Exploring Protein Conformations and Functions Through Molecular Dynamics Simulations and Machine Learning

Hao Tian
Southern Methodist University, hhaootian@gmail.com

Follow this and additional works at: https://scholar.smu.edu/hum_sci_chemistry_etds

Part of the Computational Chemistry Commons

Recommended Citation
Tian, Hao, "Exploring Protein Conformations and Functions Through Molecular Dynamics Simulations and Machine Learning" (2023). Chemistry Theses and Dissertations. 34.
https://scholar.smu.edu/hum_sci_chemistry_etds/34

This Dissertation is brought to you for free and open access by the Chemistry at SMU Scholar. It has been accepted for inclusion in Chemistry Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
EXPLORING PROTEIN CONFORMATIONS AND FUNCTIONS THROUGH MOLECULAR DYNAMICS SIMULATIONS AND MACHINE LEARNING

Approved by:

Prof. Peng Tao
Associate Professor

Prof. Brian Zoltowski
Associate Professor

Prof. Devin Matthews
Assistant Professor

Prof. Eric Larson
Associate Professor
EXPLORING PROTEIN CONFORMATIONS AND FUNCTIONS
THROUGH MOLECULAR DYNAMICS SIMULATIONS
AND MACHINE LEARNING

A Dissertation Presented to the Graduate Faculty of the
Dedman College
Southern Methodist University
in
Partial Fulfillment of the Requirements
for the degree of
Doctor of Philosophy
with a
Major in Theoretical and Computational Chemistry
by
Hao Tian

B.Eng., Chemical Engineering, Beijing University of Chemical Technology, China

May 13, 2023
ACKNOWLEDGMENTS

“It was the best of times, it was the worst of times.” - Charles Dickens

It was a unique journey to pursue a Ph.D. during the COVID-19 pandemic. I still remember the day I was required to go home and take online courses. It was a tough time in life and research, but I’m glad I made it eventually.

I would like to express my deepest gratitude to my advisor, Prof. Peng Tao, for his guidance and expertise in my Ph.D. studies. I would also like to thank my committee members, Prof. Devin Matthews, Prof. Eric Larson, and Prof. Brian Zoltowski, for their insightful critiques and valuable contributions to my research. I would like to extend my thanks to my fellow graduate students for their camaraderie and support. I am grateful to my family for their unwavering love.

I want to thank my partner Xi Jiang for her love, tolerance, and patience. I would not go this far without her support.

Thank you all for your commitment, dedication, and contributions to my academic and personal growth throughout my Ph.D. study.
Proteins are essential biomacromolecules that perform a variety of critical functions in living organisms. The tertiary structure of a protein plays a crucial role in its biological activity as it determines how the protein interacts with other molecules. Consequently, understanding protein conformation and function is an important area of research with implications in medicine, biotechnology, and other fields.

The first part of this dissertation focuses on protein allostery, a process by which proteins transmit perturbations caused by binding at one site to a distal site, thereby regulating activity. With the development of computational methods like molecular dynamics simulations and machine learning, it is now possible to study protein allostery at the atomistic level. A machine learning-based framework is presented to understand the allosteric process of the light-oxygen-voltage domain of the diatom Phaeodactylum tricornutum aureochrome 1a protein. Upon understanding allostery mechanisms, the identification of allosteric sites is of great importance in allosteric drug discovery and design. To approach this problem, the Protein Allosteric Sites Server (PASSer) is designed for accurate allosteric site prediction.

The second part of this dissertation explores protein conformations using deep learning. Variational autoencoders, a class of deep learning models, are utilized to learn a low-dimensional representation that captures the essential features of high-dimensional protein
conformations. The success of this approach is demonstrated through the study of the enzyme adenosine kinase and Vivid. Furthermore, an adaptive sampling method is presented that can accelerate the exploration of protein conformational space.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>xiv</td>
</tr>
<tr>
<td>LIST OF ACRONYMS</td>
<td>xv</td>
</tr>
</tbody>
</table>

## CHAPTER

1. INTRODUCTION ......................................................... 1

2. METHODOLOGIES ....................................................... 5
   2.1. Molecular Dynamics Simulations ................................ 5
   2.2. Trajectory Analysis ........................................... 6
   2.3. Machine Learning ............................................... 7
   2.4. Performance Assessment Criteria ............................. 11
   2.5. Markov State Model ............................................ 14
   2.6. Machine Learning-Based Community Analysis ............... 14
   2.7. Transition Path Theory ...................................... 15
   2.8. Autoencoders and Variational Autoencoders ............... 16

3. DECIPHERING THE ALLOSTERIC PROCESS OF *PHAEODACTYLUM TRI-CORNUTUM* AUREOCHROME 1A PROTEIN .......................... 19
   3.1. Introduction ................................................... 19
   3.2. Markov State Model Analysis of Kinetically Separated Macrostates ...... 22
   3.3. Key Residues Identified by One-vs-One Random Forest ........ 25
   3.4. Machine Learning-Based Community Analysis of Protein Communities .... 28
   3.5. Discussion ..................................................... 32
   3.6. Conclusion ..................................................... 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>PASSER: PREDICTION OF ALLOSTERIC SITES SERVER</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>Prediction Performance of XGBoost</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Prediction Performance of GCNN</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>Prediction Performance of Ensemble Model</td>
<td>40</td>
</tr>
<tr>
<td>4.5</td>
<td>Novel Allosteric Sites Prediction</td>
<td>41</td>
</tr>
<tr>
<td>4.6</td>
<td>Web Server and Command Line Usage</td>
<td>42</td>
</tr>
<tr>
<td>4.7</td>
<td>Discussion</td>
<td>44</td>
</tr>
<tr>
<td>4.8</td>
<td>Conclusion</td>
<td>45</td>
</tr>
<tr>
<td>5</td>
<td>EXPLORING PROTEIN CONFORMATIONAL SPACE WITH VARIATIONAL AUTOENCODER</td>
<td>46</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>46</td>
</tr>
<tr>
<td>5.2</td>
<td>Learned Protein Conformational Space of ADK Protein</td>
<td>47</td>
</tr>
<tr>
<td>5.3</td>
<td>Additional Simulations From Seed Structures</td>
<td>52</td>
</tr>
<tr>
<td>5.4</td>
<td>Discussion</td>
<td>53</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>57</td>
</tr>
<tr>
<td>6</td>
<td>LAST: LATENT SPACE-ASSISTED ADAPTIVE SAMPLING FOR PROTEIN TRAJECTORIES</td>
<td>58</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>58</td>
</tr>
<tr>
<td>6.2</td>
<td>Latent Space-Assisted Adaptive Sampling</td>
<td>60</td>
</tr>
<tr>
<td>6.3</td>
<td>A Comparative Study of LAST, SDS, and cMD</td>
<td>63</td>
</tr>
<tr>
<td>6.4</td>
<td>Discussion</td>
<td>69</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusion</td>
<td>73</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>BIBLIOGRAPHY</td>
<td>77</td>
</tr>
</tbody>
</table>
APPENDIX

A  APPENDIX (FIRST AUTHOR ONLY) ................................................. 88

A.1. PAPER I ................................................................. 89

A.2. PAPER II ............................................................. 103

A.3. PAPER III ............................................................ 117

A.4. PAPER IV ............................................................. 126

A.5. PAPER V ............................................................. 137

A.6. PAPER VI ............................................................ 149

A.7. PAPER VII .......................................................... 156
<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>25</td>
</tr>
</tbody>
</table>

**2.1** Architecture of 1-layer GCNN model. 1-layer GCNN is composed of a graph convolution layer, a readout layer and a linear classification layer. Rectified linear unit is used as the activation function. An atomic graph is constructed for a given pocket and GCNN predicts the probability of this pocket being an allosteric site.

**2.2** 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.

**3.1** The native dark and light structures of AuLOV monomer. (A) Native dark structure; (B) Native light structure; (C) A covalent bond is formed between C4a position of FMN and sulfur atom in Cys 287 upon light excitation.

**3.2** t-SNE dimensionality reduction result and the implied timescales with regard to different lag times. (A) The distribution of four protein states onto the 2-dimensional space; (B) Estimated relaxation timescale with different lag times calculated by MSMs.

**3.3** Macrostates in MSM with transition probability. Based on the transition probabilities, states 2 and 9 were considered as the native dark and native light states among macrostates, respectively. Other macrostates were treated as intermediate states.

**3.4** Representative conformations in the transient dark and light states. Monomer structures in (A) macrostate 3 from the transient dark state, and (B) macrostate 5 from the transient light state. Corresponding structures in native dark and native light states are shown in transparent grey color.
3.5 Tree-based models for macrostate classification. (A) Prediction accuracy of the random forest model with different tree depths; (B) Prediction accuracy of the one-vs-one random forest model with different tree depths; (C) Accumulated explained importance of the OvO random forest model in 8 tree depth with regard to the number of features. The top 550 features account for 90.2% of the overall importance. .................................................. 26

3.6 ML based Community analysis results of AuLOV. (A) Total feature importance among ML communities with regard to different number of communities; (B) Four ML communities named as Commu. A, Commu. B, Commu. C and Commu. D are illustrated in blue, orange, red and grey colors, respectively. ............................................................. 29

3.7 Change of community importance along transitions. Community importance change along two identified paths from the native dark state to the native light state. (A) 2 (native dark) - 3 - 5 - 7 - 6 - 9 (native light); (B) 2 (native dark) - 3 - 5 - 6 - 9 (native light). .................................................. 33

4.1 Fine-tuned results of weight and depth parameters in XGBoost model. Two sets of parameters, depth 5 weight 40 and depth 8 weight 40, exhibited exceptional high F1 score and were selected for further prediction. ............ 38

4.2 Fine-tuned results of ratio and distance threshold parameters in GCNN model. (A) The ratio between number of negative labels and number of positive labels was fine-tuned. Ratio of 2 is considered reaching a balance between recall and precision. (B) The atomic distance threshold was fine-tuned from 7 to 11 Å. There is no significant increase in F1 score after 10Å, which is selected as the distance cutoff. For each parameter value, GCNN was run 10 times independently. .......................................................... 40

4.3 Prediction results of two examples not included in the training set: (A) Dynamics-driven PDZ2 protein in bound states (PDB ID 3LNY); (B) The LOV domain of conformational-driven Phaeodactylum tricornutum Aureochrome 1a in dark state (PDB ID: 5DKK). Red regions are the most probable pockets in the predicted results with probabilities of (A) 45.14% and (B) 89.46% and are also the true allosteric sites. ................................. 41

4.4 Web server workflow. User can upload either a PDB ID or a PDB file in the web server. FPocket is used to detect pockets. For each pocket, physical properties are calculated and predicted using a pretrained XGBoost model; while an atomic graph is constructed and fed into a pretrained GCNN model. The final probability is given by averaging results from both models. 42
4.5  PASSer web server pages. (A) Users can either submit a PDB ID or a PDB file in the home page. (B) Predicted top 3 pockets are summarized in a table with corresponding probabilities and pocket residues. (C) Protein structures and pocket sites are displayed in an interactive window.  

5.1  Protein structures of ADK in (A) open state and (B) closed state. The LID, NMP and CORE regions are colored as orange, blue, and grey, respectively. Four vectors, V1-V2 for CORE-NMP angle and V3-V4 for CORE-LID angle, are used to characterize protein conformations. The related residues are illustrated in table 5.1.  

5.2  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.  

5.3  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.  

5.4  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.  

5.5  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.  

5.6  Protein conformational spaces from (A) 50 ns simulation from the closed state and (B) four sets of 5 ns simulation from the closed state, the open state and two generated conformations (point 3 and 4).  

5.7  Estimated timescales with different lag times. The subplots (A) and (B) correspond to the trajectories in Figure 5.6 (A) and (B), respectively. Top 8 timescales were selected and each was plotted with 95% confidence interval.
6.1 Workflow of LAST method. To begin with, a short MD simulation is conducted from crystal structures. All sampled conformations are stored in a pool. In each round, if there is no increase of the maximum RMSD of the newly sampled conformations in consecutive five rounds, the workflow stops. Otherwise, a VAE model is trained using all conformations in the pool. Then, the seeds are selected on the latent space. For each of the selected seeds, new MD simulations are conducted, and the sampled conformations are stored in the pool. ................................. 61

6.2 Structures of (A) ADK and (B) VVD. ADK is composed of a CORE domain, an LID domain, and an NMP domain. LID-CORE and NMP-CORE angles are calculated by four vectors to represent protein conformations. Both proteins are colored at the secondary structure level using ChimeraX. ...... 64

6.3 Seed structure distribution on the low-dimensional protein representations. (A, B) ADK protein conformations are represented in LID-CORE and NMP-CORE angle vectors. (C, D) VVD protein conformations are represented in RMSDs with regard to the native dark and native light states. Seed structures are represented in red stars. The analysis in these plots was carried out after seven rounds of LAST simulations for illustration purposes. 65

6.4 Comparison of ADK conformational spaces of LAST and cMD. Protein conformations are shown in blue at iterations 1, 5, 10, and 15 in the LAST method. Protein conformations produced by cMD are shown in gray with the same simulation time. In each round, LAST explored a larger conformational space compared with cMD. ................................. 66

6.5 Explored conformational spaces of (A) ADK and (B) VVD proteins using LAST (red dashed line), SDS (blue dotted line), and cMD (black solid line) methods. LAST took 22 and 30 iterations to complete for ADK and VVD proteins, respectively. In each system, SDS and cMD simulations were conducted under the same simulation time. In the ADK conformational space, LAST explored two paths to the open and closed states, while SDS explored one path toward the open state.......................... 68

6.6 Mean RMSDs in (A) ADK and (B) VVD systems. Mean RMSD values are connected with black lines. The standard deviation in each iteration is plotted as vertical red lines. The gap between minimum and maximum RMSD values is colored as a light gray background. ......................... 69
LIST OF TABLES

Table                                         Page

3.1 Top 20 residues identified by OvO random forest. ........................................ 27
3.2 Accumulated feature importance of secondary structures in AuLOV. .......... 28
3.3 Accumulated feature importance between each ML community pair. .......... 30
3.4 The changes of Commu. A, Commu. B and Commu. C during transitions between macrostates. ................................................................. 31
3.5 The probability of top 10 channels simulated using transition path theory. .... 32

4.1 Evaluation and performance comparison of different models. The average values and standard errors of 6 indicators are calculated in 10 independent runs. The ensemble learning method can achieve better performance compared to single XGBoost and GCNN models. .................................................. 39
4.2 Probabilities of predicting allosteric sites in the top 3 positions. Ensemble learning method can rank an allosteric site in the top 3 positions with a probability of 84.9%, which is higher than previous results. ................. 39

5.1 Residue numbers in the centers of mass of heads and tails in the four vectors as shown in figure 5.1. ................................................................. 48
5.2 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. ................. 50
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADK</td>
<td>Adenosine Kinase</td>
</tr>
<tr>
<td>AE</td>
<td>Autoencoder</td>
</tr>
<tr>
<td>ATP</td>
<td>Adenosine Triphosphate</td>
</tr>
<tr>
<td>bZIP</td>
<td>Basic Region Leucine Zipper</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CV</td>
<td>Collective Variable</td>
</tr>
<tr>
<td>DGL</td>
<td>Deep Graph Library</td>
</tr>
<tr>
<td>FAD</td>
<td>Flavin Adenine Dinucleotide</td>
</tr>
<tr>
<td>FMN</td>
<td>Flavin Mononucleotide</td>
</tr>
<tr>
<td>GCNN</td>
<td>Graph Convolutional Neural Network</td>
</tr>
<tr>
<td>LOV</td>
<td>Light Oxygen Voltage</td>
</tr>
<tr>
<td>MD</td>
<td>Molecular Dynamics</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MSM</td>
<td>Markov State Model</td>
</tr>
<tr>
<td>PAS</td>
<td>Per-Arnt-Sim</td>
</tr>
<tr>
<td>PASSer</td>
<td>Protein Allosteric Sites Server</td>
</tr>
<tr>
<td>PBC</td>
<td>Periodic Boundary Condition</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>PCC</td>
<td>Pearson Correlation Coefficient</td>
</tr>
<tr>
<td>PDB</td>
<td>Protein Data Bank</td>
</tr>
<tr>
<td>PDZ2</td>
<td>The Second PDZ Domain</td>
</tr>
</tbody>
</table>

xv
<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PtAu1a</td>
<td>Diatom <em>Phaeodactylum tricornutum</em> aureochrome 1a</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forest</td>
</tr>
<tr>
<td>RMSD</td>
<td>Root-Mean-Square Deviation</td>
</tr>
<tr>
<td>SDS</td>
<td>Structural Dissimilarity Sampling</td>
</tr>
<tr>
<td>SNN</td>
<td>Siamese Neural Network</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>t-ICA</td>
<td>Time-Structure Based Independent Component Analysis</td>
</tr>
<tr>
<td>TPT</td>
<td>Transition Path Theory</td>
</tr>
<tr>
<td>t-SNE</td>
<td>t-Distributed Stochastic Neighbor Embedding</td>
</tr>
<tr>
<td>VAE</td>
<td>Variational Autoencoder</td>
</tr>
<tr>
<td>VVD</td>
<td>Vivid</td>
</tr>
<tr>
<td>XGBoost</td>
<td>eXtreme Gradient Boosting</td>
</tr>
</tbody>
</table>
To Xi, my partner and beloved.
Chapter 1
INTRODUCTION

Protein conformations and functions are regulated through allostery, whereby proteins transmit the perturbation caused by the effect of binding at one site to a distal functional site. [1] The allosteric process is fundamental in the regulation of activity.

Light, oxygen, and voltage (LOV) domains are a subdivision of the Per-Arnt-Sim (PAS) superfamily that are sensitive to blue light and undergo conformational as well as dynamical changes upon light activation. [2, 3] This activation begins with the formation of a covalent bond between a cofactor and a conserved cysteine residue. Possible cofactors include flavin adenine dinucleotide (FAD), flavin mononucleotide (FMN), or riboflavin. [4] This covalent bond further promotes the overall structural changes, resulting in the alteration of the protein-protein interactions and thus signal transduction. [5]

Various computational methods have been applied to explore protein allosteric mechanisms at the atomic level. [6–8] Molecular dynamics (MD) simulations are capable of providing atomic-scale information, as well as structure-function relationships, [9,10] and are widely used in sampling protein motions and structure landscapes. [11] To obtain more biologically meaningful information from trajectories, Markov state models (MSMs) are often used to extract asymptotic kinetic information based on limited simulations. [12,13] Kinetically separate macrostates can be obtained from MSMs in the reduced dimension. Differences among these subspaces can then be quantified to gain insight into protein structure and function relations.
The success of MSMs depends on appropriate dimensionality reduction methods that can preserve global distances while retaining the most structural information. [14] New dimensionality reduction methods have been developed to project the high-dimensional trajectories to lower dimensions for thorough study. However, many methods, such as principal component analysis (PCA), [15] time-structure-based independent component analysis (t-ICA), [16] and t-distributed stochastic neighbor embedding (t-SNE) method, [17] suffer from problems including maintaining the similarity between high-dimensional space and low-dimensional space and are not resistant to system noise. [18] Thus, appropriate dimensionality reduction methods are needed for MD simulation data.

The study of allosteric proteins promotes the research on allosteric drugs. Compared with non-allosteric drugs, allosteric drugs have many advantages: they are conserved and highly specific; [19] they can either activate or inhibit proteins; they can be used in conjunction with orthosteric (non-allosteric) drugs. Although allosteric drugs are essential in the pharmaceutical industry, [20] they are still poorly understood. [21] Most allosteric mechanisms remain elusive because of the difficulty of identifying potential allosteric sites. [22] Thus, identifying allosteric sites is important for allosteric drug development and has attracted a wide range of interests.

MD simulations have been applied extensively to understand protein structure, function, and kinetics. [23, 24] Through the development of hardware and software, e.g., GPUs [25] and OpenMM [26], 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. [27] Moreover, the energy landscapes of proteins are discretized with many local energy minima separated by high energy barriers. [28] This rough energy landscape limits the applications of MD simulations and hinders a complete sampling of protein movements.
In Chapter 3, the allosteric process of *Phaeodactylum tricornutum* aureochrome 1a (PtAu1a) is studied using a combination of computational methods. PtAu1a is a recently discovered LOV protein that consists of an unstructured N-terminal region and a basic region leucine zipper (bZIP) DNA-binding domain connected to a C-terminal LOV core. [29] A photo-induced covalent bond is formed between the C4a position of the cofactor FMN and a nearby sulfur in Cys287. This covalent bond triggers a series of conformational changes, including the undocking and unfolding of Jα helix from the LOV core surface, the release of A’α helix from the hydrophobic site on the LOV domain surface, and dimerization of the LOV domains. [30] These events lead to increased PtAu1a affinity for DNA binding and are proposed to be allosteric. [31]

In Chapter 4, an ensemble model is presented to predict protein allosteric sites. The PASSer: Protein Allosteric Sites Server (https://passer.smu.edu) is deployed with trained machine learning models and has been extensively tested to complete prediction within seconds. The prediction submissions and result retrievals can be accessed via the web page or API. The web server is designed to provide insights for further analysis in drug discovery.

In Chapter 5, a deep learning model, variational autoencoder (VAE), is applied to the enzyme adenosine kinase (ADK). The VAE model is proven to learn a low-dimensional latent space that can capture high-dimensional protein structures. Additional MD simulations starting from the predicted conformations, together with the training simulations, sampled a complete transition from the closed to the open states and explored hidden conformational spaces.

In Chapter 6, latent-space adaptive sampling (LAST) is proposed as a new adaptive sampling method to accelerate the exploration of protein conformational space. The LAST method was applied on four conformations in two protein systems: two metastable states of *E. Coli* adenosine kinase (ADK) and two native states of Vivid (VVD), and compared with traditional MD simulations and structural dissimilarity sampling (SDS) method. In
metastable ADK simulations, LAST explored two transition paths toward two stable states, while SDS explored only one and cMD neither. In VVD light state simulations, LAST was three times faster than cMD simulation with a similar conformational space.

Overall, this thesis studies protein conformations and functions using a combination of computational methods. Machine learning-based workflows are introduced that can help understand and predict allostery. This thesis further expands the scope to general protein conformations and presents a novel adaptive sampling method to accelerate protein sampling process.
2.1. Molecular Dynamics Simulations

The initial structures of native dark state monomer and native light state of AuLOV dimer were taken from the Protein DataBank (PDB) [32] with the PDB ID being 5dkk for the native dark state and 5dkl for the native light state. To keep the same number of residues in all structures, the longest common residue sequences (from Ser240 to Glu367) were modeled. Both the native dark and light structures contain FMN as a cofactor. The force field for the cofactor FMN was used from a previous study. [33] In order to fully explore the protein dynamics with regard to the formation of the covalent bond between cysteine 287 and FMN, two new transient states, referred to as transient dark state and transient light state, were generated. Specifically, the transient dark state was generated by forming the Cysteinyl-Flavin C4a adduct bond in the native dark state structure. The transient light state was generated by removing the Cysteinyl-Flavin C4a adduct bond and constructing the dark state configuration in the native light state structure. These transient structures facilitate analysis of allosteric interconversion between the light- and dark-state structures.

The crystal structures were added with hydrogen atoms and were further solvated in a water box with the TIP3P water molecules. [34] Sodium cations and chloride anions were added for charge neutralization. For each structure, energy minimization was done with the steep descent method and the adopted basis Newton-Raphson minimization. System temperature was raised to 300K through a 20 picoseconds (ps) MD simulations. Another 20ps simulations were done for equilibrium. 10 nanoseconds (ns) of isothermal-isobaric ensemble (NPT) followed by 1.1 microseconds (µs) of canonical ensemble (NVT) Langevin
MD simulations were carried out at 300K. The first 0.1µs NVT simulations was considered as an equilibration stage and was discarded. Three NVT MD simulations were conducted independently for each protein structure. Therefore, a total of 12µs simulations were generated for analysis. SHAKE method was used to constrain all bonds associated with hydrogen atoms. 2 femtoseconds (fs) step size was used for all MD simulations. Trajectories were saved for every 100ps. Periodic boundary condition (PBC) was applied in simulations. Particle mesh Ewald (PME) algorithm was used to calculate the electrostatic interactions. [35] MD simulations were conducted using GPU accelerated OpenMM and CHARMM27 force field. [36,37]

2.2. Trajectory Analysis

Root-Mean-Square Deviation (RMSD) and Root-Mean-Square Fluctuation (RMSF)

The dynamics stability of a MD simulation trajectory is measured by the root-mean-square deviation, which is calculated as:

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(r_i^0 - Ur_i)^2}{N}}$$

(2.1)

where $r_i^0 = (x_i, y_i, z_i)$ represents the coordinate of an atom $i$ in Cartesian coordinate system and $U$ is the most appropriate alignment transformation matrix between two structures. For each trajectory, the first frame was treated as the reference structure.

The root-mean-square fluctuation is used to measure the fluctuation of atoms in each frame with regard to the first frame in a MD simulation trajectory. Specifically, Cα atoms were considered important in representing the protein motions and the corresponding RMSFs of each Cα were calculated as:

$$\text{RMSF} = \sqrt{\frac{1}{N} \sum_{i=1}^{N}(r_i^0 - Ur_i - U_r)^2}$$

where $r_i^0 = (x_i, y_i, z_i)$ represents the coordinate of an atom $i$ in Cartesian coordinate system and $U$ is the most appropriate alignment transformation matrix between two structures. For each trajectory, the first frame was treated as the reference structure.
\[
\text{RMSF}_i = \sqrt{\frac{1}{T} \sum_{j=1}^{T} (\mathbf{r}_i(t) - \mathbf{\bar{r}}_i)^2}
\]  

where \( T \) is the number of frames and \( \mathbf{\bar{r}}_i \) is the averaged Cartesian coordinate of the \( i^{th} \) \( \text{C\alpha} \) in the given trajectory.

