Pearson correlation coefficient, (C) RMSD and (D) DOPE score with different models and number of hidden layers. The variational autoencoder with 4 layers performed the best with high Spearman and Pearson coefficients and low RMSD and DOPE scores.

angle map, respectively; points 3 and 4, from the middle of two states in the latent space, also locate in the boundary of these two states in the angle map. This indicates that the learned latent space can be used to generate similar or different protein conformations by selecting nearby or distant points in the latent space, respectively.

To further explore the conformational spaces starting from the generated structures, additional 5 ns simulations, following the same procedure as described in the Molecular dynamics simulations section, were conducted using the decoded structures of points 3 and 4. For comparison, the training data set of 5 ns closed state simulations was extended to 50 ns.

Two angle maps are plotted in Figure 7 to show the conformational spaces from the
Figure 5. Comparison between the ADK native structures (grey) and decoded structures (red) in (A) the open state and (B) the closed state in the 4-layer VAE.

50 ns closed state simulations and a combined MD simulations from the original 5 ns MD simulations in the open and closed states and the additional 5 ns simulations from the generated structures. It is shown that the MD simulations consisting of four short trajectories covered a similar conformational space compared with one long MD simulations. Both of these simulations explored the regions near the intermediate state of LID-open NMP-closed structure (PDB id: 2AK3). A full transition from the closed state to the open state can be constructed using both landscapes. Moreover, the combined simulations sampled hidden spaces near LID-closed NMP-open structure (PDB id: 1DVR) while these regions are less sampled in the long trajectory.

To quantitatively compare the sampling efficiency, implied timescales in both trajectories were estimated based on the 2D coordinates on the angle maps. K-means clustering method was used with 100 cluster centers. Markov state models were built and the implied timescales were calculated for each trajectory. The results are shown in Figure 8. It is shown that the
Figure 6. Four points (point 1-4 in (A) and (B)) in the latent space were selected and their decoded structures are displayed in (C)–(F). It is shown that point 1 and 2 locate in the open and closed states, respectively. Points 3 and 4 in the frontiers of the latent space also locate in the intermediate regions of the protein angle map.

short combined simulations in Figure 7B exhibited similar implied timescales as the reference trajectory.

4 Discussion

We proposed an application of variational autoencoders to sample protein conformational spaces. The model is demonstrated to capture the key variables in characterizing protein structures as the decoded conformations are similar to the training frames. This capability comes from the non-linear nature of variational autoencoders. As shown in the case of other methods,\textsuperscript{35,36} this leads to an improved ability to learn the movement of covalently bonded atoms.\textsuperscript{11}

With the high accuracy in projecting low-dimensional data points back to high-dimensional
protein structures, the latent space can be used to generate new and plausible protein structures not in the training space. Since the latent space holds a distance similarity—that is, the distances between points in the latent space are proportional to the deviations of their corresponding decoded protein structures—it can be used to produce either similar conformations by selecting points near the training set or distinct conformations from distant points. In the current study, both kinds of points were selected. The decoded protein structures from points near the training data are compared through visualization and LID NMP angle map. The produced protein structures from the intermediate regions could be used to start new MD simulations for additional sampling. This strategy led to highly efficient conformational spaces sampling with less computational cost. It should be noted that those data points were selected manually via latent space visualization in the current study. Automatic data selection for massive parallel simulations is possible within the framework of the current results.

We heuristically applied four metrics for quantification and comparison based on the previous studies. Specifically, the performance of encoder modules was determined by Spearman and Pearson correlation coefficients, as the encoder module can be treated as a dimension-
Figure 8. Estimated timescales with different lag times. The subplots (A) and (B) correspond to the trajectories in Figure 7 (A) and (B), respectively. Top 8 timescales were selected and each was plotted with 95% confidence interval.

A good model is expected to have a low DOPE score. Unexpectedly, compared with the training protein structures, the decoded protein structures in VAE have lower DOPE scores (on average of -1171). This indicates that the decoded structures have lower system energy, and thus are more stable than the training ones. However, this absolute value of difference in DOPE scores is negligible due to the fact that the DOPE scores of the training structures are lower than -24000, leading to the relative difference being below 5%.

Sampling efficiency is compared between the 20ns and 50ns trajectories. Currently, this is defined as the implied timescale based on the sampled protein NMP-LID angles. It can be seen from Figure 7 that, compared with the complete reference trajectory, the 20ns trajectory sampled similar conformational regions within shorter time. Moreover, the implied timescale calculation reveals that we can observe markovian behavior from the 20ns sampling
and get the "correct" timescales we would obtain from a 50ns simulation, showing that our assisted ML-based sampling strategy is able to capture biological-relevant transitions between conformational states with significant less sampling.

5 Conclusion

In summary, we demonstrated the success of variational autoencoders in exploring protein conformational spaces through short molecular dynamics simulations. A well-trained variational autoencoder is capable of projecting trajectories onto a low-dimensional latent space, which can be used to produce realistic conformations, either similar or distant to the training frames, that are not in the training space. This capability allows the prediction of unsampled and physically plausible protein conformations. These conformations can be used as restarters for additional MD simulations to accelerate sampling process.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

HT carried out all the computations, performed data analysis and wrote the article. XJ, FT and EL provided critical advice. SX helped with the figures. HT, EL and PT conceived and designed the research. All authors contributed to the manuscript, read, and approve the submitted version.
Funding

Research reported in this paper was supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award No. R15GM122013.

Acknowledgments

Computational time was generously provided by Southern Methodist University’s Center for Research Computing.

Data Availability Statement

The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author. The codes used for this study is available on GitHub at https://github.com/smu-tao-group/protein-VAE.

References


(12) Ramaswamy, V. K.; Musson, S. C.; Willcocks, C. G.; Degiacomi, M. T. Deep Learning


(20) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. Numerical integration of the cartesian
equations of motion of a system with constraints: molecular dynamics of n-alkanes.  


(22) Baldi, P. Autoencoders, unsupervised learning, and deep architectures. Proceedings of ICML workshop on unsupervised and transfer learning, 2012; pp 37–49.