**Feature Processing**

The \( 3N \) degrees of freedom in the Cartesian coordinate system hinders a thorough analysis of MD simulations in biological systems. Pairwise \( \text{C\alpha} \) distances are usually extracted to represent the structural characteristics of protein configurations. [38] In the current study, a feature vector of each structure was constructed by calculating the distance pairs between one \( \alpha \) carbon atom and another \( \alpha \) carbon atoms in amino acids following the order of residue sequence. This feature vector was further encoded by a previously proposed transformation method with a cutoff of 10 Å. [39]

**2.3. Machine Learning**

**Random Forest and One-vs-one Random Forest**

Random forest as a tree-based machine learning technique was applied to learn the structural differences among macrostates in this study. [40,41] Each random forest model is composed of 50 decision trees. Decision trees were trained individually, and the final result of a random forest model is formed by a voting algorithm. Scikit-learn version 0.20.1 was used to implement the random forest model. [42]

The random forest model overcomes the problem of overfitting by employing several decision trees. However, in multi-task classification jobs, one-vs-one random forest model is more common and superior to random forest model by constructing one classifier for each pair of classes. [43] The overall output is the weighted sum of all base classifiers. In the current
study, 10 macrostates were trained with 45 random forest models. One-vs-one random forest model provides weighted sum of overall feature importance with specific feature importance regarding two given classes.

**Feature Importance**

The feature importance in a random forest model is calculated using the Gini impurity, which is calculated as:

\[
\text{Gini impurity} = \sum_{i=1}^{C} -f_i(1 - f_i) \tag{2.3}
\]

where \(f_i\) and \(C\) are the frequency of one label at a node that are chosen to divide the data set and the number of labels, respectively. A random forest model consists of multiple decision tree models. The importance of feature \(i\) in each decision tree is calculated as:

\[
f_i = \frac{\sum_j s_{nj}}{\sum_{k \in \text{all nodes}} n_k} \tag{2.4}
\]

where \(s\) is the frequency of node \(j\) split on feature \(i\). The importance of feature \(i\) in a random forest model is calculated by averaging its importance among decision tree models:

\[
F_i = \frac{\sum_{j \in \text{all decision trees}} \text{norm}f_i}{N} \tag{2.5}
\]

where \text{norm} \(f_i\) and \(N\) are the normalized feature importance of one decision tree and the number of decision trees, respectively. [44]

Pairwise Cα distances were extracted as the input features and the corresponding feature importance was calculated. For each Cα distance, the importance was added to the related two residues. The accumulated feature importance of residues implies their contributions in the allosteric process.
**eXtreme Gradient Boosting**

Extreme gradient boosting (XGBoost) is an ensemble learning method that combines several decision trees in sequence.

Let $D = \{(x_i, y_i) | D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}^n\}$ represents a dataset with $m$ features and $n$ labels. The $j$-th decision tree in XGBoost predicts a sample $(x_i, y_i)$ by:

$$g_j(x_i) = w_q(x_i)$$  \hspace{1cm} (2.6)

where $w_q$ is the leaf weights of this decision tree. The final prediction of XGBoost is given by the summation of predictions from each decision tree:

$$\hat{y}_i = \sum_{j=1}^{M} g_j(x_i)$$  \hspace{1cm} (2.7)

where $M$ is the total number of decision trees. To overcome overfitting introduced by decision trees, the objective function in XGBoost is composed of a loss function $l$ and a regularization term $\Omega$:

$$\text{obj}(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{M} \Omega(f_i)$$  \hspace{1cm} (2.8)

where $\Omega(f) = \gamma T + \frac{\lambda}{2} \sum_{t=1}^{T} \omega_t^2$ with $T$ represents the number of leaves and $\gamma, \lambda$ are regularization parameters.

During training, XGBoost iteratively adds new decision trees. The prediction of the $t$-th iteration is expressed as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + g_t(x_i)$$  \hspace{1cm} (2.9)
Correspondingly, the objective function of the $t$-th iteration is:

$$
\text{obj}^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)}) + g_t(x_i)) + \Omega(f_i)
$$  \hspace{1cm} (2.10)

XGBoost introduces both first derivative and second derivative of the loss function. By applying Taylor expansion on the objective function at second order, the objective function of the $t$-th iteration can be expressed as:

$$
\text{obj}^{(t)} \simeq \sum_{i=1}^{N} \left[ l(y_i, \hat{y}_i^{(t-1)}) + \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) f_t(x_i) \\
+ \frac{1}{2} \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) f_t^2(x_i) \right] + \Omega(f_i)
$$  \hspace{1cm} (2.11)

XGBoost can predict the labels of sample data with corresponding probabilities. For one pocket, XGBoost outputs the probability of this pocket being an allosteric pocket. This pocket is labeled as positive (allosteric) if the predicted probability is over 50% or negative otherwise.

The XGBoost algorithm is implemented using Scikit-learn package version 0.23.2. [42]

**Graph Convolutional Neural Network**

Graph convolutional neural network (GCNN) in this work follows this formula: [45]

$$
H^{(l+1)} = \text{ReLU}(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)})
$$  \hspace{1cm} (2.12)

where $H^{(l)}$ and $H^{(l+1)}$ represents the $l^{th}$ and $l+1^{th}$ layer, respectively. $H^{(l)} \in \mathbb{R}^{N \times D}$, where $N$ is the number of nodes and $D$ is the number of features. Rectified linear unit
(ReLU$(x) = \max(0, x)$) is used as the activation function. $W^{(l)}$ denotes the weight matrix in the $l$th layer. $D$ and $A$ represent degree matrix and adjacent matrix, respectively, with $\tilde{D}_ii = \sum_j \tilde{A}_{ij}$. Renormalization (indicated by $\sim$ symbol) is applied for the undirected graph $G$ where each node is added with a self-connection. Therefore, $\tilde{A} = A + I_N$ where $I_N$ is the identity matrix.

A graph readout is calculated through the average of node features for each graph.

$$h_g = \frac{1}{|V|} \sum_{v \in V} h_v$$  \hspace{1cm} (2.13)

where $h_g$ is the readout result of graph $g$ and $h_v$ is the node feature in node $v$. $V$ represents the nodes in graph $g$.

An example of 1-layer GCNN model is shown in Figure 2.1. A graph is first fed into a convolution layer. The in-degree and out-degree of a node refer to the number of edges coming into and going out from that node, respectively. In-degree of each node was calculated as the node feature. Graph feature is calculated as the average of node features in the readout layer with ReLU activation function. The output was further fed into a linear classification layer $g$, which predicts the probability of being an allosteric pocket. Previous research has shown the limitations of 1-layer GCNN. [46] In the current study, atomic graphs of each protein pocket are constructed and fed into a 2-layer GCNN model. This model consists of two graph convolution layers, each with 256 dimensions of a hidden node feature vector, followed by a readout layer and a linear classification layer. The node degree is used as the initial node feature. The in-degree is the same as the out-degree in an undirected atomic graph. Graph representation is calculated as the average of node representations. The linear classification layer outputs the probabilities of pockets being allosteric sites.

The GCNN model is implemented using Deep graph library (DGL) package v0.4.3. [47]
2.4. Performance Assessment Criteria

Pearson Correlation Coefficient (PCC)

Pearson correlation coefficient reflects the linear correlation between two variables. [48] PCC has been rigorously applied to estimate the linear relation between distances in the original space and the reduced space. [49] L2 distance, which is also called Euclidean distance, is used for the distance calculation and is shown as follows:

$$d_2(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$ \hspace{1cm} (2.14)

Based on the L2 distance expression, PCC is calculated as:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$ \hspace{1cm} (2.15)

where \( n \) is the sample size, \( x_i, y_i, \bar{x}, \bar{y} \) are the distances and the mean value of distances, respectively. \( x \) and \( y \) represent distances in original data and projected data, respectively.
Spearman’s Rank-Order Correlation Coefficient

Spearman’s rank-order correlation coefficient is used to quantitatively analyze how well distances between all pairs of points in the original spaces have been preserved in the reduced dimensions. Specifically, Spearman correlation coefficient measures the difference in distance ranking, which is calculated as the following:

\[ \rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \]  

(2.16)

where \( d_i \) is the difference in paired ranks and \( n \) equals the total number of samples.

Mantel Test

The Mantel test is a non-parametric method that is originally used in genetics, [50] which tests the correlation between two distance matrices. A common problem in evaluating the correlation coefficient is that distances are dependent to each other and therefore cannot be determined directly. The Mantel test overcomes this obstacle through permutations of the rows and columns of one of the matrices. The correlation between two matrices is calculated at each permutation. MantelTest GitHub repository was used to implement the algorithm. [51]

Shannon Information Content (IC)

While chemical information in the original space could be lost to a certain degree in the reduced space, dimensionality reduction methods are expected to keep the maximum information. Shannon information content is applied to test the information preservation in the reduced space, which is defined as:

\[ I(x) = -\log_2(P) \]  

(2.17)
where $P$ is the probability of a specific event $x$.

To avoid the possible dependency among different features in the reduced dimensions, original space was reduced to 1 dimension (1D) to calculate the IC. The values in the 1D were sorted and put into 100 bins of the same length. The bins were treated as events and the corresponding probabilities were calculated as the ratio of the number of samples in each bin to the total number of samples.

2.5. Markov State Model

The long timescale protein dynamics is tracked by the Markov state model. [52] Each simulation frame is assigned to different microstates through MiniBatch k-means clustering method. Compared with microstates, macrostates are more biologically meaningful as they are considered as kinetically separate equilibrium states. 10 macrostates were generated by Perron-cluster cluster analysis (PCCA). [53] Lag time is needed to build a MSM and was determined as 40 ns based on the implied relaxation timescale. Transition matrix and corresponding transition probabilities were estimated based on this MSM. MSMBuild package (version 3.8.0) was used to build MSMS. [54]

2.6. Machine Learning-Based Community Analysis

Machine learning-based community analysis is a newly proposed method by Zhou et al., [55] which groups residues into communities. The main idea of this analysis is maximizing the overall feature importance across different communities while minimizing the total feature importance within each community. For an undirected graph characterizing the protein, nodes can be used to represent residues, and edges can be used to represent weighted Cα distances. For node $i$ in community $C_m$, the inner edges of $i$ are defined as the summation of edge values between node $i$ and any other node in $C_m$, whereas the external edges of $i$ are defined as the summation of edge values between node $i$ and any other node in other communities. For each iteration of ML communities partition, node $i$ can be moved to
another community or swapped with another node in different communities. The benefit of these two explorative moves can be calculated as the external edges subtracted by the inner edges. The algorithm of this community analysis method is listed below.

1. ML communities are randomly partitioned;
2. The benefits of moving one node into another community and swapping one node with another between different communities are estimated to search for the maximum moving and swapping strategy, respectively;
3. One moving or swapping strategy with the highest benefit is chosen;
4. Repeat steps 2 and 3 with new ML community configuration until the highest benefit of all moving and swapping strategy is less than 0;
5. ML communities construction is completed if any strategy will increase the number of inner edges for each ML community.

The Kernighan-Lin algorithm has been implemented to search for local minimum values in graph theory. [56,57] In the current research, the feature importance of Cα distances from the one-vs-one random forest model based on AuLOV dimer simulations was used. In order to apply ML-based community analysis on monomer, the averaged importance for each Cα distance in monomer was calculated based on the dimer feature importance results.

2.7. Transition Path Theory

Transition path theory (TPT) is used to identify the most probable routes from one macrostate to another. [58,59] Dark state and light state were chosen based on the transition probability estimated in MSMs as initial and final states, respectively. All other states are considered as intermediate states. Possible transition paths from the dark state to the light state were simulated. The definition of the committor probability $q_i^+$ is the probability from
one state to a target state. Based on this definition, \( q_i^+ \) is equal to zero for all microstates in initial state and \( q_i^- \) is equal to one for all microstates in final state. The committor probability of other microstates is calculated as:

\[
-q_i^+ + \sum_{k \in I} T_{ik} q_k^+ = - \sum_{k \in \text{target state}} T_{ik}
\]  

(2.18)

where \( T \) is the transition probability matrix and \( T_{ik} \) represents the transition probability from state \( i \) to state \( k \).

\[
f_{ij} = \pi_i q_i^- T_{ij} q_j^+
\]

(2.19)

where \( \pi \) is the stationary probability of \( T \) and \( \pi_i T_{ij} \) is the absolute probability of finding the system at the transition from \( i \) to \( j \). \( q_i^- \) is the backward committor probability calculated as \( q_i^- = 1 - q_i^+ \). The backward flux \( f_{ji} \) were also considered and subtracted in calculating the net flux \( f_{ij}^+ = \max(0, f_{ij} - f_{ji}) \).

The flux from the initial state to the final state can be decomposed to individual pathways \( p_i \), which can be calculated as:

\[
p_i = \frac{f_i}{\sum_j f_j}
\]

(2.20)

### 2.8. Autoencoders and Variational Autoencoders

Autoencoders (AEs) 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. [60] For this purpose, autoencoders normally have a hourglass shaped architecture, as shown in Figure 2.2. The first part of the hourglass is an encoder module for compression and the later part
Figure 2.2: 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.

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. [61,62] These shortcomings limit the application of AEs for a wider range of problems. To address this, variational autoencoders (VAEs) are built upon autoencoders with an additional optimization constraint that latent space follows a certain distribution (like a normal distribution). [63] 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):

$$
\mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \leq \log p_\theta(x) \quad (2.21)
$$

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. [64]
Chapter 3

DECIPHERING THE ALLOSTERIC PROCESS OF PHAEODACTYLM TRICORNUTUM AUREOCHROME 1A PROTEIN

3.1. Introduction

*Phaeodactylum tricornutum* aureochrome 1a (PtAu1a) is a recently discovered LOV protein that consists of an unstructured N-terminal region and a basic region leucine zipper (bZIP) DNA-binding domain connected to a C-terminal LOV core. [29] The LOV domain, together with two flanking helices (A’α and Jα), is usually referred to as AuLOV. [65] Figure 3.1 shows the structure of AuLOV. The protein is dynamically stable in the dark state due to the interaction between the LOV core and bZIP. [30] This interaction prohibits the protein binding with DNA. [66] A photoinduced covalent bond is formed between the C4a position of the cofactor FMN and a nearby sulfur in Cys287. This covalent bond triggers a series of conformational changes, including the undocking and unfolding of the Jα helix from the LOV core surface, the release of the A’α helix from the hydrophobic site on the LOV domain surface, and dimerization of the LOV domains. [30]

These events lead to the increase of PtAu1a affinity for DNA binding and are proposed to be allosteric. [31] Recent research has revealed that a combination of structural changes in the LOV core and the undocking of Jα helix are essential for the release of the A’α helix and LOV domain dimerization. [30] The allosteric mechanism in PtAu1a is considered to be different from other LOV proteins, since the location of the LOV domain is in the C-terminus
Various computational methods have been applied to explore protein allosteric mechanisms at the atomic level. [6–8] Molecular dynamics (MD) simulations are capable of providing atomic-scale information, as well as structure-function relationships, [9, 10] and are widely used in sampling protein motions and structure landscapes. [11] The significant computational power provided by graphical processing units (GPUs) has promoted the time scale of MD simulations from nanoseconds to milliseconds. [69, 70] To obtain more biologically meaningful information from trajectories, Markov state models (MSMs) are often used to extract asymptotic kinetic information based on limited simulations. [12, 13] Kinetically separate macrostates can be obtained from MSMs in the reduced dimension. Differences among these subspaces can then be quantified to gain insight into protein structure and function relations.

The success of MSMs depends on appropriate dimensionality reduction methods that can preserve global distances while retaining the most structural information. [14] New dimensionality reduction methods have been developed to project the high-dimensional trajectories
to lower dimensions for thorough study. However, many methods, such as principal component analysis (PCA), [15] time-structure based independent component analysis (t-ICA), [16] and t-distributed stochastic neighbor embedding (t-SNE) method, [17] suffer from problems including maintaining the similarity between high dimensional space and low dimensional space, and are not resistant to system noise. [18] In the current study, MD simulations were projected onto a 2D space via the ivis framework, [71] which is a nonlinear method based on Siamese neural networks (SNNs) and has been shown powerful in interpreting biological systems. [72]

Machine learning has recently achieved great accomplishments in chemistry and biology. Raccuglia et al. applied machine learning algorithms trained on failed experimental data to predict reaction results with high accuracy. [73] Faber et al. employed machine learning techniques for feature vector representations of crystal structures. [74] Botu et al. integrated machine learning framework to accelerate ab initio molecular dynamics simulation. [75] The broad applications of machine learning stem from the ability to process large datasets and, more importantly, provide explanatory details. [76, 77] These favorable metrics offer a new prospective direction for the research on protein allostery. In this study, two tree-based machine learning models, random forest (RF) and one-vs-one random forest (OvO RF), were used to study the structural differences between macrostates and determine the contribution of residues to the allosteric process. In combination with machine learning and dynamic community analysis, Zhou et al. developed a new approach, known as machine learning based community analysis [55], to identify important structural features in dynamic-driven protein allostery. Here we applied this method on AuLOV and demonstrated the feasibility of this method in analyzing conformational-driven protein allostery.

The AuLOV is investigated in this study through the MD simulations, tree-based machine learning models, machine learning-based community analysis, and transition path theory. Our results identified key residues that are consistent with experimental discoveries and
suggested the importance of the Cα helix, overlooked thus far. Moreover, we quantified the important role of N- and C-terminal linkers in modulating AuLOV allostery. The integrated methods determined the importance of each residue in the allosteric process and therefore provided new insights into the allosteric mechanisms, which may promote future research on PtAu1a as an optogenetic tool.

3.2. Markov State Model Analysis of Kinetically Separated Macrostates

To represent the protein structure and movements, pairwise Cα distances were calculated as the representation of protein configurations. A total of 32,131 Cα distances were extracted from the AuLOV dimer, composed of 254 residues. For each Cα distance, the value was further encoded through the feature preprocessing method outlined in the methods. For feature transformation, 10.0Å was chosen as the threshold. The ivis dimensionality reduction method was applied to extract the collective variables and project the embedding layer onto a 2D surface. The distribution of four states in the ivis result is plotted in Figure 3.2A. The plot revealed that the transient dark state partially overlaps with the native dark state and the transient light state. The large region of transient dark state distribution is mainly because of the enhanced dynamics caused by the formation of the covalent bond. The distribution of the native light state is divided into two separate regions. The distribution of the transient light state covers a large area, and overlaps with both regions of the native light state distribution.

Markov state model is based on the clustering results on the reduced dimension projected by ivis framework. To construct MSMs, MiniBatch k-means clustering method was applied to partition the distribution of protein simulations in the 2D region into 300 microstates. The top 20 relaxation timescales calculated by different MSMs with different lag times are plotted in Figure 3.2B. The implied timescale converges after 40 ns, which was chosen as the lag time for MSM. The number of macrostates depends on the gap between the timescales, and a total of 10 macrostates were chosen to divide the reduced protein distribution into kinet-
ically separated macrospaces. For each microstate, the corresponding labels of macrostates were determined by the PCCA method, which is based on the eigenfunction of the transition probability matrix in MSM. The resulting macrostates with their associated transition probabilities are illustrated in Figure 3.3. Two dark states and two light states are divided into 4 and 6 macrostates, respectively. Macrostates (states 1, 2, 3 and 10) are in the area of the native dark state and the transient dark state. Based on the similarity to the crystal dark state structure, macrostate 2 was considered as the native dark state. State 9 was recognized as the native light state using the same method. Other macrostates were considered as intermediate states. The low transition possibilities starting from macrostate 2 and 9 to adjacent macrostates indicate the stability of both the dark and light states. On the contrary, it is more likely for protein to shift between intermediate states. Two representative structures in the transient dark and transient light states are illustrated in Figure 3.4. Both A’α and Jα helices in the representative conformation of macrostate 3 (transient dark state, Figure 3.4A) move further from the LOV domain comparing to the native dark state (transparent grey structure in Figure 3.4A). These differences agree with the experimental finding that Jα helix interacts with the LOV core through hydrogen bonding between Gln365 and Cys316 as well as Tyr357 and Gln330 in the native dark state. [66] In the light state, the
Figure 3.3: Macrostates in MSM with transition probability. Based on the transition probabilities, states 2 and 9 were considered as the native dark and native light states among macrostates, respectively. Other macrostates were treated as intermediate states.

The hydrogen bond between Gln365 and Cys316 is broken after the formation of photo-induced covalent bond between FMN and Cys287, leading to the release of $J\alpha$ helix from the LOV core. The $A'\alpha$ helix also interacts with the LOV core via a hinge region (Ala248, Glu249, Glu250 and Gln251) and covers a hydrophobic patch (as the back of $A'\alpha$ helix) in the native dark state. Due to the change of $A'\alpha$ helix orientation, the back of this helical structure as the hydrophobic patch is exposed in the light state. The protein structure of macrostate 5
Figure 3.4: Representative conformations in the transient dark and light states. Monomer structures in (A) macrostate 3 from the transient dark state, and (B) macrostate 5 from the transient light state. Corresponding structures in native dark and native light states are shown in transparent grey color.

(transient light state, Figure 3.4B) is similar to the structure of native light state due to the stabilizing interaction within the dimer structure.

3.3. Key Residues Identified by One-vs-One Random Forest

In order to extract the key residues that play a vital role in AuLOV allostery, supervised machine learning models were applied to explore the structural differences among macrostates. Here, pairwise Cα distances were chosen as the translation and rotation invariant collective variables for the description of protein structures in the simulations. For each simulation, frames were saved for every 100 picoseconds (ps), resulting in 10,000 frames for every 1 µs MD trajectory. Accordingly, 120,000 samples with 32,131 features were extracted from the simulated trajectories. Each frame was labeled based on the macrostate results. Random forest and one-vs-one random forest models were applied to distinguish the intrinsic conformational differences among macrostates. Training scores and testing scores were plotted in Figure 3.5. The testing accuracy was 93.5% in the random forest model at depth
Figure 3.5: Tree-based models for macrostate classification. (A) Prediction accuracy of the random forest model with different tree depths; (B) Prediction accuracy of the one-vs-one random forest model with different tree depths; (C) Accumulated explained importance of the OvO random forest model in 8 tree depth with regard to the number of features. The top 550 features account for 90.2% of the overall importance.

9 and 94.5% in the OvO random forest model at depth 8. The high classification accuracy indicated that the two tree-based models were able to capture the characteristics of protein configuration of each macrostate using pairwise Cα distances.

The advantage of the tree-based models comes from the ability to quantitatively evaluate the contribution of each feature in classification model through the value of feature importance. Superior to random forest model, one-vs-one random forest model was applied to compute the feature importance for any two different macrostate pairs by conducting a random forest classification between these two specific macrostates. Therefore, for any two different macrostates, one distinct random forest estimator was built. A combination of 10 * 9/2 = 45 basic random forest classifiers were constructed for the pairwise macrostates classifications. Accumulated feature importance of one-vs-one random forest at depth 8 was plotted in Figure 3.5C. Overall, this method is an effective model, in which the top 550 features out of 32,131 features account for 90.2% of the overall feature importance.

Those Cα distances related to two residues located on different chains are named as cross-monomer features. These cross-monomer features accounts for 59.77% of the overall importance. Therefore, the Cα distances within the same chain accounts for 40.23% of the
Table 3.1: Top 20 residues identified by OvO random forest.

<table>
<thead>
<tr>
<th>Residue ID</th>
<th>Residue Type</th>
<th>Residue ID</th>
<th>Residue Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>GLN</td>
<td>349</td>
<td>VAL</td>
</tr>
<tr>
<td>245</td>
<td>LEU</td>
<td>247</td>
<td>THR</td>
</tr>
<tr>
<td>252</td>
<td>PHE</td>
<td>248</td>
<td>ALA</td>
</tr>
<tr>
<td>351</td>
<td>CYS</td>
<td>249</td>
<td>GLN</td>
</tr>
<tr>
<td>244</td>
<td>ALA</td>
<td>331</td>
<td>PHE</td>
</tr>
<tr>
<td>312</td>
<td>ASP</td>
<td>314</td>
<td>SER</td>
</tr>
<tr>
<td>246</td>
<td>GLN</td>
<td>334</td>
<td>ALA</td>
</tr>
<tr>
<td>268</td>
<td>SER</td>
<td>313</td>
<td>MET</td>
</tr>
<tr>
<td>350</td>
<td>GLN</td>
<td>336</td>
<td>LEU</td>
</tr>
<tr>
<td>335</td>
<td>ALA</td>
<td>251</td>
<td>ASN</td>
</tr>
</tbody>
</table>

* Experimentally confirmed important residues are shown in bold font.

overall importance. This shows that the OvO random forest can capture the structural changes within each monomer, as well as the relative motions between monomers.

In order to identify key residues based on the results of the OvO random forest model, the feature importance value of each Cα distance was added and accumulated to the two related individual residues. The top 20 residues are listed in Table 3.1. Among the identified residues, several have been experimentally confirmed to be important to AuLOV allostery and are shown in bold font. Residues Met313, Phe331, and Cys351 are found to undergo changes in orientation. Ala248, Gln249, Gln250, and Asn251 are residues linking the A’α helix to the Aβ strand that are important for signal transduction. [66] Gln350 was also identified as essential for signal transduction in LOV domains, where it either undergoes a Gln-flip process in response to N5 protonation [78] or undergoes rotation between exposed and buried conformations [79] to relay the signal from the flavin active site to N- or C-terminal components. We also identify Phe252 as important for allostery. Notably, Phe252 was found by HDX-MS to be important in the destabilization of the A’α helix that is coupled to conformational changes in Bβ strand and Cα helix. [66] Therefore, the OvO random forest can successfully identify important residues reported in experimental results.
Table 3.2: Accumulated feature importance of secondary structures in AuLOV.

<table>
<thead>
<tr>
<th>Secondary structure</th>
<th>Importance percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A'α</td>
<td>15.17%</td>
</tr>
<tr>
<td>Aβ</td>
<td>7.12%</td>
</tr>
<tr>
<td>Bβ</td>
<td>2.28%</td>
</tr>
<tr>
<td>Cα</td>
<td>9.12%</td>
</tr>
<tr>
<td>Dα</td>
<td>0.70%</td>
</tr>
<tr>
<td>Eα</td>
<td>0.05%</td>
</tr>
<tr>
<td>Fα</td>
<td>2.14%</td>
</tr>
<tr>
<td>Gβ</td>
<td>6.43%</td>
</tr>
<tr>
<td>Hβ</td>
<td>14.13%</td>
</tr>
<tr>
<td>Iβ</td>
<td>13.46%</td>
</tr>
<tr>
<td>Jα</td>
<td>8.04%</td>
</tr>
<tr>
<td>Linkers</td>
<td>21.36%</td>
</tr>
</tbody>
</table>

The residue importance can be accumulated to the protein’s secondary structures and the results were shown in Table 3.2. A’α and Jα helices account for 15.17% and 8.04% of the overall importance, respectively. The importance of Cα helix and linkers in AuLOV are also significant at 9.12% and 21.36%, respectively.

3.4. Machine Learning-Based Community Analysis of Protein Communities

To explore the significance of different protein secondary structures, machine learning based community analysis was applied to split the protein structure into communities. This analysis was developed to divide residues into several communities (referred to as ML communities) so that the feature importance for pairwise Cα distances across different communities is maximum, while the feature importance within each community is minimum.

The relationship between the feature importance for pairwise Cα distances within ML communities and the number of ML communities are plotted in Figure 3.6A. Applying an elbow criterion, four ML communities were selected with the total feature importance within each ML community accounting for 0.50% and the total feature importance among ML communities
communities accounting for 99.50%. Therefore, the changes among ML communities account for the dominant majority of the overall feature importance and are able to explain the changes between different communities. The changes within each ML community are ignored due to the negligible importance. By applying ML based community analysis, dynamics in each protein structure can be attributed to the changes among partitioned ML communities.

The distribution of different communities, with a complete partition result corresponding to protein secondary structure, is shown in Figure 3.6B. Commu. A (blue) includes most of A’α helix and Aβ strand, Commu. B (orange) includes Jα helix with part of Gβ and Hβ strands on the LOV core, Commu. C (red) includes Cα helix, part of Fα helix and linkers. Commu. D (gray) includes part of Fα helix, Gβ, Hβ and Iβ strands.

The machine learning based community analysis offered additional information based on the selected four ML communities and the corresponding different regions in the protein structure during simulation. The accumulated overall feature importance among each ML community pair is listed in Table 3.3. Correlations between Commu. A, Commu. B and the rest of the protein accounted for 82.99% of the total feature importance. This is not
Table 3.3: Accumulated feature importance between each ML community pair.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Commu. A</td>
<td>0.12%</td>
<td>13.58%</td>
<td>25.87%</td>
<td>13.87%</td>
</tr>
<tr>
<td>Commu. B</td>
<td>0.03%</td>
<td>13.17%</td>
<td>16.44%</td>
<td></td>
</tr>
<tr>
<td>Commu. C</td>
<td>0.15%</td>
<td>16.57%</td>
<td></td>
<td>0.20%</td>
</tr>
<tr>
<td>Commu. D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

surprising since the A’α helix in Commu. A and Jα helix in Commu. B are the most distinguishing structures, which undergo significant conformational changes from the native dark state to the native light state. Through the accumulated feature importance of ML communities, A’α and Jα helices are confirmed to convey significant allosteric characteristics. Unexpectedly, the correlation between Commu. C and Commu. D accounts for 16.57% of the total feature importance. Several transitions between adjacent macrostate pairs have significant contribution from Commu. C (Table 3.4). However, for transitions between non-adjacent macrostates, Commu. C accounts for less importance which explains the difference between macrostate pairs.

For those transitions between macrostates where Commu. C accounts for a large component, two promising routes from the dark state to the light state can be identified as: 1) State 2 → 3 → 5 → 7 → 6 → 9 and 2) State 2 → 3 → 5 → 6 → 9. These two proposed pathways lead to a hypothesis that Commu. C is important in propagating allosteric perturbations.

To estimate the probability of the two identified channels which include significant Comm. C contribution, the transition path theory was employed to generate an ensemble of pathways to estimate the probability of every pathway from State 2 (native dark state) to State 9 (native light state). A total of 3,151 pathways were generated and divided as 212 distinct channels connecting these two states. The probability of each channel was calculated based on the net flux from the initial state to the target state. Overall, the probability of top 10
Table 3.4: The changes of Commu. A, Commu. B and Commu. C during transitions between macrostates.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>State 2 (Dark) → State 3</td>
<td>7.26%</td>
<td>26.69%</td>
<td>4.01%</td>
</tr>
<tr>
<td>State 2 → State 10</td>
<td>6.05%</td>
<td>18.51%</td>
<td>3.03%</td>
</tr>
<tr>
<td>State 3 → State 5</td>
<td>16.91%</td>
<td>12.74%</td>
<td>7.68%</td>
</tr>
<tr>
<td>State 5 → State 7</td>
<td>7.44%</td>
<td>2.04%</td>
<td>18.41%</td>
</tr>
<tr>
<td>State 5 → State 4</td>
<td>9.55%</td>
<td>6.40%</td>
<td>3.64%</td>
</tr>
<tr>
<td>State 5 → State 6</td>
<td>10.96%</td>
<td>6.98%</td>
<td>16.27%</td>
</tr>
<tr>
<td>State 7 → State 6</td>
<td>11.22%</td>
<td>3.97%</td>
<td>8.86%</td>
</tr>
<tr>
<td>State 6 → State 9 (Light)</td>
<td>12.35%</td>
<td>10.56%</td>
<td>19.13%</td>
</tr>
<tr>
<td>Non-Adjacent macrostates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State 10 → State 9</td>
<td>21.39%</td>
<td>23.63%</td>
<td>0.23%</td>
</tr>
<tr>
<td>State 10 → State 6</td>
<td>26.97%</td>
<td>15.11%</td>
<td>2.26%</td>
</tr>
</tbody>
</table>

* State-transitions with large Commu. C component are shown in bold font.

channels is listed in the Table 3.5. The population of these 10 channels account for 80.0% of the total pathway population.

Among all 212 channels, the two identified channels 2-3-5-7-6-9 and 2-3-5-6-9 are the top 2 populated channels with 28.8% and 25.6% of overall probability, respectively. The sum of contributions from these top two channels accounts for 54.4% contributions, which is significant compared to all other pathways, suggesting the importance of Commu. C movement during the allosteric process. The first channel is more probable than the second one. This agrees with the observation that the transition probability from 5 to 7 (9.5%) as one step in the first channel is greater than that from 5 to 6 (1.6%) as one step in the second channel. Interestingly, the ML based community analysis reveals higher contribution from the Commu. C to the transition between states 5 and 7 than that between states 5 and 6.

Different communities account for different importance in each macrostate transition. To better show the trend of components in Commu. A, Commu. B and Commu. C with regard to Commu. D, the change of importance along the two proposed paths is plotted in Figure
Table 3.5: The probability of top 10 channels simulated using transition path theory.

<table>
<thead>
<tr>
<th>Channels</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 3, 5, 7, 6, 9</td>
<td>28.8%</td>
</tr>
<tr>
<td>2, 3, 5, 6, 9</td>
<td>25.6%</td>
</tr>
<tr>
<td>2, 10, 5, 7, 6, 9</td>
<td>5.4%</td>
</tr>
<tr>
<td>2, 3, 7, 6, 9</td>
<td>4.7%</td>
</tr>
<tr>
<td>2, 3, 5, 8, 5, 7, 6, 9</td>
<td>3.9%</td>
</tr>
<tr>
<td>2, 3, 4, 8, 5, 7, 6, 9</td>
<td>3.0%</td>
</tr>
<tr>
<td>2, 3, 8, 5, 7, 6, 9</td>
<td>2.9%</td>
</tr>
<tr>
<td>2, 10, 5, 6, 9</td>
<td>2.3%</td>
</tr>
<tr>
<td>2, 1, 3, 5, 7, 6, 9</td>
<td>2.1%</td>
</tr>
<tr>
<td>2, 10, 5, 8, 5, 7, 6, 9</td>
<td>1.3%</td>
</tr>
<tr>
<td>Top 10 channels</td>
<td>80.0%</td>
</tr>
</tbody>
</table>

3.7. Two paths share similar characteristics: 1) Commu. A accounts for little importance at the beginning of allostery process while the contribution goes up in later transitions; 2) Commu. B starts with high importance and decreases drastically after the first transition; 3) Commu. C is more important at the end of allostery process.

3.5. Discussion

PtAu1a is an allosteric protein that undergoes a series of conformational changes upon light activation beginning with the formation of covalent bond between Cys287 and FMN. [80] This computational study of AuLOV is integrated with MD simulations and other computational methods to provide quantitative analysis of the dynamics and importance of residues with regard to the overall allosteric process. While there is extensive research on the regulatory role of Jα helix and dimerization controlling A'α helix, a detailed mechanism of allostery with signal transmission route still needs scrutinization.

Signal transduction in LOV domain containing proteins typically involves coupling of adduct formation to conformational changes in the N- and C-termini via propagation across
Central to this signal transduction are key residues within the Iβ strand that enables its coupling with the Jα helix and interaction with Α’α helix in the dark state, specifically the residue equivalent to Gln350 that is essential for LOV signal transduction. In AuLOV, several additional light-induced rotamers (Met313, Leu317, Leu331, Leu333, and Cys351) were observed on the β-sheet surface. Here, through the accumulated residue importance in the one-vs-one random forest model, we successfully identified Met313, Leu331, and Leu351 as being important in differentiating allosteric changes in AuLOV. In our models, these residues contribute to conformational changes linking the β-sheet surface to Α’α helix through Gln350. We note that our computational methods mirror those identified experimentally where Α’α helix contributes to the dynamic stability of the dark state by the interaction with LOV core through a hinge region. The hinge region consists of four conserved residues (Ala248, Gln249, Gln250, and Asn251), which were also found to be important via our approach (Table 3.1). Overall, the strong correlations between previous experimental results, and our Markov state model and OvO random forest analysis, confirm our methodology as being able to discern allosteric pathways in AuLOV.
Most proteins undergo allosteric process within a long timescale from milliseconds to seconds, including AuLOV, making it difficult to collect sufficiently long trajectories. Markov state model addresses this difficulty by extracting the slowest motion and long timescale information from limited simulations. However, while the slowest dynamical processes are often involved in protein allostery and are assumed to be the process of interest, fast-moving flanking helices or side chain rotations could play significant role in protein allostery. Due to their short timescales, these motions may not be represented well in the MSM. Although there are some studies focusing on fast protein motions and their relations with slow motions, the functions of fast motions in protein allostery remains elusive and requires more studies. Regarding the allostery of AuLOV, the kinetics between dark- and light-states are beyond scale of minutes. Therefore, sub-ns protein local motions are unlikely to be determinant factors in AuLOV allosteric mechanism and are not the focus of the present study.

Although chain A and chain B in AuLOV are dynamically identical in the dark state, the A’α helices of the two chains differ in conformations upon dimerization. Our simulation results confirmed the differences between these two chains through a comparison of RMSF values. The RMSF results reflect that A’α helix in chain A is more dynamically active than that in chain B. The asymmetrical property in A’α helix could originate from either the interaction between A’α and Jα helices on different chains or the asymmetrical conformational change, thus requiring further detailed study.

ML based community analysis used in this study provided an approach to partition protein conformation into communities based on the feature importance of pairwise Cα distances. Through this analysis, three important communities were identified. Commu. A containing A’α helix and Commu. B containing Jα helix were expected to account for great contribution, since these two helices undergo notably conformational changes upon light activation (Table 3.2). The Cα and Fα helices stand out as Commu. C, and surprisingly
provided additional information for allosteric process. Commu. C accounts for great importance in adjacent transitions between macrostates and accounts for less importance in nonadjacent transitions compared with Commu. A and B.

Transition path simulations further validated the important allosteric function of Commu. C. For all possible transition pathways found by TPT, the top 2 channels are those with large Commu. C components and together constitute over 50% of the overall possibility. Although Commu. C consists of two helices as Cα and Fα, these two helices are not equally important. The allosteric role of Fα helix should be evaluated with caution since its accumulated feature importance is relatively low (Table 3.2), and the importance in Commu. D, which also includes part of Fα helix, is the least important community. Because Cα helix is important in both OvO random forest result and ML based community analysis, it is reasonable to conclude that Cα helix may play an important role in controlling AuLOV allostery. Moreover, Commu. C also includes several linking residues that account for a large portion of the overall importance, indicating the indispensable role of linkers in the allosteric process as reported in previous studies. [16, 88]

Examination of the two most probable channels linking conformational changes through the identified communities can allow construction of allosteric paths (Figure 3.7). In this study we identify that the Jα helix is fundamental in the early stage of AuLOV allostery, followed by changes in the A’α helix in later stages. In the first transition step from macrostate 2 → macrostate 3, Commu. B accounts for a large component compared with Commu. A, indicating the importance of Jα helix in the initial stage of allostery. As the allosteric perturbation propagates, the importance of Commu. B decreases and Commu. A becomes the more significant region. This important shift implied and confirmed the experimental finding that, after initial Cys287-FMN covalent bond formation, the first response of the protein structure is the undocking of the Jα helix, which is essential to the release of A’α helix. [79, 89, 90] The rising importance of Commu. C, together with the transition path
theory results, suggests that Commu. C, especially C$\alpha$ helix and linkers, is vital in the allostery process and should be investigated further.

3.6. Conclusion

The LOV protein PtAu1a is a member of Aureochrome family that binds DNA upon blue-light activation. [29] Studies of the LOV domain with N- and C-terminal helices indicate that in the absence of light it exists as monomeric units; upon blue-light absorption, cysteinyl-flavin bond formation triggers a global conformational change that ultimately results in the dimerization of the LOV domains. In the present study, the protein dynamics of AuLOV with N- and C-terminal helices is simulated using MD simulations and analyzed using a series of computational methods. We quantified the differences of A$'$$\alpha$ and J$\alpha$ helices dynamics in four functional states and the importance of each residue in the two chains with regard to the protein allostery process. Key residues in overall structural changes identified by an OvO random forest agree with the results reported in other experimental work. The Markov state model, combined with transition path theory, studied the importance of protein structures by a machine learning-based community analysis. The functional role of key Commu. C, which includes the C$\alpha$ helix and linkers, is revealed through in-depth analysis as propagating the allosteroic perturbation. Overall, this study quantitatively analyzed the allostery process of AuLOV and linked the macroscopic conformational change to residue level importance. Our results provided new opportunities for a detailed mechanism explanation and offered further opportunities for the research of PtAu1a as an optogenetic tool. Future studies can facilitate our understanding of global protein conformational changes in the context of full-length PtAu1a.
4.1. Introduction

Many allosteric site prediction methods have been developed based on molecular dynamics (MD) simulations, [91] normal mode analysis, [92] two-state Gō models, [93] and machine learning (ML) models. [94–96]. Among the existing methods, AllositePro, [97] AlloPred, [98] SPACER, [99] and PARS [100] are available as web servers or open-source packages. These previous studies have shown that it is promising to identify allosteric sites by combining static pocket features with protein dynamics. In these studies, static features are calculated by site descriptors describing physical properties of protein pockets, while the protein dynamics are extracted by MD simulation or perturbation.

Machine learning methods have been shown to be superior in the classification of protein pockets. For example, Allosite [96] and AlloPred [98] used support vector machine (SVM) [101] with optimized features. Chen et al. [102] used random forest (RF) [40] to construct a three-way predictive model. With the development of ML, more advanced models have been developed and can contribute to the allosteric site classification. [103] eXtreme gradient boosting is one of the most powerful machine learning techniques in classification. It is an implementation of the gradient boosting algorithm with regularized terms to reduce overfitting. Compared with SVM and RF, XGBoost achieved superior predictive performance in the protein-protein interactions [104] and hot spots [105].

Though physical properties are largely contained in many methods, topological information is largely ignored and is considered important in classifying pockets. In order to
explore the geometry features, an atomic graph is constructed for each pocket. Atoms are treated as nodes and the pairwise bond distances are calculated as edges. [94] Graph convolutional neural networks (GCNNs), [45] a popular concept in deep learning, have been applied in biological-related predictions, ranging from chemical reactions, [106] molecular properties, [107] to drug-target interactions. [108]

In this study, protein pockets are predicted using an ensemble learning method, which combines the results of XGBoost and GCNN. This model can learn both physical properties and topology information of allosteric pockets and has been proven to be superior to the single XGBoost and GCNN models. Various performance indicators validated the success of this ensemble learning method compared with previous methods.

4.2. Prediction Performance of XGBoost

XGBoost model can overcome the limitation of data imbalance by controlling the weight difference between negative labels and positive labels. This parameter was fine-tuned along with the maximum depth of trees. The results are plotted in Figure 4.1. Two sets of parameters reached high F1 scores, and both were selected in the final model. The final
Table 4.1: Evaluation and performance comparison of different models. The average values and standard errors of 6 indicators are calculated in 10 independent runs. The ensemble learning method can achieve better performance compared to single XGBoost and GCNN models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
<th>Specificity</th>
<th>F1 score</th>
<th>ROC AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XGBoost</td>
<td>0.969</td>
<td>0.799</td>
<td>0.732</td>
<td>0.982</td>
<td>0.764</td>
<td>0.897</td>
</tr>
<tr>
<td></td>
<td>± 0.002&lt;sup&gt;a&lt;/sup&gt; ± 0.023 ± 0.030 ± 0.003 ± 0.016 ± 0.016</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCNN</td>
<td>0.923</td>
<td>0.604</td>
<td>0.427</td>
<td>0.943</td>
<td>0.500</td>
<td>0.832</td>
</tr>
<tr>
<td></td>
<td>± 0.006 ± 0.023 ± 0.046 ± 0.007 ± 0.031 ± 0.015</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model ensembling</td>
<td>0.974</td>
<td>0.847</td>
<td>0.726</td>
<td>0.980</td>
<td>0.782</td>
<td>0.914</td>
</tr>
<tr>
<td></td>
<td>± 0.010 ± 0.095 ± 0.085 ± 0.013 ± 0.072 ± 0.018</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allosite [96]</td>
<td>0.962</td>
<td>0.852</td>
<td>0.688</td>
<td>0.970</td>
<td>0.761</td>
<td>0.911</td>
</tr>
</tbody>
</table>

<sup>a</sup> Standard error (SE) = Standard deviation (SD) / √sample size.

Table 4.2: Probabilities of predicting allosteric sites in the top 3 positions. Ensemble learning method can rank an allosteric site in the top 3 positions with a probability of 84.9%, which is higher than previous results.

<table>
<thead>
<tr>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARS [100]</td>
<td>44%</td>
<td>62%</td>
</tr>
<tr>
<td>AlloPred [98]</td>
<td>57.5%</td>
<td>70.0%</td>
</tr>
<tr>
<td>Ensemble learning</td>
<td>60.7%</td>
<td>81.6%</td>
</tr>
</tbody>
</table>

<sup>a</sup> Not available in the reported results.

XGBoost model is composed of two models, each with one set of parameters. The results in any given pocket are the averaged results predicted by these two models.

The results of the fine-tuned XGBoost model are listed in Table 4.1. Compared with the reference results, XGBoost model exhibited higher accuracy, precision, specificity, and ROC AUC values with comparable results in recall and F1 scores. Therefore, XGBoost model performs well in allosteric site prediction.

4.3. Prediction Performance of GCNN

Unlike XGBoost, GCNN models suffer from imbalanced dataset. To address this problem, the ratio between negative labels and positive labels was evaluated first. The results are
Figure 4.2: Fine-tuned results of ratio and distance threshold parameters in GCNN model. (A) The ratio between number of negative labels and number of positive labels was fine-tuned. Ratio of 2 is considered reaching a balance between recall and precision. (B) The atomic distance threshold was fine-tuned from 7 to 11 Å. There is no significant increase in F1 score after 10 Å, which is selected as the distance cutoff. For each parameter value, GCNN was run 10 times independently.

plotted in Figure 4.2A. A ratio of 2 (number of negative labels : number of positive labels = 2 : 1) was selected. The distance threshold was further fine-tuned, and the results are plotted in Figure 4.2B. 10 Å was selected as the distance cutoff when constructing atomic graphs.

The results of the fine-tuned GCNN model with 10 independent runs are listed in Table 4.1. Compared with XGBoost, GCNNs are less effective in classifying allosteric sites. However, it is expected that combining XGBoost and GCNN will result in better performance than either model.

4.4. Prediction Performance of Ensemble Model

The ensemble learning model is composed of both XGBoost model and GCNN model. For a given pocket, physical properties are calculated and fed into the XGBoost model; a representative atomic graph is fed into the GCNN model. The final result is calculated as the averaged probability of these two models. This final model contains both the physical properties and topological features of protein pockets. The combined results are listed in Table 4.1. Compared with the XGBoost model, model ensembling leads to a 6.00% increase
Figure 4.3: Prediction results of two examples not included in the training set: (A) Dynamics-driven PDZ2 protein in bound states (PDB ID 3LNY); (B) The LOV domain of conformational-driven Phaeodactylum tricornutum Aureochrome 1a in dark state (PDB ID: 5DKK). Red regions are the most probable pockets in the predicted results with probabilities of (A) 45.14% and (B) 89.46% and are also the true allosteric sites.

in recall, a 0.82% decrease in precision, and a 2.89% increase in F1 score. The AUC ROC value also had a 1.89% increase.

For each protein, the identified pockets are ranked based on the predicted probabilities. Overall, 60.7% of allosteric pockets are predicted as the first position, while 81.6% among the top 2 and 84.9% among the top 3. In other words, if a pocket is an allosteric pocket, there is a probability of 84.9% that it can be predicted in the top 3 among all detected pockets in the same protein. The prediction results, together with values reported in other studies, are listed in Table 4.2. It should be noted that the types and amounts of proteins in the test set are different from each other.

4.5. Novel Allosteric Sites Prediction

To test this ensemble learning method, two proteins not in the dataset (Table 6.2) were used. The predicted allosteric pockets of these two proteins are illustrated in Figure 4.3. These two proteins represent two different types of allosteric proteins. The second PDZ
Figure 4.4: Web server workflow. User can upload either a PDB ID or a PDB file in the web server. FPocket is used to detect pockets. For each pocket, physical properties are calculated and predicted using a pretrained XGBoost model; while an atomic graph is constructed and fed into a pretrained GCNN model. The final probability is given by averaging results from both models.

domain (PDZ2) is a dynamics-driven protein in human PTP1E protein which undergoes allosteric process upon binding with peptides. [43] The light-oxygen-voltage (LOV) domain of Phaeodactylum tricornutum Aureochrome 1a (AuLOV) is a conformational-driven allosteric protein. [66] AuLOV is a monomer in the dark state and undergoes dimerization upon blue light perturbation. [39] In both cases, our prediction model ranks the allosteric sites as the top 1 with probabilities of 45.14% and 89.46%, respectively. This indicates that this model is capable of predicting both dynamics-driven and conformational-driven allosteric proteins. Apparently, the probability for the dynamics-driven allosteric protein is much smaller than the one of the conformational-driven allosteric protein, which is not unexpected.

4.6. Web Server and Command Line Usage

A web server based on the allosteric prediction method developed in this study is implemented using a Python web framework, Django. JSmol is a JavaScript implementation of
Figure 4.5: PASSer web server pages. (A) Users can either submit a PDB ID or a PDB file in the home page. (B) Predicted top 3 pockets are summarized in a table with corresponding probabilities and pocket residues. (C) Protein structures and pocket sites are displayed in an interactive window.

Web pages are rendered using Bootstrap. This server is named as Protein Allosteric Sites Server (PASSer). A workflow of PASSer is outlined in Figure 4.4.

An example of input and output of PASSer is displayed in Figure 4.5. Users can submit a PDB ID if available or upload a custom PDB file as shown in Figure 4.5A. By default, all chains in the protein are analyzed. Prediction results are displayed as two parts: top 3 pockets with the highest probability rendered with the protein structure (Figure 4.5B) and their probabilities (Figure 4.5C). For each pocket, the corresponding residues can be retrieved by clicking the “Show Residues” texts. Protein structure is visualized using JSmol. Each pocket is either displayed upon clicking its “Load pocket” icon or hidden by clicking its “Hide pocket” or overall “Reset” icons.

A command line interface (CLI) is provided to facilitate potential developments. Similar with the web usage, this CLI can take either a PDB ID or a local PDB file for predictions.
4.7. Discussion

The quality of the dataset used for training is critical. Classification models often fail in prediction performance, and lack the generalization with poorly collected datasets, such as insufficient training data or high similarity between structures. ASD is an online database that provides allosteric proteins and sites with high resolution, bringing opportunities for allosteric site prediction. There are other databases, such as ASBench [110] and sc-PDB [111], which can also be used to improve data quality and model performance.

In order to predict allosteric sites, proper pockets need to be identified on the surface of proteins. Several open-source pocket detection software has been developed. Previous results [112] have shown that, the geometry-based algorithm FPocket is superior compared with other methods, such as PASS [113] and LIGSITEcsc [114], and can cover known allosteric sites. In addition, FPocket is under active development and can be integrated with other methods to build a complete pipeline for site prediction.

Several computational methods have been developed for allosteric site prediction over the past few years. Due to the fast development of machine learning methods, many models integrate ML methods, such as support vector machine and random forest, for accurate predictions [96, 102]. One critical issue is that many ML models fail when dealing with imbalanced datasets [115]: in the allosteric site database, negative samples account for a majority of the dataset with a limited proportion of positive samples. Undersampling is one way to rebalance the dataset. For example, Allosite discarded some negative labels and used a ratio of 1:4 between positive and negative labels. However, undersampling leads to insufficient usage of the overall dataset. XGBoost as a gradient boosting method overcomes this data imbalance by controlling the relative weights between classes so that the dataset can be fully used. Various performance indicators, as listed in Table 4.1, validated the effectiveness of XGBoost for the identification of allosteric sites.
It is worth noting that some features are highly correlated. This collinearity should be addressed in regression models, which reduces the model precision and thus weakens prediction results. In contrast, XGBoost is free from this problem. When several features are found to be highly correlated to each other, XGBoost will choose one of these features. Therefore, collinearity does not affect the prediction results. Nevertheless, collinearity influences model interpretation, such as feature importance, which should be considered with caution.

Physical properties have been widely used to describe the characteristics of pockets. These features are normally calculated using static protein structures. To probe the dynamical behavior of pockets, normal mode analysis and MD simulations are normally conducted [92,116]. Results from these methods have shown that models can achieve satisfactory performance through the combination of both static features and protein dynamics. However, pocket geometry is often ignored, which could play an important role in prediction. Therefore, a graph convolutional neural network is applied to retain the topological information. Specifically, pockets are represented as undirected graphs at atomic level, and GCNN is designed to learn the local connectivity among atoms. While a previous study [94] included energy-weighted covalent and weak bonds in the prediction of allosteric sites, it should be noted here that: (1) it is assumed that the physical properties are implicitly retained in the site descriptors and GCNN only studies the node degree; (2) GCNN does not require any a priori information about the location of active sites. The ensemble learning method, consisting of GCNN and XGBoost, exhibited higher performance compared with single models.

4.8. Conclusion

The proposed ensemble learning method involves XGBoost and GCNN, which can learn both the physical properties and topology of protein pockets. The results are comparable with previous studies and have a higher percentage of ranking allosteric sites at top positions. This ensemble learning method, embedded in PASSer, can help exploration on protein allostery and drug development.
5.1. Introduction

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), [117] to expand the landscape. However, some domain knowledge is required to define the essential coordinates, e.g., collective variables (CVs). [118] 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 or dimensionality reduction methods. [119]

The advancement of deep learning provides an alternative approach for enhanced sampling. Several studies have demonstrated the success of both autoencoders (AEs) and variational autoencoders (VAEs) in their applications to enhanced sampling. [120–123] 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. [124]

In this study, we proved the success of variational autoencoders in enhanced 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. [125] 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.

5.2. Learned Protein Conformational Space of ADK Protein

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
Table 5.1: Residue numbers in the centers of mass of heads and tails in the four vectors as shown in figure 5.1.

<table>
<thead>
<tr>
<th></th>
<th>1AKE-4AKE</th>
<th></th>
<th>1DVR</th>
<th></th>
<th>2AK3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tail</td>
<td>Head</td>
<td>Tail</td>
<td>Head</td>
<td>Tail</td>
<td>Head</td>
</tr>
<tr>
<td>V3</td>
<td>115-125</td>
<td>179-185</td>
<td>124-134</td>
<td>188-194</td>
<td>119-129</td>
<td>183-189</td>
</tr>
</tbody>
</table>

NMP domain. According to the previous research, [126] 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 5.1.

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 5.2A-B. The four characterizing vectors were also calculated and plotted as a 2D angle map in Figure 5.2C. 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. [127–129] 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.
Figure 5.2: 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 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 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 5.2.

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
Table 5.2: 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</th>
<th>Latent space</th>
<th>Decoder</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>128</td>
<td></td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>512,32</td>
<td></td>
<td>32,512</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4980</td>
<td>3</td>
<td>1024,128,16</td>
<td>2</td>
<td>16,128,1024</td>
<td>4980</td>
</tr>
<tr>
<td>4</td>
<td>1024,256,64,16</td>
<td></td>
<td>16,64,256,1024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2048,512,128,32, 8</td>
<td></td>
<td>8,32,128,512,2048</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

data. Each model was trained three times independently and the mean value of each metric was calculated. The results of performance assessment metrics are plotted in Figure 5.3.

In terms of the encoder (Figure 5.3A-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 5.3C-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 5.4. The mean RMSD between the training and decoded structures is 1.03 Å. The learned latent space is plotted in Figure 5.5A. 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 5.5C-F with the NMP-CORE and LID-CORE angles plotted in Figure 5.5B.
Figure 5.3: 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.

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 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.
5.3. Additional Simulations From Seed Structures

To further explore the conformational spaces starting from the generated structures, additional 5 ns simulations, following the same procedure as described in the Methods 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 5.6 to show the conformational spaces from the 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
Figure 5.5: 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.

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 5.7. It is shown that the short combined simulations in Figure 5.6B exhibited similar implied timescales as the reference trajectory.

5.4. Discussion

The protein energy landscapes could be divided into many local energy minima which are represented as the metastable conformational states. These conformations are separated by
Figure 5.6: Protein conformational spaces from (A) 50 ns simulation from the closed state and (B) four sets of 5 ns simulation from the closed state, the open state and two generated conformations (point 3 and 4).

free energy barriers that are much higher than $k_B T$. Due to this reason, MD simulations are often trapped in a local minimum for a long simulation time before jumping to another. In this study, we aim to accelerate this inefficient process by directly taking protein structures from the less sampled regions as the initial structures for additional MD simulations. However, protein structures are high-dimensional data with the degrees of freedom as $3N$ in the Cartesian space. Unlike ADK protein which has known intrinsic collective variables (NMP-LID angles) to characterize protein conformational space, most proteins do not have such representations.

To overcome this problem, 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, this leads to an improved ability to learn the movement of covalently bonded atoms. [121]
Figure 5.7: Estimated timescales with different lag times. The subplots (A) and (B) correspond to the trajectories in Figure 5.6 (A) and (B), respectively. Top 8 timescales were selected and each was plotted with 95% confidence interval.

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 several 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 dimensionality reduction technique and these two indicators have been widely used in such tasks. The performance of decoder modules was defined as the resemblance between the training and decoded protein conformations. RMSD and the DOPE score were used to quantify structure and system energy differences, respectively. The DOPE score has been used in the assessment of computationally generated models [133,134]. A good model is expected to have a low DOPE score. We systematically compared the DOPE scores of generated protein conformations in different VAE settings. It is shown that a complicated model architecture with more hidden layers can generate protein conformations with lower DOPE scores, while this is converged after 4 hidden layers. We further compared the RMSD distributions between training and generated protein conformations with MMD and EMD indicators. Under both cases, 4-layer VAE achieved the lowest scores and was considered the best in representing protein conformational spaces.

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 5.6 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.

Through the angle plot (Figure 5.6) and the estimated timescale calculation (Figure 5.7), it is demonstrated that short MD simulations including trajectories starting from the generated conformations in the latent space could achieve the sampling efficiency comparable to a
single long MD simulation. This suggests that iteratively conducting short MD simulations starting from conformations generated in the learned conformational space could serve as an alternative approach to extensive MD simulations.

5.5. 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 which can be used as restarters for additional MD simulations.
6.1. Introduction

Molecular dynamics (MD) simulation has a wide application on the study of protein conformations and dynamics. [39, 120, 135] Recent developments in biocomputing, such as Anton, [25] AMBER, [136] and OpenMM, [26] have enabled the simulation time scale to milliseconds, which promotes the research in sampling protein motions and structure landscapes. [11, 137] However, the time scales of many protein functions exceed the time scales achievable through traditional MD simulations. Moreover, protein sampling suffers from being trapped within local energy minima, proving difficult to escape. [28, 138] As a result, most of the computational time is typically spent in sampling previously visited regions, which hinders the efficient exploration of protein conformational space.

Many enhanced sampling methods have been developed to address this issue. These methods can be classified into two types. In the first type, biasing potentials are introduced to expand the sampling space, such as metadynamics [139, 140] and Gaussian-accelerated MD. [117] In the second type, seed structures are selected as restarts for iterative MD simulations. This is referred to as adaptive sampling, and numerous methods have been proposed that differ in seed selection methods. Markov state models have been applied to cluster conformations into microstates; [119] parallel cascade selection MD (PaCS-MD) [141] and nontargeted PaCS-MD [142] calculate the root-mean-square deviation (RMSD) to select top snapshots; frontier expansion sampling [143] conducts dimensionality reduction with principal component analysis and Gaussian mixture models to select frontier structures;
Recent innovations in deep learning have provided new insights into sampling protein conformational space. [145, 146] Autoencoders (AEs) and variational autoencoders (VAEs) are a class of deep learning models that learn a representation (encoding), which can capture the key features of input data. Several studies have demonstrated the success of AEs and VAEs in their applications to protein conformations and functions. [120, 122, 147, 148] In our previous work, [149] we showed that VAEs are capable of learning a low-dimensional representation (referred to as the latent space) of protein systems. Through a quantitative study, the learned latent space is shown to be able to represent conformational characteristics. This property indicates that the larger differences the two protein conformations have, the farther their corresponding latent points are from each other.

In this study, we proposed a new adaptive sampling method, latent space-assisted adaptive sampling for protein trajectories (LAST), to accelerate the exploration of protein conformational space. Initially, a short MD simulation is conducted starting from the crystal structure. Afterward, the following steps are repeated iteratively until certain criteria are met. First, a VAE is trained using sampled protein conformations. Then, seed structures are selected in the learned latent space. Finally, starting from these selected seed structures, additional simulations are conducted to sample more protein conformations that will be used in the next round. To quantify the performance, we applied LAST on four conformations in two protein systems: two metastable states of *Escherichia coli* adenosine kinase (ADK) and two native states of Vivid (VVD). To better explore the protein conformational space, ADK conformations sampled from the simulation were projected onto its two intrinsic angles, and VVD conformations were projected onto the space using two RMSD values with reference to the two native structures in dark and light states, respectively. These collective variables are unrelated and unknown to the VAE models. Our results showed that seed structures
were consistently located on the boundary of sampled conformational distributions in all four conformations regardless of protein projection methods. We further compared the sampling efficiency among LAST, SDS, and conventional MD (cMD). In both systems, large conformational changes were observed in a shorter time in LAST simulations. To be specific, LAST explored two transition paths toward two stable states, while SDS explored one and cMD neither in the metastable ADK simulations. In VVD simulations, LAST only took one-third of cMD simulation time to discover a similar conformational space.

6.2. Latent Space-Assisted Adaptive Sampling

AST method includes three steps, and its workflow is shown in Figure 6.1. First, a variational autoencoder is trained using all previous simulations. Second, the lowest-probability samples are selected on the latent space and their corresponding protein structures are treated as seeds. Third, additional MD simulations are conducted from seed structures.

VAE Training

In each iteration, some preprocessing procedures are needed. The simulation trajectories are first aligned to the first frame, and heavy atoms are extracted with Cartesian coordinates being expanded as a feature vector (Figure 1A,B). Then, each feature is transformed to a range of 0-1 using min-max linear scaling, which is used to construct a data set for VAE training.

The architecture of the VAE model is shown in Figure 4.4C. In the current study, we design the encoder model being composed of three hidden layers with a size of 512, 128, and 32 and the decoder model with a size of 32, 128, and 512. The number and size of hidden layers can be adjusted based on the size of proteins. The dimension of latent space is set as two for simplicity and ease of visualization.
Figure 6.1: Workflow of LAST method. To begin with, a short MD simulation is conducted from crystal structures. All sampled conformations are stored in a pool. In each round, if there is no increase of the maximum RMSD of the newly sampled conformations in consecutive five rounds, the workflow stops. Otherwise, a VAE model is trained using all conformations in the pool. Then, the seeds are selected on the latent space. For each of the selected seeds, new MD simulations are conducted, and the sampled conformations are stored in the pool.

**Seeds Selection**

Appropriate seed selection method is needed to expedite the sampling of protein conformational space. In LAST, seeds are selected on the two-dimensional learned latent space of VAE, which has two important characteristics to enable an efficient seed selection. First, as demonstrated in our previous work, the distance between two data points on the latent space is meaningful. Two structurally similar proteins have a shorter distance between their corresponding latent vectors. Second, the sampling distribution of latent space in the VAE is similar but does not strictly follow a normal distribution. It is likely that the KL divergence term in the loss function contributes to the normal distribution, and the reconstruction loss component in the loss function may contribute to the deviation from the normal distribution. As for the distribution of the VAE latent space of protein conformations, the most
common protein structures are encoded in the center of the latent space, while structurally
different proteins are encoded on the boundary. In a data distribution, samples with the
lowest probabilities refer to those points that differ significantly from other data. Based on
the above two points, it is reasonable to treat the lowest-probability samples on the latent
space as seeds to accelerate conformational space exploration, as their conformations deviate
from the majority of the sampled ones.

To implement the seed selection method, we propose three improvements to make LAST
computationally efficient:

1. Latent space of VAE is not strictly normal after optimization even though the normality
   is incentivized in the loss function. Therefore, a nonparametric multivariate kernel
density estimator, instead of multivariate normal density function, is used to fit the
   latent space. The estimator is developed in Python statsmodels library. [150]

2. Latent space distribution might be skewed so that the top N lowest-probability samples
   with the smallest probability densities tend to gather on one side of the distribution.
   To avoid the above issue, the cumulative distribution function (CDF) of the fitted
   nonparametric multivariate kernel density estimator on the latent space, instead of
   probability density, is applied to guarantee that samples from both sides of CDF (values
   close to 0 and 1) are equally selected. In this case, the first order of the density estimator
   was accumulated in the latent space.

3. Protein conformations corresponding to the lowest-probability samples can be located
   and selected based on data index. These protein conformations might be similar to
   each other, resulting in sampling repeated conformational space from MD simulations
   starting from these conformations. Thus, to further boost sampling efficiency, we
   require new seed structures to have at least 1 Å RMSD with all previously selected
   seeds.
One example of seeds selection result is shown in Figure 4.4D, where seeds are highlighted in red stars in the latent space visualization.

**Additional MD Simulations**

Short MD simulations are conducted in each round. In the current study, 10 seeds are selected in one round and a 100 ps simulation is done starting from each seed. Thus, the total simulation time in each round is 1 ns. The detail of these simulations is described in Methods.

The above three steps are iteratively done until convergence. Here, we design the convergence criterion by calculating the mean RMSD of Cα atoms with regard to the starting protein structure. The iterative sampling process is terminated once the mean RMSD stops to increase for successive five rounds or reaches the maximum round number.

**6.3. A Comparative Study of LAST, SDS, and cMD**

Four structures of two protein systems (ADK and VVD) were prepared for MD simulations, as described in Methods. For each protein structure, 100 ps of NVT and 200 ps of NPT simulations were conducted. During the iterative process, all previous simulations were aligned to the first frame with Cartesian coordinates of heavy atoms being extracted as a feature vector to represent protein conformation. Afterward, a variational autoencoder model was trained. Ten seed structures were selected with an additional 100 ps simulation starting from each of them. Therefore, each iteration takes a 1 ns simulation time.

ADK protein is composed of a rigid CORE domain, a lid-shaped ATP-binding domain (LID), and an AMP-binding domain (NMP). Many computational studies have shown ADK to carry out large conformational transitions between the closed state to the open state during the ATP-ADP catalyzation process. [151,152] Four vectors that form NMP-CORE and LID-CORE angles, as shown in Figure S1, have been widely used to characterize ADK protein...
Figure 6.2: Structures of (A) ADK and (B) VVD. ADK is composed of a CORE domain, an LID domain, and an NMP domain. LID-CORE and NMP-CORE angles are calculated by four vectors to represent protein conformations. Both proteins are colored at the secondary structure level using ChimeraX.

VVD is a light-oxygen-voltage domain that undergoes global conformational changes upon perturbation. VVD is shown to be flexible in the native light state and relatively stable in the native dark state. [152] ADK and VVD structures are illustrated using ChimeraX [153] (Figure 6.2).

Proper low-dimensional protein representations are needed to evaluate the quality of seed selection. In the current study, ADK protein structure is projected to LID-CORE and NMP-CORE two-dimensional (2D) angle plots. We followed the same residue selection rule to calculate vectors and angles. [149] For the VVD structure, 2D root-mean-square deviation (RMSD) with reference to the native dark and light structures was used to show the sampled protein conformational space.

Both the angle plot in ADK and RMSD plot in VVD were used to display the protein conformation of seed structures (Figure 6.3). In each subplot, seed structures are highlighted
as red stars. In two metastable ADK conformations (Figure 6.3A,B), seed structures are mainly located in the less sampled regions with small or large LID/NMP angles. This indicates that the variational autoencoder can capture the structural differences of protein conformations within the learned latent space. In the native dark and native light VVD conformations (Figure 6.3C,D), seed structures are also shown to be evenly distributed in the boundary of protein conformational space defined by RMSD to two native VVD structures.
To compare the effectiveness of LAST to conventional molecular dynamics simulations, the sampled protein conformational space in each round of the LAST method is displayed together with cMD sampled conformations. Figure 6.4 shows the protein conformations in 1, 5, 10, and 15 ns for both LAST and cMD. It is shown that under the same simulation time, LAST can explore more protein conformations than cMD. Moreover, the trained variational autoencoder can consistently learn a low-dimensional protein representation in the latent space, regardless of the growing number of simulations and changing shape of conformational space and guide MD simulations to explore less sampled regions. In contrast, there are
limited new conformations being explored in cMD simulations from 10 to 15 ns, indicating that it might be trapped in a local energy minimum.

We continued the LAST simulation of ADK until the convergence of LAST. For comparison, both SDS and cMD simulations were conducted under the same simulation time. The sampled protein conformational spaces are shown in Figure 6.5A. The LAST sampling method took 22 iterations (22 ns simulation time) and explored two paths from the metastable state to the two native states. This aligns with the computational finding that ADK protein undergoes conformational transitions between the open and the closed states.36 Moreover, the sampled conformational space in LAST spans in the intermediate regions between the closed and open states, with some coverage in the open state and no coverage in the closed state. Meanwhile, SDS only explored one path toward the open state, and cMD mostly sampled the overlap of LAST and SDS methods. The sampled two transition pathways align well with a previous study, [151] in which a 200 ns AMBER simulation was conducted, showing that the LID-open NMP-closed metastable ADK structure could visit both native open and closed states. The same experimental setting was applied to the open and closed states of ADK protein. While these two states are stable, LAST can still cover the majority of cMD results and sample more conformations when compared with SDS simulations. The sampled conformations in the LID-open NMP-closed metastable ADK structure were also projected using the first two components in PCA. In contrast to Figure 6.5A, the SDS sampled conformations do not fully overlap with LAST. Instead, both methods sampled different conformational regions and are complementary to the cMD results.

There are 120 ADK structures in the PDB. The minimum RMSDs in LAST and SDS produced trajectories that were calculated with reference to each ADK structure. More than two-thirds (84 out of 120) of minimum RMSDs in LAST are less than those in SDS. On average, the minimum RMSD in LAST is 0.07 Å less than that in SDS. These indicate
Figure 6.5: Explored conformational spaces of (A) ADK and (B) VVD proteins using LAST (red dashed line), SDS (blue dotted line), and cMD (black solid line) methods. LAST took 22 and 30 iterations to complete for ADK and VVD proteins, respectively. In each system, SDS and cMD simulations were conducted under the same simulation time. In the ADK conformational space, LAST explored two paths to the open and closed states, while SDS explored one path toward the open state.

that the LAST method is comparable to SDS and allows the structural integrity of protein to be reasonably maintained.

For the VVD system, LAST simulation took 30 iterations (30 ns simulation time) to converge. The conformational space is illustrated in Figure 6.5B. SDS and LAST methods sampled similar conformational spaces and both covered a majority of cMD sample regions. To compare the efficiency of LAST and cMD methods, this cMD simulation was continued while this 2D RMSD map was being monitored. It took 100 ns simulation time for cMD simulation to have a similar space shape to LAST. In terms of the MD simulation time, LAST was three times faster than cMD. Considering the VAE training time, the overall time cost for LAST was around 40% of cMD, with all computations carried out on a Tesla P100 GPU node.
The mean RMSDs with regard to the starting protein structure in each iteration were calculated for both ADK and VVD systems and are shown in Figure 6.6. Mean RMSDs are presented with black lines, and the standard deviation is shown in red lines for each round. The maximum and minimum RMSD values are shown as the upper and lower bounds in the colored regions. Currently, we set the patience as 5: the iteration loop stops if the maximum mean RMSD does not increase in five consecutive rounds. For the simulation in the ADK system, RMSD starts with 2 Å, gradually increases to 3.5 Å, and stops at iteration 22. In contrast, the RMSDs in the VVD system are smaller and the total simulation lasts longer with a total of 30 iterations.

6.4. Discussion

In this study, we proposed a new adaptive sampling method to explore protein conformational space. LAST iteratively trains a VAE model using previous simulations, selects seeds that are structurally different from the sampled conformations, and uses them to initiate additional short MD simulations. LAST differs from previous methods in seed selection design, where the lowest-probability samples are selected and treated as seeds on the latent space of VAE. VAE has been demonstrated to be effective in learning a low-dimensional protein
representation in the latent space. [124, 146] The embeddings in the latent space are known
to keep a distance similarity: if two protein structures are similar in structure, their embed-
dings in the latent space are close to each other. With this nature, the lowest-probability
samples on the latent space are worth further exploration through MD simulations, as their
corresponding protein structures deviate from the most common structures. In LAST, these
low-probability samples are treated as seed structures to conduct additional MD simulations.

The normality of latent space provides a new opportunity for seed selection. However,
the latent space does not strictly follow a normal distribution. This is mainly because of
the relatively strong emphasis on reconstruction loss and lesser emphasis on KL divergence
during VAE training. The reconstruction loss term controls the quality of latent space data
reconstruction (how well the VAE can reconstruct a protein structure), and KL divergence
term constrains the distribution of the latent space (to what degree the latent space needs to
follow a normal distribution). Therefore, to have a well-constructed and normal regularized
latent space, appropriate weights are needed to be set for both terms. This is a challenging
task with fine tuning by hand, as the sample size keeps growing linearly with additional
MD simulations in each round. Therefore, instead of trying to find weights to balance the
reconstruction loss and KL divergence, we allow the latent space to not strictly follow a
normal distribution and use a nonparametric multivariate kernel density estimator to fit the
latent space.

One potential problem is that the distribution of the latent space might be skewed or
kurtotic. In such cases, one side of probability density function will have a long tail with
low values. This could lead to the situation that all selected seed structures lie on the long
tail side, and the corresponding protein structures of these seeds might be similar to each
other. Seed gathering on one side of latent space distribution decreases the chance to explore
more structurally different conformations and thus leads to a less efficient protein sampling
process. To partially overcome this issue, we used the cumulative distribution function to
select the lowest probability samples: data points on the two sides of the CDF are evenly selected. This improvement prevents sampling similar seeds on the boundary of protein conformational spaces.

Still, seed structures might be similar to each other. Nontargeted PaCS-MD proposed a nonredundant selection rule, which calculates pairwise RMSDs between the current simulation cycle and seeds selected in all of the past cycles. [154] Protein configurations with large RMSD are then selected as new seeds in the current cycle. We took reference from this idea when selecting seeds. The lowest-probability samples from two ends of the estimated CDF are picked sequentially, while the pairwise RMSDs to previously selected seeds are calculated. We set the RMSD threshold as 1 Å and require that the RMSD values of the newly selected seeds should be greater than the threshold. If not, LAST discards this sample and moves to the next. Moreover, LAST is a memory method: the selected seed structures are stored for RMSD calculation in future iterations, which avoids repeated sampling in the same conformational region and further improves the sampling efficiency.

For ADK, two angles with prior knowledge of its conformational dynamics were chosen to reveal the sampling efficiency. Similarly, RMSD values with reference to VVD native dark and light structures, respectively, were used for the same purpose. These preselected order parameters do not reduce the generality of LAST method because they were not used to develop VAE models. In the other words, the VAE models are “unaware” and do not require this information.

There are some tuning parameters in the LAST sampling scheme, including the dimensions of the latent space, the number of seed structures, the RMSD threshold in seed selection, the architecture of VAE model, and the number of rounds in convergence. In LAST method, the seed structures need to be selected in the frontier regions of conformational space, which has been sampled. These so-called frontier regions could not be easily identified in the Cartesian coordinates. On the contrary, after being projected onto a low-dimensional la-
tent space, the frontier regions of the conformational space representing existing simulations could be easily identified based on the distribution of existing simulations. Consequently, the seed structures for further simulations could be chosen in these frontier regions in the low-dimensional latent space. The latent space is one of the hidden layers in a VAE model. Typically, its dimension is much lower than the input dimension and is considered the bottleneck. In this study, the latent space was set as 2D to visualize, project, and compare high-dimensional protein conformations. The performance of higher dimensions in the latent space is worth further study. For the number of seed structures, 10 seeds are selected in each round. This could be changed under different protein systems and is subjected to the available computing resources. Also, the MD simulation time starting from seeds, currently set as 100 ps, can be adjusted accordingly. However, it should be noted that this simulation time should match the RMSD threshold: the simulation time should not be too short with a large RMSD threshold. Given that the conformational space of selected seeds is not likely to be visited again, it is expected to have a reasonable simulation time to fully explore the conformations in each additional MD run. Besides, the number of hidden layers in the VAE model is important to learn a useful latent space. Our previous finding suggests that a VAE model with three hidden layers is sufficient to learn the ADK protein conformations. Larger model architectures do not have a significant improvement but instead will lead to longer training time. The proper architecture of VAE, in terms of the number of hidden layers and the number of dimensions in the latent space, is worth studying to provide general guidelines when dealing with different protein families. In general, LAST method is applicable in all protein systems. The implementation of LAST method is similar regardless of whether the protein systems contain nonprotein components. However, the user needs to obtain appropriate force field parameters for the system under simulation. Lastly, it is worth noting that the convergence criterion used in this study does not represent the “true” convergence of protein systems. The notion of “true” convergence, as discussed in previous studies, [155–157]
is elusive in simulations. More appropriate criteria are needed for the convergence signal in adaptive sampling, through either numerical indicators or self-consistency checks.

6.5. Conclusion

In this study, we present an adaptive sampling method, latent space-assisted adaptive sampling for protein trajectories, to accelerate the exploration of protein conformational spaces. LAST iterates through variational autoencoder training, seed selection, and additional short MD simulations. LAST differs from previous methods in seed selection where the lowest-probability samples in the learned latent space are selected and treated as seed structures. LAST method is compared with SDS and cMD using ADK and VVD protein systems, each with different low-dimensional representations. In both systems, LAST can capture the key protein characteristics and select seeds that lie in the boundary of conformational space. For ADK simulations, LAST explored two transition paths that are in agreement with previous findings. For VVD simulations, LAST is three times faster than conventional MD for exploring the same conformational regions. To conclude, LAST provides an alternative method for efficient adaptive sampling.
CHAPTER 7
CONCLUSION

The function of a protein is directly related to its three-dimensional conformation. Thus, the study of protein conformations and functions is crucial for understanding the fundamental biology of living organisms.

Allostery is one of the critical processes in biology, where an effector molecule binds to a site separate from the active site of a protein. This binding results in conformational and dynamical changes that can regulate the protein’s function, making it an essential aspect of cellular signaling and considered the second secret of life. Despite its importance, the allosteric mechanisms of most proteins remain unknown, with no universal mechanism yet being discovered.

The first half of this thesis summarizes several projects related to protein allostery.

The mechanism of signaling transduction in PtAu1α LOV domain (AuLOV), including flanking helices, remains unclear because of this dissimilarity, which hinders the study of PtAu1α as an optogenetic tool. To clarify this mechanism, a combination of tree-based machine learning models, Markov state models, machine learning-based community analysis, and transition path theory were employed to analyze the allosteric process quantitatively. The results agree with the reported experimental findings and reveal a previously overlooked $\alpha$ helix and protein linkers as necessary in promoting protein conformational changes. This integrated approach can be considered a general workflow and applied to other allosteric proteins to provide detailed information about their allosteric mechanisms.
Besides the application of computational methods to understand allostery, this thesis also focuses on the prediction of allostery with machine learning models. Allostery is considered important in regulating protein activity. Drug development depends on the understanding of allosteric mechanisms, especially the identification of allosteric sites, which is a prerequisite in drug discovery and design. Many computational methods have been developed for allosteric site prediction using pocket features and protein dynamics. This thesis presents an ensemble learning method, consisting of eXtreme gradient boosting and graph convolutional neural network, to predict allosteric sites. The model can learn physical properties and topology without any prior information and shows good performance under multiple indicators. Prediction results showed that 84.9% of allosteric pockets in the test set appeared in the top 3 positions. The PASSer: Protein Allosteric Sites Server (https://passer.smu.edu) provides insights for further analysis in drug discovery.

This thesis further expands from protein allostery to general protein conformations with the help of deep learning.

The variational autoencoders, a type of deep learning model, are demonstrated to explore a protein’s conformational space through MD simulations. VAEs were shown to be superior to autoencoders through a benchmark study, with a low deviation between the training and decoded conformations. Moreover, the learned latent space in the VAE can generate unsampled protein conformations. Additional simulations starting from these generated conformations accelerated the sampling process and explored hidden spaces in the conformational landscape.

Based on this idea, an adaptive sampling method, LAST, was designed to accelerate the exploration of protein conformational space. This method comprises cycles of (i) variational autoencoder training, (ii) seed structure selection on the latent space, and (iii) conformational sampling through additional MD simulations. The proposed approach is validated on four conformations of two protein systems: two metastable states of ADK and two native
states of VVD. In all four conformations, seed structures were shown to lie on the boundary of conformation distributions. Moreover, large conformational changes were observed in a shorter simulation time when compared with conventional MD simulations in both systems. In metastable ADK simulations, LAST explored two transition paths toward two native states while cMD was trapped in an energy basin. In VVD light state simulations, LAST only took one-third of cMD simulation time to have a similar conformational space.

 Overall, this thesis presents new computational methods that can be applied to understand protein allostery and validate the allosteric process with experimental results. This thesis also explored the possibility of variational autoencoders to understand protein conformation and provided a new adaptive sampling method to accelerate the sampling process.
BIBLIOGRAPHY


[92] A. Panjkovich and X. Daura, Exploiting protein flexibility to predict the location of allostERIC sites, BMC bioinformatics 13 (2012) 273. 37, 45


[100] A. Panjkovich and X. Daura, Pars: a web server for the prediction of protein allostERIC and regulatory sites, Bioinformatics 30 (2014) 1314–1315. 37, 39


[125] T. P. Schrank, J. O. Wrabl and V. J. Hilser, Conformational heterogeneity within the lid domain mediates substrate binding to escherichia coli adenylate kinase: function follows fluctuations, Dynamics in Enzyme Catalysis (2013) 95–121. 46


86


APPENDIX A

APPENDIX (FIRST AUTHOR ONLY)


A.1. PAPER I

ABSTRACT: Molecular dynamics (MD) simulations have been widely applied to study macromolecules including proteins. However, the high dimensionality of the data sets produced by simulations makes thorough analysis difficult and further hinders a deeper understanding of biomacromolecules. To gain more insights into the protein structure–function relations, appropriate dimensionality reduction methods are needed to project simulations onto low-dimensional spaces. Linear dimensionality reduction methods, such as principal component analysis (PCA) and time–structure-based independent component analysis (t-ICA), could not preserve sufficient structural information. Though better than linear methods, nonlinear methods, such as t-distributed stochastic neighbor embedding (t-SNE), still suffer from the limitations in avoiding system noise and keeping inter-cluster relations. ivis is a novel deep learning-based dimensionality reduction method originally developed for single-cell data sets. Here, we applied this framework for the study of light, oxygen, and voltage (LOV) domains of diatom Phaeodactylum tricornutum aureochrome 1a (PtAu1a). Compared with other methods, ivis is shown to be superior in constructing a Markov state model (MSM), preserving information of both local and global distances, and maintaining similarity between high and low dimensions with the least information loss. Moreover, the ivis framework is capable of providing new perspectives for deciphering residue-level protein allostery through the feature weights in the neural network. Overall, ivis is a promising member of the analysis toolbox for proteins.

INTRODUCTION
Molecular dynamics (MD) simulations have been widely used in biomolecules to provide insights into their functions in atomic-scale mechanisms.1 For this purpose, an extensive time scale is generally preferred for the simulations to study protein dynamics and functions. Due to the emergence of graphics processing units (GPU) and their application in biomolecular simulations, the MD simulation time scale has reached from nanoseconds to experimentally meaningful microseconds.2,3 However, simulation data for biomacromolecules such as proteins are high-dimensional and suffer from the curse of dimensionality,1 which hinders an in-depth analysis, including extracting slow time-scale protein motions,5 identifying representative protein structures,6 and clustering kinetically similar macrostates.7 To make these analyses feasible, it will be informative to construct a low-dimensional space to characterize protein dynamics in the best way possible.

In recent years, new dimensionality reduction algorithms have been developed and can be applied to analyze protein simulations, construct representative distributions in a low-dimensional space, and extract intrinsic relations between the protein structure and functional dynamics. These methods can be generally categorized into linear and nonlinear methods.8,9 Linear dimensionality reduction methods produce new variables as the linear combination of the input variables, such as principal component analysis (PCA)10 and time–
structure-based independent component analysis (t-ICA). Nonlinear methods construct variables through a nonlinear function, including t-distributed stochastic neighbor embedding (t-SNE) and auto encoders. It was reported that nonlinear methods were more powerful in reducing dimensionality while preserving representative structures.

Information is inevitably lost to a certain degree through the dimensionality reduction process. It is expected that the distances among data points in the low-dimensional space resemble the original data in the high-dimensional space. The Markov state model (MSM) is often applied to study the dynamics of biomolecular systems. MSM is constructed by clustering states in the reduced dimensional space to obtain long-time kinetic information. However, many dimensionality reduction methods, such as PCA and t-ICA, fail to keep the similarity characteristics in the low dimension, which would cause a misleading clustering analysis based on the projections of low-dimensional space. Therefore, more appropriate dimensionality reduction methods are needed to build a proper MSM.

A novel framework, ivis, is a recently developed dimensionality reduction method for single-cell data sets. ivis is a nonlinear method based on Siamese neural networks (SNNs). The SNN architecture consists of three identical neural networks and ranks the similarity to the input data. The loss function used for the training process is a triplet loss function that calculates the Euclidean distance among data points and simultaneously minimizes the distances between the same labels while maximizing the distances between data of different labels. Due to this intrinsic property, the ivis framework is capable of preserving both local and global structures in a low-dimensional space.

With the success in single-cell expression data, the ivis framework is promising as a dimensionality reduction method for simulations of biomacromolecules to investigate their functional dynamics such as allostery. Diatom *Phaeodactylum tricornutum* aureochrome 1a (PtAu1a) is a recently discovered light, oxygen, or voltage (LOV) protein from the photosynthetic stramenopile alga Vaucheria frigida. This protein consists of an N-terminal domain, a C-terminal LOV core, and a basic region leucine zipper (bZIP) DNA-binding domain. PtAu1a is a monomer in the native dark state. The interaction between its LOV core and bZIP prohibits DNA binding. Upon light perturbation, a covalent bond forms between the C4a position of the cofactor flavin mononucleotide (FMN) and sulfur in cysteine 287, triggering a conformational change that leads to the LOV domain dimerization. In the current study, the PtAu1a LOV domain (AuLOV) with two flanking helices (A′α and Jα helices) is simulated through MD simulations. The structures of both the dark and light states are shown in Figure 1. The main difference between AuLOV and most other LOV proteins is that the LOV domain lies in the C-terminal in AuLOV while it lies in the N-terminal in other LOV proteins. Therefore, the conformational changes in AuLOV are expected to differ from other LOV proteins, raising the question of how the allosteric signal transmits in AuLOV. In the current study, the ivis framework, together with other dimensionality reduction methods, is applied to project the AuLOV simulations onto reduced dimensional spaces. The performances of the selected methods are assessed and compared, validating the ivis as a superior framework for dimensionality reduction of biomacromolecule simulations.

## METHODS

### Molecular Dynamics (MD) Simulations

The crystal structures of AuLOV dark and light states were obtained from the Protein Data Bank (PDB) with PDB IDs 5dkk and 5dkl, respectively. The light structure sequence starts from Gly234, while the dark structure sequence starts from Phe239 in chain A and Ser240 in chain B. For consistency, residues before Ser240 were removed to keep the same number of residues in all chains. Therefore, simulations of dark and light states can be treated similarly. Both structures contain FMN as a cofactor. The FMN force field from a previous study was used in this study. Two new states, named the transient dark state (forcing the cysteinyl–flavin C4a adduct in the dark state structure) and the transient light state (breaking the cysteinyl–flavin C4a adduct in the light state structure), were constructed to fully explore the protein conformational space. Two monomers (Figure 1A) and a dimer (Figure 1B) were simulated in the dark and light states, respectively.

The crystal structures with added hydrogen atoms were solvated within a rectangular water box using the TIP3P water model. Sodium and chloride ions were added for charge neutralization. Energy minimization was done for each water box. The system was further subjected to 20 picoseconds (ps) of MD simulations to increase the temperature from 0 to 300 K and another 20 ps simulation for equilibrium. In all, 10 nanoseconds (ns) of isothermal–isobaric ensemble (NPT) MD simulations under 1 bar pressure were conducted. The
canonical ensemble (NVT) is usually applied in the production runs to investigate the allosteric process. In all, 1.1 microseconds (μs) of a canonical ensemble (NVT) Langevin MD simulation at 300 K was carried out for each production run. The Langevin dynamics friction coefficient that couples the system to a heat bath was set to 1 ps$^{-1}$. With minimum perturbation to the dynamical properties of the protein system. For all production simulations, the first 100 ns simulation is treated as the equilibration stage and not included in the analysis. For each structure, three independent MD simulations were carried out and a total of 12 μs simulations were used in the analysis. All chemical bonds associated with hydrogen atoms were constrained with the SHAKE method. A step size of 2 femtoseconds (fs) was used and simulation trajectories were saved for every 100 ps. The periodic boundary condition (PBC) was applied in simulations. Electrostatic interactions were calculated with the particle mesh Ewald (PME) algorithm and a cutoff of 1.2 nanometers (nm). Simulations were conducted using graphics processing unit accelerated calculations of OpenMM with the CHARMM simulation package, version c41b1, and the CHARMM27 force field.

**Feature Processing.** In MD simulations, protein structures are represented as atom positions in Cartesian coordinates. However, this representation is neither rotation invariant nor feasible for analysis purposes due to the significant number of atoms with a total of 3 N degrees of freedom. To represent the protein structures with rotational invariance and essential structural information, pair-wise backbone Cα distances were selected to represent the overall protein configuration. Following our previously proposed feature-processing method, distances were encoded as a rectified linear unit (ReLU)$^{38}$-like activation function and further expanded as a vector.

$$ReLU(x) = \max(0, x)$$

(1)

**Dimensionality Reduction Methods.** ivis. ivis is a deep learning-based method for structure-preserving dimensionality reduction. This framework is designed using Siamese neural networks, which implement a novel architecture to rank similarity among input data. Three identical networks are included in the SNN. Each network consists of three dense layers and an embedding layer. The size of the embedding layer was set to 2, aiming to project high-dimensional data into a two-dimensional (2D) space. The scaled exponential linear unit (SELU)$^{39}$ activation function is used in the dense layers

$$\text{selu}(x) = \lambda \begin{cases} 
    x & \text{if } x > 0 \\
    \alpha \exp(x) - \alpha & \text{if } x \leq 0
\end{cases}$$

(2)

The LeCun normal distribution is applied to initialize the weights of these layers. For the embedding layer, the linear activation function is used, and weights are initialized with Glorot’s uniform distribution. To avoid overfitting, dropout layers with a default dropout rate of 0.1 are used for each dense layer.

A triplet loss function is used as the loss function for training

$$L_{\text{t}}(\theta) = \sum_{k \neq p, n} D_{a,p} - \min(D_{a,n}, D_{p,n}) + m$$

(3)

where $a$, $p$, and $n$ are the anchor points, positive points, and negative points, respectively. $D$ and $m$ are the Euclidean distance and margin, respectively. Anchor points are points of interest. The triplet loss function aims to minimize the distance between anchor points and positive points while maximizing the distances between anchor points and negative points. The distances between positive points and negative points are also taken into account, as shown in $\min(D_{a,n}, D_{p,n})$ in the above equation.

The $k$-nearest neighbors (KNNs) are used to obtain data for the triplet loss function. $k$ is a tuning parameter and is set to 100. For each round of calculation, one point in the data set is selected as an anchor. A positive point is randomly selected among the nearest $k$ neighbors around the anchor, and a negative point is randomly selected outside the neighbors. For each training epoch, the triplet selection is updated to maximize the distances in both local and global distances.

If the data set can be classified into different groups based on their intrinsic properties, ivis can also be used as a supervised learning method by combining the distance-based triplet loss function with a classification loss. Supervision weight is a tuning parameter to control the relative importance of loss function in labeling data.

The neural network is trained using the Adam optimizer function with a learning rate of 0.001. Early stopping is a method to prevent overfitting in a training neural network and is applied in this study to terminate the training process if the loss function does not decrease after 10 consecutive epochs.

**t-Distributed Stochastic Neighbor Embedding (t-SNE).** t-SNE is a nonlinear dimensionality reduction method for macromolecular simulations. For a given $n$-dimensional data, t-ICA is employed by solving the following equation

$$\tilde{CF} = CKF$$

(4)

where $K$ is the eigenvalue matrix, $C$ is the correlation matrix, and $F$ is the eigenvector matrix. $\tilde{C}$ is the time lag correlation matrix defined as

$$\tilde{C} = \langle (x(t) - \langle x(t) \rangle)(x(t + \tau) - \langle x(t) \rangle) \rangle$$

(5)

The results calculated by t-ICA are linear combinations of input features that are highly autocorrelated.

**Principal Component Analysis (PCA).** PCA is a method that finds the projection vectors that maximize the variance by conducting an orthogonal linear transformation. In the new coordinate system, the greatest variance of the data lies on the first coordinate and is called the first principal component. Principal components can be solved through the singular value decomposition (SVD). Given data matrix $X$, the covariance matrix can be calculated as

$$C = X^TX/(n - 1)$$

(6)

where $n$ is the number of samples. $C$ is a symmetric matrix and can be diagonalized as

$$C = VLV^T$$

(7)

where $V$ is a matrix of eigenvectors and $L$ is a diagonal matrix with eigenvalues $\lambda_i$ in descending order.

**t-Distributed Stochastic Neighbor Embedding (t-SNE).** t-SNE is a nonlinear dimensionality reduction method that tries to embed similar objects in high dimensions to points close to each other in a low-dimensional space. t-SNE has been demonstrated to be a suitable dimensionality reduction method for protein simulations.
consists of two stages. First, conditional probability is calculated to represent the similarity between two objects as

$$P_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_i - x_k\|^2 / 2\sigma^2)}$$  \hspace{3cm} (8)$$

where \(\sigma\) is the bandwidth of the Gaussian kernels.

While the conditional probability is not symmetric since \(P_{ij}\) is not equal to \(P_{ji}\), the joint probability is defined as

$$P_{ij} = \frac{P_{ij} + P_{ji}}{2N}$$ \hspace{3cm} (9)$$

To better represent the similarity among objects in the reduced map, the similarity \(q_{ij}\) is defined as

$$q_{ij} = \frac{\sum_{i \neq j} (1 + \|y_i - y_j\|^2)^{-1}}{\sum_{i \neq j} (1 + \|y_i - y_j\|^2)^{-1}}$$ \hspace{3cm} (10)$$

Combined with the joint probability \(P_{ij}\) and similarity \(q_{ij}\), Kullback–Leibler (KL) divergence is used to determine the coordinates of \(y\), as

$$\text{KL}(P\|Q) = \sum_{i \neq j} P_{ij} \log \frac{P_{ij}}{q_{ij}}$$ \hspace{3cm} (11)$$

The KL divergence measures the differences between high-dimensional data and low-dimensional points, which is minimized through the gradient descent method.

A drawback of the traditional t-SNE method is the slow training time. To speed up the computational time of the dimensionality reduction process, multicore t-SNE\(^4^2\) is used training time. To speed up the computational time of the dimensional data and low-dimensional points, which is

$$\bar{\lambda}_1$$ \hspace{3cm} (17)$$

where \(\lambda_1\) is the second eigenvalue and \(\tau\) is the lag time.

The generalized matrix Rayleigh quotient (GMQR)\(^4^9\) generated using the combination of cross-validation and variational approaches, was used to assess the effectiveness of MSM in dimensions and dimensionality reduction methods. State decompositions are different through various dimensionality reduction methods. A good method is expected to lead to a Markov state model with larger GMQR values.
**Machine Learning Methods. Random Forest (RF).** Random forest is a supervised machine learning method that was used in this study for trajectory-state classification. A random forest model consists of multiple decision trees, which are a class of partition algorithm that recursively groups data samples of the same label. Features at each split are selected based on the information gained. A final prediction result of the random forest is made from the results in each decision tree through a voting algorithm. For random forest models at each depth, the number of decision trees was set to 50. Scikit-learn (version 0.20.1) was used for RF implementation.

**Artificial Neural Network (ANN).** An artificial neural network was used to learn the nonlinear relationships of coordinates on the reduced 2D dimension. An ANN is generally formed with an input layer, a hidden layer, and an output layer. In each layer, different neurons (nodes) are assigned and connected with adjacent layer(s). During the training process, input data are fed through the input and hidden layers and prediction results are made in the output layer. For each training step, the error between the predicted result and the actual result is propagated from the output layer back to the input layer, which is also called back-propagation, and the weight in every neuron is updated. When there is more than one hidden layer, ANN is also referred to as a deep neural network (DNN), which requires more computation power. To minimize the training cost, only two hidden layers, each with 64 nodes, were used. The Adam optimizer was used for weight optimization. ANN was implemented with Keras (version 2.2.4-tf).

**RESULTS**

**Data Set of Cα Distances Represents the Protein Structures.** There are two native states of AuLOV: native dark state and native light state. To explore the protein response with regard to the formation of the covalent bond between cysteine 287 and FMN, two new states were constructed as a transient dark state and a transient light state by forcing the cysteinyl–flavin C4a adduct in the native dark state and breaking this adduct in the native light state, respectively. The RMSDs of MD simulations are plotted in **Figure 2**. For each trajectory, the RMSD values were calculated with regard to the first frame. Averaged RMSDs were 1.75, 2.04, 2.39, and 2.08 Å in native dark, transient dark, transient light, and native light states, respectively. Compared with the result in the native dark state, the higher RMSD value in the transient dark state indicates that the light-induced covalent bond Cys287-FMN increases the protein flexibility and dynamics. The transient light state displays the highest averaged RMSD value, indicating the highest flexibility or the largest conformational change.

The pair-wised distances of backbone Cα in simulations were extracted as features representing the character of protein configurations. There are 254 residues in the AuLOV structure, and a total of $254 \times 253/2 = 32,131$ Cα distances were calculated. Before further analysis, features were transformed into vectors with our proposed technique outlined in the Methods section. Considering the nonbonded chemical interaction, 10.0 Å was selected as the threshold for feature transformation. There are 10,000 frames in each trajectory, leading to a sample size of 120,000 in the overall data set. Full data sets were applied in all analyses. To gain more statistical significance, each MD trajectory was split into five subtrajectories at equal intervals. The performance assessments were conducted for each subtrajectory independently. The mean and standard deviation values of the five subsets were calculated.

**Information is Well Preserved in the ivis Dimensionality Reduction Method.** Several hyperparameters of the ivis model were selected based on the recommended values for different observation sizes. Given the large number of sample size, $k$ was set to 100 and the number of early stopping epochs was 10. The Maaten neural network architecture was selected, which consists of three dense layers with 500, 500, and 2000 neurons, respectively. To select the best parameter of supervision weight, the full trajectory data set was randomly split into a training set (70%) and a testing set (30%). ivis models were trained on the training set and validated on the testing set. The prediction result with different supervision weights is plotted in **Figure 3**. The ivis model performed poorly at 0.0 supervision weight, which corresponds to unsupervised ivis, with an average accuracy below 25%. The average accuracy values for other supervision weights were stable and over 95%. Specifically, there was no significant increase in the accuracy value after the 0.1 supervision weight, which was chosen as the hyperparameter for the supervised ivis model. An unsupervised ivis framework with the same value of other hyperparameters was applied for comparison.

Four dimensionality reduction models (supervised ivis, PCA, t-SNE, and t-ICA) were applied to the MD simulations to project a high-dimensional (32,131) space to a 2D surface.
The supervised ivis dimensionality reduction method, as well as PCA, successfully separated dark and light states while keeping the corresponding transition states close (Figure 4A,B). These states are important for dynamical analysis as they could be used to reveal the free-energy and kinetic transition landscape for the target system. For t-SNE (Figure 4C) and t-ICA (Figure 4D) projections, the transient dark state and native dark state overlap significantly, thus hindering the extraction of thermodynamics and kinetics information. Therefore, the supervised ivis dimensionality reduction method and PCA are demonstrated to be proper in representing the chemical information in the low dimension among the investigated methods.

The $k$-means clustering was used in the reduced dimensions to partition a total of 120,000 frames from AuLOV MD trajectories into 1000 microstates. Within each cluster, the RMSDs were calculated for each structure pair. An RMSD value of each cluster is defined as the average RMSD value among all structure pairs within that cluster. The results of five dimensionality reduction models are shown in Figure 5A. The average RMSD value of an appropriate microstate should be lower than 1.0 Å. From this perspective, unsupervised ivis and supervised ivis show similar values in each microstate and are the best two methods among the selected methods. As reported previously, t-SNE also exhibited good performance in measuring the similarity with the Cartesian coordinates.

**Figure 4.** Two-dimensional projections of four dimensionality reduction methods: (A) supervised ivis, (B) PCA, (C) t-SNE, and (D) t-ICA.
A metric to compare different dimensionality reduction methods is the implied relaxation time scale calculated from the Markov state model. To build MSM, MD simulations were projected onto a 2D space and 1000 microstates were sampled through k-means with the corresponding estimated relaxation time scales. For each method, the slowest time scale in each lag time was extracted based on different lag times ranging from 10 to 100 ns and is shown in Figure 5B. The convergence of time scales is important for eigenvalue and eigenvector calculations. For all five models, relaxation time scales converged, indicating the Markovianity of the MSMs. Both supervised ivis and unsupervised ivis models show long time scales, indicating the effectiveness of MSM built on the reduced spaces.

Euclidean distances between data points in the low-dimensional space are expected to reflect the similarity in the high dimension. To quantify the degree of this relationship kept in reduced dimensional space, Spearman correlation coefficients were calculated between Euclidean distance pairs. The results are shown in Figure 6A. The mean values for unsupervised ivis, supervised ivis, PCA, t-SNE, and t-ICA are 0.69, 0.84, 0.86, 0.41, and 0.70, respectively. The height of each box represents the interquartile range.

Figure 5. Analysis results of 2D projections for different dimensionality reduction methods. (A) Average values of RMSDs in microstates clustered within a projected 2D dimensional space. (B) Estimated implied time scales from Markov state models with regard to different lag times. For each model, the mean value of the implied time scale is calculated among five subsets and is plotted in solid color. The standard deviation is calculated to show the stability for each lag time and is illustrated using a light color.

Figure 6. Results of quantitative analysis. (A) Spearman correlation coefficient results of different dimensionality reduction methods. The mean values for unsupervised ivis, supervised ivis, PCA, t-SNE, and t-ICA are 0.69, 0.84, 0.86, 0.41, and 0.70, respectively. The height of each box represents the interquartile range. (B) Mantel test scores of different dimensionality reduction methods. The mean values for unsupervised ivis, supervised ivis, PCA, t-SNE, and t-ICA are 0.83, 0.95, 0.98, 0.15, and 0.89, respectively. (C) Shannon information content of different dimensionality reduction methods. The mean values for unsupervised ivis, supervised ivis, PCA, t-SNE, and t-ICA are 449.0, 714.6, 327.0, 707.3, and 54.0, respectively. To avoid dependent variables in the information content calculation, high-dimensional Cα distances were projected to 1D. (D) Generalized matrix Rayleigh quotient with different dimensions and dimensionality reduction methods.
in the original space and those in the reduced space. The results are shown in Figure 6A. While PCA preserved the Euclidean distances well with an average coefficient of 0.86, the supervised ivis model showed a comparable high coefficient of 0.84. The unsupervised ivis model also exhibited the ability to preserve the linear relationship. The poor performance of the t-SNE model may be due to the fact that t-SNE is a nonlinear method and therefore suffers from the problem that distance in the high-dimensional space is not linearly projected to the low-dimensional space, as reported in other studies.58,59

While ivis models showed good ability in keeping the linear projection relation, the Spearman correlation coefficient fails to overcome the problem that features are not independent. The pair-wised distances are subjected to the molecular motion of $\alpha_C$, wherein changing the coordinate of one $\alpha_C$ atom would affect the distances related to this atom. Therefore, to address this issue, the Mantel test was used to randomize the Euclidean distances. Permutations of rows and columns in the Euclidean distance matrix were done 10,000 times, with the Pearson correlation coefficient calculated each time. The results of the Mantel test are plotted in Figure 6B. Both unsupervised ivis and supervised ivis showed remarkable results in preserving the correspondence relationship in a randomized order, with the mean coefficients of 0.83 and 0.95, respectively.

During the process of dimensionality reduction, information is inevitably lost to some degree. To measure the retaining information through the dimensionality reduction process, the Shannon information is applied to the coordinates in the low-dimensional space. However, when dealing with multiple variables, especially for the dependent $\alpha_C$ distances, the total Shannon information is not equal to the sum of the Shannon information of each variable. To reduce the computation complexity, high-dimensional features were reduced to 1D for calculation and the results are plotted in Figure 6C. It shows that the supervised ivis model is superior in preserving information content with the least information loss. It is also worth noting that t-SNE showed better performance than the unsupervised ivis model.

To study the performance of the Markov state model on dimensions and dimensionality reduction methods, the generalized matrix Rayleigh quotient was calculated for each dimension and method (Figure 6). The results of four methods showed different trends. Supervised ivis and t-ICA methods were the least and most affected by the number of dimensions, respectively. For PCA and t-SNE, the optimal parameter of the number of dimensions is in the tens. Two dimensions are typically used for MSM construction and visualization purpose.60–62 In this regard, supervised ivis exhibited the highest GMRQ value.

ivis Helps in Understanding Biological Systems and the Allosteric Mechanism. The transition probabilities among macrostates in ivis projections are shown in Figure 7. Based on the similarity to crystal structures, macrostates 2 and 8 are referred to as the native dark state and native light state, respectively. Other macrostates are considered as transient states. The probabilities of the native dark state and native light state to remain to themselves are among the highest ones and indicate the high stability of these two states. It is interesting to observe that macrostate 9 may have the highest stability among all Markov states.

The effectiveness of MSM depends on the projected 2D space, where appropriate discrete states are produced by clustering the original data points in the projection space. The number of macrostates is determined based on the implicated time scales using different lag times in different reduced spaces. In this study, nine, nine, seven, nine, and seven macrostates were selected for unsupervised ivis, supervised ivis, PCA, t-SNE, and t-ICA, respectively. The samples were clustered through Perron-cluster cluster analysis (PCCA). The data set was further split into a training set (70%) and a testing set (30%). Two machine learning methods (random forest and artificial neural network) were applied to predict the macrostates of each data point based on the pair-wised $\alpha_C$ distances. Prediction accuracy results are plotted in Figure 8A,B. It shows that the supervised ivis framework is the best among the five dimensionality reduction methods. Surprisingly, while the unsupervised ivis model was trained without class labels in the loss function, the high prediction accuracy of this model demonstrates its good performance on the 2D projections. Random forest is often applied to distinguish the macrostates since it provides feature importance, which is important for the interpretation of biological systems. The accumulated feature importance of a random forest model on the supervised ivis model is plotted in Figure 8C. The top 490 features account for 90.2% of the overall feature importance.

The high prediction accuracy of the supervised ivis framework suggests that supervised ivis is more promising in elucidating the conformational differences among macrostates. The neural network architecture on the first dense layer of the supervised ivis model was 32 131 × 500, where 32 131 and 500 represent the number of $\alpha_C$ distances and dense layers, respectively. To identify key residues and structures that are important in the dimensionality reduction process, 32 131 feature weights on the last layer were treated as the feature importance and shown as the protein contact map in Figure 9. The contact map is symmetrical along the diagonal. The upper triangular part is divided into four regions as follows: region 1: pair wised $\alpha_C$ distances within chain A, region 4: $\alpha_C$ distances within chain B, and regions 2 and 3: $\alpha_C$ distances between chains A and B. Our results show characteristics similar to those in a previous study.17 Local protein structures are encoded to features close to the diagonal. Global structures are encoded to features further from the diagonal. In Figure 9, the local information is shown as a red rectangle as the $\alpha_C$ and $\delta_C$ helices in the AuLOV system, and global information is shown as a black rectangle as the $G\beta$ and $H\beta$ strands. While regions 2 (protein interactions from chain A to chain B) and 3 (protein interactions from chain B to chain A) are mostly symmetrical,
we found an asymmetrical behavior (red circle in Figure 9) in which the interaction between Jα in chain A and linkers in chain B is stronger than the interaction between Jα in chain B and linkers in chain A.

To examine the important residues identified in the protein contact map, for each Cα distance, the corresponding feature weight was accumulated to the two related residues. Therefore, the significance of residues and structures is quantified. The top 20 residues are listed in Table 1, with important residues that are experimentally identified, shown in bold font.

![Figure 8](image-url) Prediction accuracy of different machine learning models. (A) A random forest and (B) an artificial neural network were used on the reduced 2D spaces to predict the labels of macrostates from MSM. (C) Accumulated feature importance of a random forest model applied in the projections of the supervised ivis framework at depth 5.

![Figure 9](image-url) Protein contact map with the corresponding protein structures. Feature weights of the first dense layer in the supervised ivis dimensionality reduction method were extracted and were colored red (positive), white (close to zero), and blue (negative). The residue sequence starts from Ser240 in chain A and ends in Glu367 in chain B.

Table 1. Top 20 Residues Identified in the Supervised Ivis Framework

<table>
<thead>
<tr>
<th>residue</th>
<th>importance (%)</th>
<th>residue</th>
<th>importance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILE 242</td>
<td>1.12</td>
<td>PHE 241</td>
<td>1.10</td>
</tr>
<tr>
<td>LEU 245</td>
<td>1.07</td>
<td>ALA 248*</td>
<td>1.04</td>
</tr>
<tr>
<td>GLN 250</td>
<td>1.01</td>
<td>SER 314</td>
<td>1.00</td>
</tr>
<tr>
<td>GLN 246</td>
<td>0.99</td>
<td>ASN 251</td>
<td>0.98</td>
</tr>
<tr>
<td>THR 247</td>
<td>0.97</td>
<td>PRO 268</td>
<td>0.97</td>
</tr>
<tr>
<td>ASN 329</td>
<td>0.96</td>
<td>GLN 350</td>
<td>0.96</td>
</tr>
<tr>
<td>MET 313</td>
<td>0.95</td>
<td>ALA 244</td>
<td>0.94</td>
</tr>
<tr>
<td>PHE 331</td>
<td>0.93</td>
<td>ASN 311</td>
<td>0.93</td>
</tr>
<tr>
<td>SER 240</td>
<td>0.92</td>
<td>GLN 365</td>
<td>0.92</td>
</tr>
<tr>
<td>CYS 351</td>
<td>0.91</td>
<td>ALA 335</td>
<td>0.90</td>
</tr>
</tbody>
</table>

*Experimentally confirmed important residues are shown in bold font.

Table 2. Accumulated Feature Importance of Secondary Structures

<table>
<thead>
<tr>
<th>secondary structure</th>
<th>importance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aα</td>
<td>13.17</td>
</tr>
<tr>
<td>Aβ</td>
<td>6.34</td>
</tr>
<tr>
<td>Bβ</td>
<td>2.36</td>
</tr>
<tr>
<td>Ca</td>
<td>8.50</td>
</tr>
<tr>
<td>Da</td>
<td>4.14</td>
</tr>
<tr>
<td>Ea</td>
<td>1.40</td>
</tr>
<tr>
<td>Fa</td>
<td>5.50</td>
</tr>
<tr>
<td>Gβ</td>
<td>6.52</td>
</tr>
<tr>
<td>Hβ</td>
<td>8.67</td>
</tr>
<tr>
<td>θβ</td>
<td>7.98</td>
</tr>
<tr>
<td>Jα</td>
<td>10.44</td>
</tr>
<tr>
<td>linkers</td>
<td>24.98</td>
</tr>
</tbody>
</table>

Ivis is More Computationally Efficient Than t-ICA and t-SNE. A key factor in comparing different dimensionality reduction methods is their computational cost, for it could be prohibitively expensive when dealing with a large-size and...
high-dimensional data set. To compare the computational efficiency of different dimensionality reduction methods with regard to sample size and feature size, three randomly generated data sets with uniform distributions between 0 and 1 were applied for each data set size. The relation between runtime and sample size, with a feature size of 1000, is shown in Figure 10A. While t-SNE is stable and fast in small data sets (≤10 000 sample size), its runtime grows the fastest among the five models and is not feasible for a large data set. t-ICA and PCA overlapped with each other since these two models are less affected by the sample size. Unsupervised ivis and supervised ivis exhibited similar runtime results. The relation between runtime and feature size with a sample size of 10 000 is shown in Figure 10B. t-ICA and t-SNE show similar trends in the runtime growth trend, as they perform fast in small feature sizes (≤10 000), but they are not practical in higher dimensions. While both ivis models are slower than PCA, the runtimes of these two models are acceptable for a large sample size and high dimension. The training process of supervised ivis is further displayed in Figure 11. Triplet loss was stable after 4 epochs and stopped at 32 epochs, with early stopping of 10.

**DISCUSSION**

As a deep learning-based algorithm, the ivis framework was originally designed for single-cell experiments to provide a new approach for visualization and explanation purposes. In this study, ivis is applied on the MD simulations of allosteric protein AuLOV for dimensionality reduction. Combined with several performance criteria, ivis is demonstrated to be effective in keeping both local and global features while offering key insights for the mechanism of protein allostery.

Various dimensionality reduction methods have been used in protein systems, such as PCA, t-ICA, and t-SNE. As linear methods, PCA and t-ICA aim to capture the maximum variance and autocorrelation of protein motion, respectively. However, nonlinear dimensionality reduction methods, such as t-SNE, have been shown to be superior to linear methods in keeping the similarity between high dimensions and low dimensions. Nevertheless, the limitations of t-SNE, such as susceptibility to system noise and poor performance in extracting the global structure, hinder further interpretations for biological systems. Compared with these dimensionality reduction methods, ivis is outstanding in preserving distances in the low-dimensional spaces and could be utilized for biological system explanations.

Dimensionality reduction methods have different strengths in preserving structural information and can be applied to various data sets. While there is no universal standard measuring the performance of different methods, an appropriate method should reflect the distance and similarity between projections in a low-dimensional space. Similar structures in the high-dimensional space should be close in the low-dimensional space. This criterion is important in the construction of the Markov state model, which requires clustering discrete microstates on the projections. Improper projections would lead to poor MSMs, thus obscuring the protein motions and hindering further structure-function study. Moreover, an adequate MSM requires the similarity between structures in each microstate. To evaluate the effectiveness of MSM, the average RMSD value is often used as a good indicator for dimensionality reduction methods. In this regard, both unsupervised ivis and supervised ivis are suitable to build MSM in a low-dimensional space. Estimated relaxation time scale reflects the number of steady states and is used to construct kinetically stable macrostates. The time scale of protein motion ranges from milliseconds to seconds in

![Figure 10](https://dx.doi.org/10.1021/acs.jcim.0c00485)

Figure 10. Computation time of each dimensionality reduction method spent on fitting high-dimensional data. (A) Runtime result of 1000 feature sizes with different sample sizes. Results of PCA and t-ICA were overlapped because of the time scale. (B) Runtime result of a sample size of 10 000 with different feature sizes.

![Figure 11](https://dx.doi.org/10.1021/acs.jcim.0c00485)

Figure 11. Triplet loss of each epoch for the supervised ivis framework with a supervision weight of 0.1 and early stopping of 10. The model is trained on a data set of 10 000 samples with 60 000 dimensions. The dashed gray line indicates the expected termination in training with early stopping of 5.
experiments. Among all of the tested dimensionality reduction methods, the ivis framework showed the longest time scale, with over $10^{-5}$ s. This experimentally meaningful time scale, combined with the average RMSD, suggests the success of ivis in the construction of MSM.

It is expected that Euclidean distances between data points in the high-dimensional space should be proportional to the distances between the projected points in the low-dimensional space. In the current study, the long distance in the original dimensional space represents a high degree of dissimilarity in protein structure and the related two data points are more likely to be in different protein folding states. A well-behaved dimensionality reduction method should keep this correspondence in the low-dimensional space. Several assessments are applied to quantify this relationship. Spearman’s rank-order correlation coefficient is calculated to test the linear relationship of pair-wised distances of data points. A potential problem is that distances are not independent. Rather, the change in position of one residue would lead to the change in the related $n-1$ pair-wised distances. Therefore, to overcome this problem, the Mantel test is used to randomly permute rows and columns of distance matrix. The result of the Mantel test showed a similar trend as that in the Spearman correlation coefficient, which indicates that all methods are free from the dependency of distances and maintain good stability. The concept of the Shannon information in information theory is utilized to compare the information content in each projection space. The results of the above criteria show that ivis is capable of effectively separating different classes in the low-dimensional space and preserving high-dimensional distances with the least information loss. While the high-dimensional data set is usually projected onto a 2D surface, the effectiveness of MSMs on different dimensions was tested. Through the results of GMRQ, different methods showed various results. It is proposed that suitable dimensions are dependent on the biological system and dimensionality reduction method. However, two-dimensional space is still desired for visualization purposes if it can represent sufficient biological information.

The protein contact map demonstrates the superiority of the ivis dimensionality reduction method that ivis can retain both local and global information. Unexpectedly, the asymmetrical nature of the AuLOV dimer is revealed by comparing the protein–protein interactions. Several important residues are identified by the ivis framework. Met313, Leu331, and Cys351 have been reported as light-induced rotamers near cofactor FMN.22 These key residues are located on the surface of the β-sheet, which is consistent with and proves the concept of the signaling mechanism that signals originating from the core of Per-ARNT-Sim (PAS) generate conformational change mainly within the β-sheet.63,64 Gln365 is important for the stability of the $\alpha$ helix through hydrogen bonding with Cys316.65 Leu248, Gln250, and Asn251 were also found to be important in modulating allostery within a single chain, reported as the $\alpha'\alpha$ linker, while Asn329 and Gln350 function as FMN stabilizers.66 Through the AuLOV dimerization, $\alpha'\alpha$ and $\alpha\alpha$ helices undergo conformational changes and are expected to account for large importance, as shown in Table 2. However, the protein linkers, as well as the $\alpha\alpha$ helix and H/J$\beta$ and I$\beta$ strands, also showed high importance. The significance of protein linkers in the current study is consistent with both experimental and computational findings69−72 that protein linkers are indispensable components in allostery and biological functions. Together, these unexpected structures are vital in AuLOV allostery and worth further study. Overall, several key residues and secondary structures identified by the ivis framework agree with the experimental finding, which consolidates the good performance of ivis in elucidating the protein allosteric process.

Computational cost should be considered when comparing dimensionality reduction methods, since it is computationally expensive for large data sets, especially for proteins. From this perspective, different models are benchmarked using a dummy data set. Results showed that PCA requires the least computational resource, not subjected to either sample size or feature size. This might be due to the reason that PCA implemented in Scikit-learn uses SVD for acceleration. Further, since the size of the data set was large, randomized truncated SVD was applied to reduce the time complexity to $O(\tilde{n}_m \tilde{n}_c)$ with $n_{\text{max}} = \max (n_{\text{samples}}, n_{\text{features}})$.73 While t-SNE is comparable with ivis regarding several assessments, the computational cost could be prohibitively expensive for large data sets as t-SNE has a time complexity of $O(n^2)$ where $N$ and $D$ are the number of samples and features, respectively. Though tree-based algorithms have been developed to reduce the complexity to,74,75 it is still challenging for the high-dimensional protein system. ivis exhibited less computational cost in higher sample sizes and dimensions. Further, as shown in Figure 11, the loss of the ivis model converges fast and the overall computational cost could be further reduced with early stopping iterations. Combined with the performance criteria and runtime comparison, the ivis framework is demonstrated as a superior dimensionality reduction method for protein systems and can be an important member in the analysis toolbox for the MD trajectory.

## CONCLUSIONS

As originally developed for single-cell technology, the ivis framework is applied in this study as a dimensionality reduction method for molecular dynamics simulations for biological macromolecules. ivis is superior to other dimensionality reduction methods in several aspects, ranging from preserving both local and global distances, maintaining similarity among data points in high-dimensional space and projections, to retaining the most structural information through a series of performance assessments. ivis also shows great potential in interpreting biological systems through the feature weights in the neural network layer. Overall, ivis reached a balance between dimensionality reduction performance and computational cost and is therefore promising as an effective tool for the analysis of macromolecular simulations.

## AUTHOR INFORMATION

### Corresponding Author

**Peng Tao** — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75205, United States; [orcid.org/0000-0002-2488-0239]; Email: ptao@smu.edu

### Author

**Hao Tian** — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75205, United States; [orcid.org/0000-0002-0186-9811]

---

https://dx.doi.org/10.1021/acs.jcim.0c00485

J. Chem. Inf. Model. 2020, 60, 4569−4581
Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jcim.0c00485

Notes
The authors declare no competing financial interest.

ACKNOWLEDGMENTS
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. Computational time was generously provided by Southern Methodist University’s Center for Research Computing. The authors thank Xi Jiang from the Biostatistics Ph.D. program in the Statistics department of SMU for fruitful discussions.

REFERENCES
(22) Heintz, U.; Schlitching, I. Blue Light-Induced LOV Domain Dimerization Enhances the Affinity of Aureochrome 1a for Its Target DNA Sequence. eLife 2016, 5, No. e11860.


A.2. PAPER II

Deciphering the Allosteric Process of the *Phaeodactylym tricornutum* Aureochrome 1a LOV Domain

Published as part of *The Journal of Physical Chemistry* virtual special issue “Machine Learning in Physical Chemistry”.

Hao Tian, Francesco Trozzi, Brian D. Zoltowski, and Peng Tao*

**ABSTRACT:** The conformational-driven allosteric protein diatom *Phaeodactylym tricornutum* aureochrome 1a (PtAu1a) differs from other light–oxygen–voltage (LOV) proteins for its uncommon structural topology. The mechanism of signaling transduction in the PtAu1a LOV domain (AuLOV) including flanking helices remains unclear because of this dissimilarity, which hinders the study of PtAu1a as an optogenetic tool. To clarify this mechanism, we employed a combination of tree-based machine learning models, Markov state models, machine-learning-based community analysis, and transition path theory to quantitatively analyze the allosteric process. Our results are in good agreement with the reported experimental findings and reveal a previously overlooked Cα helix and protein linkers as important in promoting the protein conformational changes. This integrated approach can be considered as a general workflow and applied on other allosteric proteins to provide detailed information about their allosteric mechanisms.

**INTRODUCTION**

Light–oxygen–voltage (LOV) domains are a subdivision of the Per-Arnt-Sim (PAS) superfamily that are sensitive to blue light and undergo conformational as well as dynamical changes upon light activation. This activation begins with the formation of a covalent bond between a cofactor and a conserved cysteine residue. Possible cofactors include flavin adenine dinucleotide (FAD), flavin mononucleotide (FMN), or riboflavin. This covalent bond further promotes the overall structural changes, resulting in the alteration of the protein–protein interactions and thus signal transduction.

*Phaeodactylym tricornutum* aureochrome 1a (PtAu1a) is a recently discovered LOV protein that consists of an unstructured N-terminal region and a basic region leucine zipper (bZIP) DNA-binding domain connected to a C-terminal LOV core. The LOV domain, together with two flanking helices (A’α and Jα), is usually referred to as AuLOV. The protein is dynamically stable in the dark state due to the interaction between the LOV core and bZIP. This interaction prohibits the protein binding with DNA. A photoinduced covalent bond is formed between the C4a position of the cofactor FMN and a nearby sulfur in Cys287. This covalent bond triggers a series of conformational changes, including the undocking and unfolding of the Jα helix from the LOV core surface, the release of the A’α helix from the hydrophobic site on the LOV domain surface, and dimerization of the LOV domains.

These events lead to the increase of PtAu1a affinity for DNA binding and are proposed to be allosteric. Recent research has revealed that a combination of structural changes in the LOV core and the undocking of Jα helix are essential for the release of the A’α helix and LOV domain dimerization. The allosteric mechanism in PtAu1a is considered to be different from other LOV proteins, since the location of the LOV domain is in the C-terminus in PtAu1a while in the N-terminus in others. This structural difference raises the question on allosteric transmission in PtAu1a.

Various computational methods have been applied to explore protein allosteric mechanisms at the atomic level. Molecular dynamics (MD) simulations are capable of providing atomic-scale information, as well as structure–function relationships, and are widely used in sampling protein motions and structure landscapes. The significant
computational power provided by graphical processing units (GPUs) has promoted the time scale of MD simulations from nanoseconds to milliseconds.\textsuperscript{18,19} To obtain more biologically meaningful information from trajectories, Markov state models (MSMs) are often used to extract asymptotic kinetic information based on limited simulations.\textsuperscript{20,21} Kinetically separate macrostates can be obtained from MSMs in the reduced dimension. Differences among these subspaces can then be quantified to gain insight into protein structure and function relations.

The success of MSMs depends on appropriate dimensionality reduction methods that can preserve global distances while retaining the most structural information.\textsuperscript{22} New dimensionality reduction methods have been developed to project the high-dimensional trajectories to lower dimensions for thorough study. However, many methods, such as principal component analysis (PCA),\textsuperscript{23} time-structure-based independent component analysis (t-ICA),\textsuperscript{24} and the t-distributed stochastic neighbor embedding (t-SNE) method,\textsuperscript{25} suffer from problems including maintaining the similarity between high-dimensional space and low-dimensional space and are not resistant to system noise.\textsuperscript{26} In the current study, MD simulations were projected onto a 2D space via the ivis framework,\textsuperscript{27} which is a nonlinear method based on Siamese neural networks (SNNs) and has been shown to be powerful in interpreting biological systems.\textsuperscript{28}

Machine learning has recently achieved great accomplishments in chemistry and biology. Raccuglia et al. applied machine learning algorithms trained on failed experimental data to predict reaction results with high accuracy.\textsuperscript{29} Faber et al. employed machine learning techniques for feature vector representations of crystal structures.\textsuperscript{30} Botu et al. integrated a machine learning framework to accelerate \textit{ab initio} molecular dynamics simulation.\textsuperscript{31} The broad applications of machine learning stem from the ability to process large data sets and, more importantly, provide explanatory details.\textsuperscript{12,32} These favorable metrics offer a new prospective direction for the research on protein allostery. In this study, two tree-based machine learning models, random forest (RF) and one-vs-one random forest (OvO RF), were used to study the structural differences between macrostates and determine the contribution of residues to the allosteric process. In combination with machine learning and dynamic community analysis, Zhou et al. developed a new approach, known as machine-learning-based community analysis,\textsuperscript{33} to identify important structural features in dynamically driven protein allostery. Here, we applied this method on AuLOV and demonstrated the feasibility of this method in analyzing conformational-driven protein allostery.

The AuLOV is investigated in this study through the MD simulations, tree-based machine learning models, machine-learning-based community analysis, and transition path theory. Our results identified key residues that are consistent with experimental discoveries and suggested the importance of the Cα helix, overlooked thus far. Moreover, we quantified the important role of N- and C-terminal linkers in modulating AuLOV allostery. The integrated methods determined the importance of each residue in the allosteric process and therefore provided new insights into the allosteric mechanisms, which may promote future research on PtAu1a as an optogenetic tool.

## METHODS

### Molecular Dynamics (MD) Simulations

The initial structures of the native dark-state monomer and native light state of the AuLOV dimer were taken from the Protein DataBank (PDB)\textsuperscript{35} with the PDB ID being 5dkk for the native dark state and 5dkl for the native light state. To keep the same number of residues in all structures, the longest common residue sequences (from Ser240 to Glu367) were modeled. Both the native dark and light structures contain FMN as a cofactor. The force field for the cofactor FMN was used from a previous study.\textsuperscript{36} In order to fully explore the protein dynamics with regard to the formation of the covalent bond between cysteine 287 and FMN, two new transient states, referred to as a transient dark state and transient light state, were generated. Specifically, the transient dark state was generated by forming the cysteinyl-flavin C4a adduct bond in the native dark-state structure. The transient light state was generated by removing the cysteinyl-flavin C4a adduct bond and constructing the dark-state configuration in the native light-state structure. These transient structures facilitate analysis of allosteric interconversion between the light- and dark-state structures.

The crystal structures were added with hydrogen atoms and were further solvated in a water box with the TIP3P water molecules.\textsuperscript{37} Sodium cations and chloride anions were added for charge neutralization. For each structure, energy minimization was done with the steep descent method and the adopted basis Newton–Raphson minimization. The system temperature was raised to 300 K through a 20 ps MD simulation. Another 20 ps simulation was done for equilibrium. Ten ns of isothermal–isobaric ensemble (NPT) followed by 1.1 μs of canonical ensemble (NVT) Langevin MD simulations were carried out at 300 K. The first 0.1 μs of NVT simulations was considered as an equilibration stage and was discarded. Three NVT MD simulations were conducted independently for each protein structure. Therefore, a total of 12 μs simulations were generated for analysis. The SHAKE method was used to constrain all bonds associated with hydrogen atoms. A 2 fs step size was used for all MD simulations. Trajectories were saved for every 100 ps. A periodic boundary condition (PBC) was applied in simulations. The particle mesh Ewald (PME) algorithm\textsuperscript{38} was used to calculate the electrostatic interactions. MD simulations were conducted using GPU accelerated OpenMM\textsuperscript{39} and CHARMM27 force field.\textsuperscript{40}

### Analysis of Simulation Trajectories

#### Root-Mean-Square Deviation (RMSD) and Root-Mean-Square Fluctuation (RMSF)

The dynamics stability of an MD simulation trajectory is measured by the root-mean-square deviation, which is calculated as

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (r_{0i} - Ur_{i})^2}$$  \hspace{1cm} (1)$$

where $r_{0i}$ represents the coordinate of an atom $i$ in a Cartesian coordinate system, and $U$ is the most appropriate alignment transformation matrix between two structures. For each trajectory, the first frame was treated as the reference structure. The root-mean-square fluctuation is used to measure the fluctuation of atoms in each frame with regard to the first frame in an MD simulation trajectory. Specifically, Cα atoms were considered important in representing the protein motions, and the corresponding RMSFs of each Cα were calculated as...
$$\text{RMSF}_i = \sqrt{\frac{1}{T} \sum_{j=1}^{T} (\tau_i(t) - \overline{\tau}_i)^2}$$

(2)

where $T$ is the number of frames, and $\overline{\tau}_i$ is the averaged Cartesian coordinate of the $i^{th}$ Ca in the given trajectory.

**Feature Processing.** The 3N degrees of freedom in the Cartesian coordinate system hinder a thorough analysis of MD simulations in biological systems. Pairwise Ca distances are usually extracted to represent the structural characteristics of protein configurations. In the current study, a feature vector of each structure was constructed by calculating the distance pairs between one $\alpha$ carbon atom and other $\alpha$ carbon atoms in amino acids following the order of residue sequence. This feature vector was further encoded by a previously proposed transformation method with a cutoff of 10 Å.

**ivis Dimensionality Reduction Method.** ivis is a machine-learning-based dimensionality reduction method that is originally developed for single cell technology. The ivis framework applies the Siamese neural network architectures that are composed of three identical base neural networks. For each base neural network, there are three dense layers consisting of 500, 500, and 2000 neurons with a final embedding layer of two neurons. A novel triplet loss function is implemented in the training process

$$L_{\text{triple}}(\theta) = \frac{1}{m} \sum_{a,p,n} \left[ \min(D_{a,p}, D_{p,n}) + m \right]$$

(3)

The symbol $a$ represents the point of interest, often referred to as anchor point. The symbol $p$ represents a positive point that is selected based on the $k$-nearest neighbors (KNNs) algorithm. The symbol $n$ represents a negative point that is randomly selected from the rest of data samples. The similarity between two points is calculated as the Euclidean distance ($D$). The margin ($m$) is defined as the minimum distance between any pair of points and was set to default value of 1. The advantage of the ivis method lies in the triplet loss function, which aims to minimize the distance between the anchor points and the positive points while maximizing the distance between the anchor points and the negative points.

An Adam optimizer with a learning rate of 0.001 was applied to train the neural network. To prevent overfitting, early stopping of 5 was used to terminate the training iteration if the triplet loss function does not decrease with five consecutive epochs.

**Machine Learning Methods. Random Forest (RF) and One-vs-One (OvO) Random Forest Models.** Random forest as a tree-based machine learning technique was applied to learn the structural differences among macrostates in this study. Each random forest model is composed of 50 decision trees. Decision trees were trained individually, and the final result of a random forest model is formed by a voting algorithm. Scikit-learn version 0.20.1 was used to implement the random forest model.

The random forest model overcomes the problem of overfitting by employing several decision trees. However, in multitask classification jobs, an OvO random forest model is more common and superior than the random forest model by constructing one classifier for each pair of classes. The overall output is the weighted sum of all base classifiers. In the current study, 10 macrostates were trained with 45 random forest models. The OvO random forest model provides a weighted sum of overall feature importance with specific feature importance regarding two given classes.

**Feature Importance.** The feature importance in a random forest model is calculated using the Gini impurity, which is calculated as

$$\text{Gini impurity} = \sum_{i=1}^{C} -f_i(1 - f_i)$$

(4)

where $f_i$ and $C$ are the frequency of one label at a node that are chosen to divide the data set and the number of labels, respectively. A random forest model consists of multiple decision tree models. The importance of feature $i$ in each decision tree is calculated as

$$F_i = \frac{\sum_{j \text{in all decision trees}} \text{norm} f_j}{N}$$

(6)

where norm $f_j$ and $N$ are the normalized feature importance of one decision tree and the number of decision trees, respectively.

Pairwise Ca distances were extracted as the input features, and the corresponding feature importances were calculated. For each Ca distance, the importance was added to the related two residues. The accumulated feature importance of residues implies their contributions in the allosteric process.

**Markov State Model.** The long-time-scale protein dynamics is tracked by the Markov state model. Each simulation frame is assigned to different microstates through a MiniBatch k-means clustering method. Compared with microstates, macrostates are more biologically meaningful, as they are considered as kinetically separate equilibrium states. Ten macrostates were generated by Perron cluster cluster analysis (PCCA). Lag time is needed to build a MSM and was determined as 40 ns based on the implied relaxation time scale. Transition matrix and corresponding transition probabilities were estimated based on this MSM. MSMBuilder package (version 3.8.0) was used to build MSMs.

**Machine-Learning-Based Community Analysis.** Machine-learning-based community analysis is a newly proposed method by Zhou et al., which groups residues into communities. For an undirected graph characterizing the protein, nodes can be used to represent residues, and edges can be used to represent the distance between residues $\alpha$ carbon atoms. For each $\alpha$ carbon atom, the inner edges of $i$ are defined as the summation of edge values between node $i$ and any other node in $C_{ni}$ whereas the external edges of $i$ are defined as the summation of edge values between node $i$ and any other node in other communities. For each iteration of an ML communities partition, node $i$ can be moved to another community or swapped with another node in different communities. The benefit of these two explorative moves can be calculated as the external edges subtracted by the inner
edges. The algorithm of this community analysis method is listed below.

1. ML communities are randomly partitioned;
2. The benefits of moving one node into another community and swapping one node with another between different communities are estimated to search for the maximum moving and swapping strategy, respectively;
3. One moving or swapping strategy with the highest benefit is chosen;
4. Repeat steps 2 and 3 with the new ML community configuration until the highest benefit of all moving and swapping strategies is less than 0;
5. ML communities construction is completed if any strategy will increase the number of inner edges for each ML community.

The Kernighan–Lin algorithm has been implemented to search for local minimum values in graph theory. In the current research, the feature importances of Cα distances from the OvO random forest model based on AuLOV dimer simulations were used. In order to apply ML-based community analysis on a monomer, the averaged importance for each Cα distance in a monomer was calculated based on the dimer feature importance results.

**Transition Path Theory.** Transition path theory (TPT) is used to identify the most probable routes from one macrostate to another. A dark state and light state were chosen based on the transition probability estimated in MSMs as initial and final states, respectively. All other states are considered as intermediate states. Possible transition paths from the dark state to the light state were simulated. The definition of the committor probability $q_i^+$ is the probability from one state to a target state. Based on this definition, $q_i^+$ is equal to 0 for all microstates in an initial state, and $q_i^+$ is equal to 1 for all microstates in a final state. The committor probability of other microstates is calculated as

$$q_i^+ = \sum_{k \in I} T_{ik} q_k^+ - \sum_{k \in \text{target state}} T_{ik}$$  

where $T$ is the transition probability matrix, and $T_{ik}$ represents the transition probability from state $i$ to state $k$.

$$f_{ij} = \pi \bar{q}_i^- T_{ij} q_j^+$$  

where $\pi$ is the stationary probability of $T$, and $\pi T_{ij}$ is the backward committor probability calculated as $q_i^- = 1 - q_i^+$. The backward flux $f_{ij}$ was also considered and subtracted in calculating the net flux $f_{ij}^n = \max(0, f_{ij} - f_{ji})$.

---

**Figure 1.** Native dark and light structures of the AuLOV monomer. (A) Native dark structure. (B) Native light structure. (C) A covalent bond is formed between the C4a position of FMN and sulfur atom in Cys 287 upon light excitation.

**Figure 2.** RMSDs of AuLOV MD simulation trajectories. (A) Native dark and transient dark states. (B) Native light and transient light states.
The flux from the initial state to the final state can be decomposed to individual pathways $p_i$, which can be calculated as

$$ p_i = \frac{f_i}{\sum_j f_j} \quad (9) $$

## RESULTS

**MD Simulations Analysis.** The native dark and light structures of AuLOV are illustrated in Figure 1. In the native light state, a covalent bond is formed between the C4a atom in FMN and the sulfur atom in residue Cys287 upon light excitation (Figure 1C). This covalent bond triggers a global conformational change and protein dimerization. To explore this effect and the protein response, the covalent bond between FMN and Cys287 is constructed in the dark-state structure to construct a transient dark state. On the other hand, the covalent bond between FMN and Cys287 is removed in the light-state structure to construct the transient light state. Both transient dark and transient light states are subjected to the simulation and analysis to aid in mapping allosteric trajectories in response to blue-light activation and thermal reversion to the dark state.

The time evolution of the RMSD in four trajectories is plotted in Figure 2. All RMSD values were calculated with reference to the first frame of each trajectory. The average RMSDs in native dark, native light, transient dark, and transient light states are 1.75, 2.04, 2.39, and 2.08 Å, respectively. The plots show that each simulation is stable with low RMSD fluctuation values. The transient dark state is more dynamically active than the native dark state, indicating that the formation of a covalent bond increases the flexibility of
the protein. The RMSD results also imply the stability of the native dark state compared with the native light state.

As the allosteric process of AuLOV is characterized by conformational changes in the secondary structures, the backbone $\alpha$ atoms is selected to measure the influence of light absorption on the protein structure. The RMSFs of the $\alpha$ atoms in AuLOV simulations are calculated and plotted in Figure 3. Both $A'\alpha$ and $J\alpha$ helices were found to be dynamically active in all four states with increased dynamics in the two transient states. Differences between the two chains can be further quantified by comparing the RMSF values. In the native dark state, chains A and B showed no difference. In the native light state, the $A'\alpha$ in chain A is more flexible than that in chain B. Through the formation of the covalent bond in the transient dark state, both $A'\alpha$ and $J\alpha$ helices in chain A showed enhanced flexibility.

**Markov State Model Partitions Kinetically Separate Macrostates.** To represent the protein structure and movements, pairwise $\alpha$ distances were calculated as the representation of protein configurations. A total of 32,131 $\alpha$ distances were extracted from the AuLOV dimer, composed of 254 residues. For each $\alpha$ distance, the value was further encoded through the feature preprocessing method outlined in the methods. For feature transformation, 10.0 Å was chosen as the threshold. The vis dimensionality reduction method was applied to extract the collective variables and project the embedding layer onto a 2D surface. The distribution of four states in the vis result is plotted in Figure 4A. The plot revealed that the transient dark state partially overlaps with the native dark state and the transient light state. The large region of the transient dark-state distribution is mainly because of the enhanced dynamics caused by the formation of the covalent bond. The distribution of the native light state is divided into two separate regions. The distribution of the transient light state covers a large area and overlaps with the both regions of the native light-state distribution.

The Markov state model is based on the clustering results on the reduced dimension projected by an vis framework. To construct MSMs, a MiniBatch $k$-means clustering method was applied to partition the distribution of protein simulations in the 2D region into 300 microstates. The top 20 relaxation time scales calculated by different MSMs with different lag times are plotted in Figure 4B. The implied time scale converges after 40 ns, which was chosen as the lag time for an MSM. The number of macrostates depends on the gap between the time scales, and a total of 10 macrostates were chosen to divide the reduced protein distribution into kinetically separated macrospaces. For each microstate, the corresponding labels of macrostates were determined by the PCCA method, which is based on the eigenfunction of the transition probability matrix in MSM. The resulting macrostates with their associated transition probabilities are illustrated in Figure 5. Two dark states and two light states are divided into four and six macrostates, respectively. Macrostates (States 1, 2, 3, and 10) are in the area of the native dark state and the transient dark state. Based on the similarity to the crystal dark-state structure, Macrostate 2 was considered as the native dark state. State 9 was recognized as the native light state using the same method. Other macrostates were considered as intermediate states. The low transition possibilities starting from Macrostates 2 and 9 to adjacent macrostates indicate the stability of both the dark and light states. On the contrary, it is more likely for a protein to shift between intermediate states. Two representative structures in the transient dark and transient light states are illustrated in Figure 6. Both $A'\alpha$ and $J\alpha$ helices in the representative conformation of Macrostate 3 (transient dark state, Figure 6A) move further from the LOV domain compared to the native dark state (transparent gray structure in Figure 6A). These differences agree with the experimental finding that the $J\alpha$ helix interacts with the LOV core through hydrogen bonding between Gln365 and Cys316 as well as Tyr357 and Gln330 in the native dark state. In the light state, the hydrogen bond between Gln365 and Cys316 is broken after the formation of a photoinduced covalent bond between FMN and Cys287, leading to the release of a $J\alpha$ helix from the LOV core. The $A'\alpha$ helix also interacts with the LOV core via a hinge region (Ala248, Glu249, Glu250, and Gln251) and covers a hydrophobic patch (as the back of $A'\alpha$ helix) in the native dark state. Due to the change of the $A'\alpha$ helix orientation, the back of this helical structure as the hydrophobic patch is exposed in the light state. The protein structure of Macrostate 5 (transient light state, Figure 6B) is similar to the structure of the native light state due to the stabilizing interaction within the dimer structure.

**OvO Random Forest Model Extracts Key Residues.** In order to extract the key residues that play a vital role in AuLOV allostery, supervised machine learning models were applied to explore the structural differences among macrostates. Here, pairwise $\alpha$ distances were chosen as the translation and rotation invariant collective variables for the description of protein structures in the simulations. For each simulation, frames were saved for every 100 ps, resulting in 10,000 frames for every 1 μs MD trajectory. Accordingly, 120,000 samples with 32,131 features were extracted from the simulated trajectories. Each frame was labeled based on the macrostate results. Random forest and OvO random forest models were applied to distinguish the intrinsic conformational differences among macrostates. Training scores and testing scores were plotted in Figure 7. The testing accuracy was 93.5% in the random forest model at depth 9 and 94.5% in the OvO random forest model at depth 8. The high classification accuracy indicated that the two tree-based models were able to
capture the characteristics of protein configuration of each macrostate using pairwise Cα distances.

The advantage of the tree-based models comes from the ability to quantitatively evaluate the contribution of each feature in a classification model through the value of feature importance. Superior to the random forest model, the OvO random forest model was applied to compute the feature importance for any two different macrostate pairs by conducting a random forest classification between these two specific macrostates. Therefore, for any two different macrostates, one distinct random forest estimator was built. A combination of $10 \times 9 / 2 = 45$ basic random forest classifiers were constructed for the pairwise macrostate classifications. The accumulated feature importance of OvO random forest at depth 8 was plotted in Figure 7C. Overall, this method is an effective model, in which the top 550 features account for 90.2% of the overall feature importance.

Those Cα distances related to two residues located on different chains are named as cross-monomer features. These cross-monomer features account for 59.77% of the overall importance. The Cα distances within the same chain accounts for 40.23% of the overall importance. This shows that the OvO random forest can capture the structural changes within each monomer as well as the relative motions between monomers.

In order to identify key residues based on the results of the OvO random forest model, the feature importance value of each Cα distance was added and accumulated to the two related individual residues. The top 20 residues are listed in Table 1. Among the identified residues, several have been experimentally confirmed to be important to AuLOV allostery and are shown in bold font. Residues Met313, Phe331, and Cys351 are found to undergo changes in orientation. Ala248, Gln249, Gln250, and Asn251 are residues linking the A′α helix to the Aβ strand that are important for signal transduction.8 Gln350 was also identified as essential for signal transduction in LOV domains, where it either undergoes a Gln-flip process in response to N5 protonation55 or undergoes rotation
between exposed and buried conformations to relay the signal from the flavin active site to N- or C-terminal components. We also identify Phe252 as important for allostery. Notably, Phe252 was found by HDX-MS to be important in the destabilization of the A′α helix that is coupled to conformational changes in the Bβ strand and Cα helix. Therefore, the OvO random forest can successfully identify important residues reported in experimental results. The residue importance can be accumulated to the protein’s secondary structures, and the results were shown in Table 2.

Table 2. Accumulated Feature Importance of Secondary Structures in AuLOV

<table>
<thead>
<tr>
<th>secondary structure</th>
<th>importance percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A′α</td>
<td>15.17%</td>
</tr>
<tr>
<td>Aβ</td>
<td>7.12%</td>
</tr>
<tr>
<td>Bβ</td>
<td>2.28%</td>
</tr>
<tr>
<td>Cα</td>
<td>9.12%</td>
</tr>
<tr>
<td>Dα</td>
<td>0.70%</td>
</tr>
<tr>
<td>Eα</td>
<td>0.05%</td>
</tr>
<tr>
<td>Fα</td>
<td>2.14%</td>
</tr>
<tr>
<td>Gβ</td>
<td>6.43%</td>
</tr>
<tr>
<td>Hβ</td>
<td>14.13%</td>
</tr>
<tr>
<td>Iβ</td>
<td>13.46%</td>
</tr>
<tr>
<td>Jα</td>
<td>8.04%</td>
</tr>
<tr>
<td>linkers</td>
<td>21.36%</td>
</tr>
</tbody>
</table>

A′α and Jα helices account for 15.17% and 8.04% of the overall importance, respectively. The importance of Cα helix and linkers in AuLOV are also significant at 9.12% and 21.36%, respectively.

**Machine-Learning-Based Community Analysis Splits Protein Structure into Four Communities.** To explore the significance of different protein secondary structures, machine-learning-based community analysis was applied to split the protein structure into communities. This analysis was developed to divide residues into several communities (referred to as ML communities) so that the feature importance for pairwise Ca distances across different communities is at a maximum, while the feature importance within each community is at a minimum.

The relationship between the feature importance for pairwise Ca distances within ML communities and the number of ML communities is plotted in Figure 8A. Applying an elbow criterion, four ML communities were selected with the total feature importance within each ML community accounting for 0.50% and the total feature importance among ML communities accounting for 99.50%. Therefore, the changes among ML communities account for the dominant majority of the overall feature importance and are able to explain the changes between different communities. The changes within each ML community are ignored due to the negligible importance. By applying ML-based community analysis, dynamics in each protein structure can be attributed to the changes among partitioned ML communities.

The distribution of different communities, with a complete partition result corresponding to protein secondary structure, is shown in Figure 8B. Commu. A (blue) includes most of the A′α helix and Aβ strand, Commu. B (orange) includes the Jα helix with part of Gβ and Hβ strands on the LOV core, Commu. C (red) includes the Cα helix, part of the Fα helix, and linkers, and Commu. D (gray) includes part of the Fa helix as well as Gβ, Hβ, and Iβ strands.

The machine-learning-based community analysis offered additional information based on the selected four ML communities and the corresponding different regions in the protein structure during simulation. The accumulated overall feature importance among each ML community pair is listed in Table 3. Correlations between Commu. A, Commu. B, and the

![Figure 8. ML-based community analysis results of AuLOV. (A) Total feature importance among ML communities with regard to different number of communities. (B) Four ML communities named as Commu. A, Commu. B, Commu. C, and Commu. D are illustrated in blue, orange, red, and gray colors, respectively.](https://dx.doi.org/10.1021/acs.jpcb.0c05842)
rest of the protein accounted for 82.99% of the total feature importance. This is not surprising, since the A′α helix in Commu. A and the Jα helix in Commu. B are the most distinguishing structures, which undergo significant conformational changes from the native dark state to the native light state. Through the accumulated feature importance of ML communities, A′α and Jα helices are confirmed to convey significant allostery characteristics. Unexpectedly, the correlation between Commu. C and Commu. D accounts for 16.57% of the total feature importance. Several transitions between adjacent macrostate pairs have a significant contribution from Commu. C (Table 4). However, for transitions between nonadjacent macrostates, Commu. C accounts for less importance, which explains the difference between macrostate pairs.


<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>State 2 (dark) → State 3</td>
<td>7.26%</td>
<td>26.69%</td>
<td>4.01%</td>
</tr>
<tr>
<td>State 2 → State 10</td>
<td>6.05%</td>
<td>18.51%</td>
<td>3.03%</td>
</tr>
<tr>
<td>State 3 → State 5</td>
<td>16.91%</td>
<td>12.74%</td>
<td>7.68%</td>
</tr>
<tr>
<td>State 5 → State 7</td>
<td>7.44%</td>
<td>2.04%</td>
<td>18.41%</td>
</tr>
<tr>
<td>State 5 → State 4</td>
<td>9.55%</td>
<td>6.40%</td>
<td>3.64%</td>
</tr>
<tr>
<td>State 5 → State 6</td>
<td>10.96%</td>
<td>6.98%</td>
<td>16.27%</td>
</tr>
<tr>
<td>State 7 → State 6</td>
<td>11.22%</td>
<td>3.97%</td>
<td>8.86%</td>
</tr>
<tr>
<td>State 6 → State 9 (light)</td>
<td>12.35%</td>
<td>10.56%</td>
<td>19.13%</td>
</tr>
<tr>
<td>nonadjacent macrostates</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State 10 → State 9</td>
<td>21.39%</td>
<td>23.63%</td>
<td>0.23%</td>
</tr>
<tr>
<td>State 10 → State 6</td>
<td>26.97%</td>
<td>15.11%</td>
<td>2.26%</td>
</tr>
</tbody>
</table>

“State transitions with a large Commu. C component are shown in bold.

Table 5. Probability of Top 10 Channels Simulated Using Transition Path Theory

<table>
<thead>
<tr>
<th>channels</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 3, 5, 7, 6, 9</td>
<td>28.8%</td>
</tr>
<tr>
<td>2, 3, 5, 6, 9</td>
<td>25.6%</td>
</tr>
<tr>
<td>2, 10, 5, 7, 6, 9</td>
<td>5.4%</td>
</tr>
<tr>
<td>2, 3, 7, 6, 9</td>
<td>4.7%</td>
</tr>
<tr>
<td>2, 3, 5, 8, 5, 7, 6, 9</td>
<td>3.9%</td>
</tr>
<tr>
<td>2, 3, 4, 8, 5, 7, 6, 9</td>
<td>3.0%</td>
</tr>
<tr>
<td>2, 3, 8, 5, 7, 6, 9</td>
<td>2.9%</td>
</tr>
<tr>
<td>2, 10, 5, 6, 9</td>
<td>2.3%</td>
</tr>
<tr>
<td>2, 1, 3, 5, 7, 6, 9</td>
<td>2.1%</td>
</tr>
<tr>
<td>2, 10, 5, 8, 7, 5, 6, 9</td>
<td>1.3%</td>
</tr>
<tr>
<td>top 10 channels</td>
<td>80.0%</td>
</tr>
</tbody>
</table>

For those transitions between macrostates where Commu. C accounts for a large component, two promising routes from the dark state to the light state can be identified as (1) State 2 → 3 → 5 → 7 → 6 → 9 and (2) State 2 → 3 → 5 → 6 → 9. These two proposed pathways lead to a hypothesis that Commu. C is important in propagating allosteric perturbations.

To estimate the probability of the two identified channels that include a significant Commu. C contribution, the transition path theory was employed to generate an ensemble of pathways to estimate the probability of every pathway from State 2 (native dark state) to State 9 (native light state). A total of 3 151 pathways were generated and divided as 212 distinct channels connecting these 2 states. The probability of each channel was calculated based on the net flux from the initial state to the target state. Overall, the probability of the top 10 channels is listed in Table 5. The population of these 10 channels account for 80.0% of the total pathway population.

Among all 212 channels, the 2 identified channels 2→3→5→7→6→9 and 2→3→5→6→9 are the top 2 populated channels with 28.8 and 25.6% of overall probability, respectively. The sum of contributions from these top two channels accounts for 54.4% of the contributions, which is significant compared to all other pathways, suggesting the importance of Commu. C movement during the allosteric process. The first channel is more probable than the second channel. This agrees with the observation that the transition probability from 5 to 7 (9.5%) as one step in the first channel is greater than that from 5 to 6 (1.6%) as one step in the second channel. Interestingly, the ML-based community analysis reveals a higher contribution from the Commu. C to the transition between states 5 and 7 than that between states 5 and 6.

Different communities account for different importance in each macrostate transition. To better show the trend of components in Commu. A, Commu. B, and Commu. C with regard to Commu. D, the change of importance along the two proposed paths is plotted in Figure 9. Two paths share similar characteristics: (1) Commu. A accounts for little importance at the beginning of allostery process, while the contribution goes up in later transitions; (2) Commu. B starts with high importance and decreases drastically after the first transition; (3) Commu. C is more important at the end of allostery process.

#### DISCUSSION

PtAu1α is an allostERIC protein that undergoes a series of conformational changes upon light activation beginning with the formation of a covalent bond between Cys287 and FMN.57 This computational study of AuLOV is integrated with MD simulations and other computational methods to provide quantitative analysis of the dynamics and importance of residues with regard to the overall allosteric process. While there is extensive research on the regulatory role of the Jα helix and dimerization controlling the A′α helix, a detailed mechanism of allostery with a signal transmission route still needs to be scrutinized.

Signal transduction in the LOV domain containing proteins typically involves coupling of adduct formation to conformational changes in the N- and C-termini via propagation across a central β-sheet.58–61 Central to this signal transduction are key residues within the Iβ strand that enable its coupling with the Jα helix and interaction with the A′α helix in the dark state, specifically the residue equivalent to Gln350 that is essential for LOV signal transduction.55,56 In AuLOV, several additional light-induced rotamers (Met313, Leu317, Leu331, and Leu333, and Cys351) were observed on the β-sheet surface.5 In our models, these residues contribute to conformational changes linking the β-sheet surface to the A′α helix through Gln350. We note that our computational methods mirror those identified experimentally where the A′α helix contributes to the dynamic stability of the dark state by the interaction with LOV core through a hinge region. The hinge region consists of four conserved residues (A1a248, Gln249, Gln250, and Asn251), which were also found
to be important via our approach (Table 1). Overall, the strong correlations between previous experimental results and our Markov state model and OvO random forest analysis confirm our methodology as being able to discern allosteric pathways in AuLOV.

Most proteins undergo an allosteric process within a long time scale from milliseconds to seconds, including AuLOV, making it difficult to collect sufficiently long trajectories. The Markov state model addresses this difficulty by extracting the slowest motion and long-time-scale information from limited simulations. However, while the slowest dynamical processes are often involved in protein allostery and are assumed to be the process of interest,

fast-moving flanking helices or side chain rotations could play a significant role in protein allostery. Due to their short time scales, these motions may not be represented well in the MSM. Although there are some studies focusing on fast protein motions and their relations with slow motions,

the functions of fast motions in protein allostery remain elusive and require more studies. Regarding the allosteroy of AuLOV, the kinetics between dark and light states remain elusive and require more studies. Regarding the allosteroy of AuLOV, the kinetics between dark and light states are beyond the scale of minutes.

Therefore, sub-ns protein local motions are unlikely to be determinant factors in the AuLOV allosteric mechanism and are not the focus of the present study.

Although chain A and chain B in AuLOV are dynamically identical in the dark state, the $A'$helix of the two chains differ in conformations upon dimerization. Our simulation results confirmed the differences between these two chains through a comparison of RMSF values. The RMSF results reflect that the $A'$helix in chain A is more dynamically active than that in chain B. The asymmetrical property in the $A'$helix could originate from either the interaction between $A'$helix and J$alpha$ helices on different chains or the asymmetrical conformational change, thus requiring further detailed study.

ML-based community analysis used in this study provided an approach to partition protein conformation into communities based on the feature importance of pairwise $Calpha$ distances. Through this analysis, three important communities were identified. Commu. A containing the $A'$helix and Commu. B containing the J$alpha$ helix were expected to account for a great contribution, since these two helices undergo notably conformational changes upon light activation (Table 2). The $Calpha$ and F$tau$ helices stand out as Commu. C and surprisingly provided additional information for the allosteric process. Commu. C accounts for great importance in adjacent transitions between macrostates and accounts for less importance in nonadjacent transitions compared with Commu. A and B.

Transition path simulations further validated the important allosteric function of Commu. C. For all possible transition pathways found by TPT, the top two channels are those with large Commu. C components and together constitute over 50% of the overall possibility. Although Commu. C consists of two helices as $Calpha$ and F$tau$, these two helices are not equally important. The allosteric role of F$tau$ helix should be evaluated with caution, since its accumulated feature importance is relatively low (Table 2), and the importance in Commu. D, which also includes part of F$tau$ helix, is the least important community. Because the $Calpha$ helix is important in both OvO random forest result and ML-based community analysis, it is reasonable to conclude that the $Calpha$ helix may play an important role in controlling AuLOV allostery. Moreover, Commu. C also includes several linking residues that account for a large portion of the overall importance, indicating the indispensable role of linkers in the allosteric process as reported in previous studies.

Examination of the two most probable channels linking conformational changes through the identified communities can allow construction of allosteric paths (Figure 9). In this study, we identify that the J$alpha$ helix is fundamental in the early stage of AuLOV allostery, followed by changes in the $A'$helix in later stages. In the first transition step from Macrostate 2 → Macrostate 3, Commu. B accounts for a large component compared with Commu. A, indicating the importance of the J$alpha$ helix in the initial stage of allostery. As the allosteric perturbation propagates, the importance of Commu. B decreases, and Commu. A becomes the more significant region. This important shift implied and confirmed the experimental finding that, after initial Cys287–FMN covalent bond formation, the first response of the protein structure is the undocking of the J$alpha$ helix, which is essential to the release of the $A'$helix. The rising importance of Commu. C, together with the transition path theory results, suggests that
Commu. C, especially the Cα helix and linkers, is vital in the allosteric process and should be investigated further.

**CONCLUSION**

The LOV protein PtAu1a is a member of Aureochrome family that binds DNA upon blue-light activation. Studies of the LOV domain with N- and C-terminal helices indicate that in the absence of light it exists as monomeric units; upon blue-light absorption, cysteinyl-flavin bond formation triggers a global conformational change that ultimately results in the dimerization of the LOV domains. In the present study, the protein dynamics of AuLOV with N- and C-terminal helices is simulated using MD simulations and analyzed using a series of computational methods. We quantified the differences of Aα and Jα helices dynamics in four functional states and the importance of each residue in the two chains with regard to the protein allosteric process. Key residues in overall structural changes identified by an OvO random forest agree with the results reported in other experimental work. The Markov state model, combined with transition path theory, studied the importance of protein structures by a machine-learning-based community analysis. The functional role of key Commu. C, which includes the Cα helix and linkers, is revealed through in-depth analysis as propagating the allosteric perturbation. Overall, this study quantitatively analyzed the allosteroly process of AuLOV and linked the macroscopic conformational change to residue level importance. Our results provided new opportunities for a detailed mechanism explanation and offered further opportunities for the research of PtAu1a as an optogenetic tool. Future studies can facilitate our understanding of global protein conformational changes in the context of full-length PtAu1a.

**AUTHOR INFORMATION**

**Corresponding Author**
Peng Tao — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75275, United States; orcid.org/0000-0002-2488-0239; Email: ptao@smu.edu

**Authors**
Hao Tian — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75275, United States; orcid.org/0000-0002-0186-9811

Francesco Trozzi — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75275, United States

Brian D. Zoltowski — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75275, United States

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcb.0c05842

**ACKNOWLEDGMENTS**

We gratefully acknowledge funding sources, including NIH research grant R15GM122013 to P.T. and NIH research grant R15GM109282 to B.D.Z. Computational time was generously provided by Southern Methodist University’s Center for Research Computing. The authors thank Ms. Xi Jiang from the Biostatistics Ph.D. program in the Statistics department of SMU for fruitful discussions.

**REFERENCES**


8. Heintz, U.; Schlüchting, I. Blue light-induced LOV domain dimerization enhances the affinity of Aureochrome 1a for its target DNA sequence. eLife 2016, 5, e11860.


15. Li, H.; Wu, H.; Li, B.; Gao, Y.; Zhao, X.; Zhang, L. Molecular dynamics simulation of fracture mechanism in the double interpenetrated cross-linked polymer. Polymer 2020, 199, 122571.


the blue light sensor phototropin 1 from Avena sativa. *Biochemistry* 2007, 46, 14001−14009.


(64) Huang, X.; Yao, Y.; Bowman, G. R.; Sun, J.; Guibas, L. J.; Carlsson, G.; Pande, V. S. Constructing multi-resolution Markov state models (MSMs) to elucidate RNA hairpin folding mechanisms. *Biocomputing 2010* 2009, 228−239.


A.3. PAPER III

Deciphering the protein motion of S1 subunit in SARS-CoV-2 spike glycoprotein through integrated computational methods

Hao Tian and Peng Tao

Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, TX, USA

Communicated by Ramaswamy H. Sarma

ABSTRACT
The novel severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) is a major worldwide public health emergency that has infected over 8 million people. Spike glycoprotein, especially the partially open state of S1 subunit, in SARS-CoV-2 is considered vital for its infection with human host cell. However, the mechanism elucidating the transition from the closed state to the partially open state still remains unclear. In this study, we applied a series of computational methods, including Markov state model, transition path theory and random forest to analyze the S1 motion. Our results showed a promising complete conformational movement of the receptor-binding domain, from buried, partially open, to detached states. We also estimated the transition probability among these states. Based on the asymmetry in both the dynamics behavior and the accumulated alpha carbon ($C\alpha$) importance, we further suggested a relation among chains in the trimer spike protein, which leads to a deeper understanding on protein motions of the S1 subunit.

Abbreviations: $C\alpha$: alpha carbon; MSM: Markov state model; NTD: N-terminal domain; PCCA: Perron-cluster analysis; PHEIC: Public Health Emergency of International Concern; RBD: receptor-binding domain; RMSD: root-mean-square deviation; RMSF: root-mean-square fluctuation; SARS-CoV-2: Severe Acute Respiratory Syndrome Coronavirus 2; SD: subdomain; $S$: protein: spike protein; TPT: Transition path theory; WHO: World Health Organization

Introduction
As of June 18 2020, there has been over 8 million confirmed cases and over 440,000 cases of death of the newly discovered coronavirus, named as SARS-CoV-2, according to the World Health Organization (WHO). The number of confirmed cases is still growing with the speed of over a hundred thousand per day. SARS-CoV-2 is related to bats-derived coronaviruses and the SARS-CoV reported in the Guangdong province of China in 2002, and identified as a new member of betacoronavirus (Lu et al., 2015; Wang et al., 2016). The S protein is a trimer (chain A, B and C) and each chain is formed by S1 and S2 subunits that are related with host receptors binding and membranes fusion, respectively (Li, 2015, 2016; Walls et al., 2020). The S1 subunit consists of an N-terminal domain (NTD), receptor-binding domain (RBD) and two subdomains (SD1 and SD2) (Wrapp et al., 2020). It is reported that RBD undergoes a conformational change from stable closed state to dynamically-less-favorable partially open state in chain A (Bosch et al., 2003; Li, 2016). In the closed state, the determinants of receptor binding are buried and inaccessible to receptors. But in the partially open state, they are exposed and expected to be necessary for the interaction with host cells (Gui et al., 2017; Pallesen et al., 2017). In the cases of SARS-CoV-2 and SARS-CoV, S glycoprotein is found to inherently sample the closed and open states. This behavior is suggested to exist in the most pathogenic coronaviruses (Shang et al., 2018; Walls et al., 2020). While the partially open state plays an important role in human cell infection, little study is done to illustrate this protein motion at residue level.

Molecular dynamics (MD) simulations can provide atomic scale information and are widely used in sampling protein movement and structure landscape (Prinz et al., 2011). Two kinds of trajectories of SARS-CoV-2 S protein initiating from the closed state (PDB ID 6VXX) and partially open state (PDB ID 6VYB) are available from D E Shaw Research (D. E. Shaw Research, 2020). However, the timescale (10 microseconds) is still relatively trivial compared with the timescale of biological processes in the real world. To gain more information...
from the result of MD simulation, Markov state model (MSM) is applied to obtain long-time kinetic information given time-limited simulation trajectories (Adelman et al., 2016; Suárez et al., 2016). One advantage of MSM is that it can divide a large number of protein structures from simulation into subspaces based on the extracted kinetic information. The differences among those spaces can be calculated and used for comparison.

Machine learning techniques have achieved great accomplishments in chemistry and biology including material discovery (Raccuglia et al., 2016), structure representation (Faber et al., 2015) and computation acceleration (Botu & Ramprasad, 2015). The great contributions from machine learning mainly come from its ability to deal with large scale data and its accurate and explainable models (Jing et al., 2018; Kotsiantis et al., 2007), which provide an opportunity to decipher protein dynamics. In this study, tree-based machine learning models were used to identify important residues. Specifically, random forest model was applied as a classification model to classify different structures and calculate the contribution of each residue and structure importance for the closed-open transition process.

The transition from the closed state to the partially open state of S1 subunits of SARS-CoV-2 S protein is investigated in this research through Markov state model, transition path theory and random forest. Our analyses provided the closed-open transition probability, showed a complete transition path from the closed to the open state, and identified a relationship between the motion of chain A and two other chains.

**Methods**

**Analysis of simulation trajectories**

The root-mean-square deviation (RMSD) is used to measure the overall conformational change of each frame with regard to a reference structure in a MD simulation trajectory. For a molecular structure represented by Cartesian coordinate vector $r_i$ ($i = 1 \text{ to } N$) of $N$ atoms, the RMSD is calculated as following:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (r_i^0 - U r_i)^2}$$

where $r_i^0$ represents the Cartesian coordinate vector of the $i$th atom in the reference structure. The transformation matrix $U$ is defined as the best fit alignment between the protein structures along trajectories with respect to the reference structure.

The root-mean-square fluctuation (RMSF) is used to measure the fluctuation of each atom in each frame with regard to a reference structure in a MD simulation trajectory.

$$\text{RMSF}_i = \sqrt{\frac{1}{T} \sum_{j=1}^{T} (v_i - \bar{v}_i)^2}$$

where $T$ is the total frames and $\bar{v}_i$ is the average position of atom $i$ in the given trajectory.

**Feature processing**

Distances between pairs of backbone $C\alpha$ were chosen as features to represent the protein configuration. The distances between each $C\alpha$ and all other $C\alpha$s were calculated. A protein contact map is formed by combining the pairedwise $C\alpha$ distances. Each element in the contact map is normally transformed into 1 if that value is below a threshold or 0 otherwise (Doerr et al., 2017). However, this feature preprocessing technique risks to ignore potentially useful spacial information forcing a boolean value on the features. Therefore, inspired by ReLU activation function (Nair & Hinton, 2010) in neural network, whose equation is shown below, we proposed a revised feature transformation method by transforming each feature value into 0 if that feature is above a threshold and keep it the same value otherwise. Compared with reference feature transformation rule, our proposed technique can still build a protein contact map while can differentiate local features with the least local information loss.

$$f(z) = \max(0, z)$$

**Random forest model**

Random forest is a machine learning technique that can be used for classification (Liaw & Wiener, 2002; Wang, Shen, et al., 2019). A random forest is composed of several decision trees (Utgoff, 1989), which are trained based on given training data. The final classification output of a random forest model is a collection of classes predicted by each decision tree model. The random forest algorithm carried out in this study is implemented in scikit-learn (Pedregosa et al., 2011) program package version 0.20.1. The number of decision trees used was 50. One advantage of random forest model over decision tree model is that employing multiple decision tree models mitigates the overfitting problem suffered by single decision tree model.

**Feature importance**

In a random forest model, a quantitative evaluation of the importance for each feature used for training is calculated through training process. This feature importance is calculated using Gini impurity:

$$\text{Gini impurity} = \sum_{i=1}^{C} -f_i (1 - f_i)$$

where $f_i$ is the frequency of a label at a node of being picked to split the data set and $C$ is the total number of unique labels. A random forest model is a collection of several decision tree models. The importance of node $i$ in decision tree is calculated as:

$$n_i = w_i C_i - \sum_{m=1}^{m} w_{m(i)} C_{m(i)}$$

where $w_i$ is the weighted number of samples reading node $i$, $C_i$ is the impurity value of node $i$ and $m$ is the number of child nodes. The feature importance of feature $i$ is calculated as
where $s$ is the number of times of node $j$ split on feature $i$. Feature importance within a decision tree is further normalized by:

$$\text{norm } f_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$$

(7)

The feature importance in random forest is the averaged importance of feature $i$ in all decision tree models:

$$F_i = \frac{\sum_{j=1}^{N} \text{norm } f_i}{N}$$

(8)

where $\text{norm } f_i$ is the normalized feature importance of one decision tree and $N$ is the number of decision trees (Breiman, 2001).

Feature importance of all pairwise Ca distances were calculated using the above methods. The feature importance of an amino acid is the summation of importance of features that are related with that amino acid. The relative accumulated feature importance of each amino acid shows the distinguishability and contribution of that amino acid among all amino acids in differentiating states.

**Markov state model**

Markov state model (MSM) is used to construct long-timescale dynamics behavior (Wang, Zhou, et al., 2019). MiniBatch k-means clustering was used to classify each simulation frame to microstates. Macrostates were clustered based on the Perron-cluster cluster analysis (PCCA) (Deuflhard & Weber, 2005). Macrostates are considered as equilibrium or steady states. Transition matrix and transition probability were calculated to quantitatively show the transition dynamics among macrostates. A specific time interval, referred to as lag time, needs to be determined to construct transition matrix. The value of the lag time, as well as the number of macrostates, is selected based on the result of estimated relaxation timescale (Bowman et al., 2009). MSMBuilder (Harrigan et al., 2017) version 3.8.0 was employed to build Markov state models in this study.

**Transition path theory**

Transition path theory (TPT) (Metzner et al., 2009; Noé et al., 2009) is used to calculate the probability of transitioning from one state to another within the framework of a MSM. In the current study, macrostates 2 and 8 were chosen as closed and open states, respectively. All other states are treated as intermediate states. Possible transition pathways from the closed to the open state were explored. The committor probability $q_i^+$ is defined as the probability from state $i$ to reach the target state rather than initial state. Based on definition, $q_i^- = 0$ for all microstates in initial state and $q_i^+ = 1$ for all microstates in target state. The committor probability for all other microstates are calculated by the following equation:

$$-q_i^- + \sum_{k \in \text{target state}} T_{ik} q_k^+ = \sum_{k \in \text{target state}} T_{ik}$$

(9)

where $T_{ik}$ is the transition probability from state $i$ to state $k$. Sequentially, the effective flux is calculated as:

$$f_{ij} = \pi_i q_i^- T_{ij} q_j^+$$

(10)

where $\pi_i$ is the equilibrium probability of state $i$ in the normalized transition matrix $T$, and $q_i^+$ is the backward-commitor probability calculated as $q_i^+ = 1 - q_i^-$. However, backward flux $f_{ij}$ should also be considered and subtracted when calculating net flux. Therefore, the net flux $f_{ij}^+ = \max(0, f_{ij}^- - f_{ij})$. Total flux can then be calculated as:

$$F = \sum_{i \in \text{initial state}} \sum_{j \notin \text{initial state}} \pi_i T_{ij} q_j^+$$

(11)

The flux from initial state to target state can be decomposed to individual pathways $p_i$. Dijkstra algorithm is implemented in MSMBuilder for pathway decompostition. A set of pathways $p_i$ can be generated along $F_i$ which provides a relative probability by:

$$p_i = \frac{f_i}{\sum_j f_j}$$

(12)

**Results**

**Simulation trajectory analysis shows dynamical activity**

Two 10 microseconds simulation trajectories of the trimeric SARS-CoV-2 S glycoprotein were treated as reference and backbone Ca of the trimer were chosen and extracted as representative features of structures.

To probe the dynamical stability of two structures, the time evolution of the RMSD were plotted in Figure 1(A). All RMSD values were calculated with reference to the first frame of each trajectory. The average RMSD values in two states are 5.9Å and 10.6Å, respectively. The plot suggested that the closed state is relatively stable while the partially open state is dynamically active and undergoes significant conformational changes after 1 microsecond. However, the simulation of open state after 6 microseconds suggested a convergence in the RMSD value and a relatively stable structure, which corresponds to the detached S1 subunit from S2 fusion machinery.

RMSF results were plotted in Figure 1(B). The asymmetry in protein motion was noticed by comparing the individual dynamics behavior among three chains. Corresponding to the RMSD results in chain A, the RMSF results showed a similar high-degree conformational change in the RBD domain. However, the detachment in chain A, chain B, and C showed different movements that both NTD and RBD in chain B are more dynamically active than those in chain C, while in closed state the chains B and C displayed similar dynamics.

**Markov state model and transition path theory elucidates the closed-open transition**

Simulation trajectories were projected onto a two-dimensional (2D) plot in RMSD of Ca atoms with reference to the closed and
open state structures, respectively (Figure 2(A)). To apply MSM analysis, MiniBatch k-means clustering was applied to divide the simulation sampling space into 300 microstates based on the reduced-dimension plot, shown in Figure S1. The estimated relaxation timescale was plotted in Figure 2(B). The trend of implied relaxation timescale showed that the estimated time-scale was converged after 15 ns, which was chosen as the lag time for MSM.

The number of macrostates were determined based on the band gap in estimated relaxation timescale plot. Total of 8 macrostates were chosen to divide simulations into kinetically separate macrospaces. PCCA was applied to map microstates onto macrostates based on the eigenfunction structure of transition probability matrix. The resulting macrostates with transition probability are shown in Figure 3(A). Closed state and open state were equally divided into 4 macrostates, as states 2, 3, 4, 7 belonging to the closed state and states 1, 5, 6, 8 belonging to the open state. The closed state is stable with 95.5% probability to stay within closed macrostates. Macrostate 2 was found important due to its high probability of 9.9% to transfer from itself to open macrostates. The average transition probability from closed macrostates to open macrostates is 4.5%.

Macrostate 2 was selected as the representative closed state based on the similarity with its corresponding crystal structure. However, it is not reasonable to apply this rule when choosing the representative macrostate of open states since the open states undergo a dramatic conformational change. Instead, it should be chosen based on the transition probability. There is a probability of 97.9% in macrostate 8 to stay within itself and therefore was selected. Transition path theory was applied to calculate possible transition
pathways connecting these two states. Total of 2,317 pathways were generated and divided as 51 distinct channels floating from state 2 to state 8. The probability of each channel was calculated based on net flux from initial state to the target state. Overall, the probability of top 5 channels was listed in Table 1, with the contribution from the top 10 channels accounting for 88.7% of total population. The most probable path was state 2 → state 3 → state 5 → state 6 → state 8, and the corresponding representative structures were plotted in Figure 3(B) to show a series of transition processes.

**Random forest identifies important residues and structures**

To better understand the shift between closed and open states in S1 subunit, the pairwised Cα distances of S1 were extracted as features representing the character of protein configurations. There are 540 residues on each chain, residue ID from 27 to 676, and total of $1,620 \times 1,619/2 = 1,311,390$ Cα distances were collected as features. Before further analysis, features were transformed into contact map with our proposed feature transformation technique described in the Methods section. Considering the non-bonded chemical interactions length, we pick 10.0 Å as threshold for feature transformation.

Supervised machine learning model was applied to extract the key residues that are vital during allosteric process and study the structural differences among macrostates. For each simulation trajectory, frames were saved for every 1.2 nanoseconds (ns), resulting in 8,334 frames. Therefore, 16,668 samples with 1,311,390 features were extracted from the trajectories. Each sample was labeled based on the above macrostate result. Full dataset was further split into training set (70%) and testing set (30%). After the preparation of data, random forest model was applied to distinguish the intrinsic conformational differences among macrostates. Training scores and testing scores were plotted in Figure 4(A). 7 was chosen for the depth with corresponding testing accuracy of 92.18%, which indicated that the random forest model was able to catch the conformational characteristics of each macrostate only using pair-wised Cα distances. To further investigate the relationship between chain A and two other chains, the original Cα distances related with chain A were excluded and applied to another random forest model. Training and testing results are shown in Figure 4(B). The top 500 features accounted for 74.8% percent of the overall feature importance, shown in Figure S2. The testing accuracy with reduced features reached 88.04% at depth 8.

The top five important Cα distances were listed in Table 2. In order to identify key residues along the transition from the closed to the partially open state, the feature importance of each Cα distance was added and accumulated to the two related residues. S1 subunit structure was plotted in Figure 5(A) as reference. Top 20 important residues on chain B and C, with corresponding structure and accumulated structure importance under each figure, were plotted in Figure 5(B, C). Full results of residue importance on chain B and C are shown in Figure S3.

**Discussion**

The significance of the partially open state of receptor-binding domain in SARS-CoV-2 for interacting with the host cell
receptor has been extensively studied (Walls et al., 2019; Yuan et al., 2017). Specifically, the opening of S1 subunit, thus exposing RBD, is necessary for engaging with ACE2 and following cleavage of \( S_0 \) site (Kirchdoerfer et al., 2018). While the RBD exhibits inherently flexibility enabling itself recognized by the receptor (Kirchdoerfer et al., 2016), the motion of this closed to open state shift still needs in-depth study.

It is reported that the SARS-CoV-2 S trimer shows a \( C_3 \) symmetry at closed state and asymmetry with chain A at open state (Wrapp et al., 2020). Through the RMSF result, we noticed the asymmetry in dynamics at both closed and open states. The S1 subunit in chain B and the S2 subunit in chain C are more dynamically active than their corresponding structures in the closed state. The S1 domain in chain B is also more flexible than that in chain C in the open state. Above results implies the asymmetrical biological functions among the three chains.

Two random forest models were applied with different input features. The first model reached high accuracy in predicting macrostates based on all pair-wised \( C_x \) distances on S1 subunit. The second model with reduced features that does not include the motion of chain A also had comparable prediction accuracy. This indicates that chain B and C contain information of the closed-open transition in chain A. Combined with the finding of asymmetric dynamics in RMSF result, we hypothesized that there is a correlation between the chain A and two other chains. The correlation among chains may come from the chain B and C’s contribution to the protein motion in chain A. This could also originate from the protein-protein interaction along the opening movement of chain A. Further investigation of this mutual influence is warranted for a detailed clarification. Moreover, in order to understand the important structures on the tertiary level, the importance of \( C_x \) distances was accumulated to residues on S1 domain structure and we numerically identified key structures as RBD in chain B, NTD in chain C, RBD in chain B and NTD in chain B in descending order.

The result of Markov state model showed a great difference in the probability of macrostates to transition within themselves with macrostate 2 (closed state) of 78.0% and macrostate 1 (open state) of 54.0%. This result implies that S1 subunit is more likely to stay in closed state, which agrees with the experimental finding that the closed state is more dynamically stable than the partially open state (Wrapp et al., 2020). Moreover, a possible dynamically stable state followed by the partially open state of the RBD was found and could be important in the closed-open transition. Specifically, macrostate 8 (open state) exhibited a high probability (97.9%) to stay within itself, where the RBD is detached from the S2 fusion machinery. Transition path theory further provided potential channels from macrostate 2 to 8 with the most probable channel (2-3-5-6-8) of 23.7% probability. This channel is considered important in representing the transient shifting and can be treated as the typical protein movement.

**Conclusion**

The spike protein is essential for SARS-CoV-2 as it destabilizes the trimer structure, causing the detachment of S1 subunit and exposing the RBD domain to host cell membrane. In this study, we used publicly available simulation trajectories of spike protein and studied the asymmetric dynamics nature of the trimer structure. Markov state model was applied to divide the conformational space into 8 macrostates. The

---

**Table 2. Top 5 \( C_x \) distances.**

<table>
<thead>
<tr>
<th>( C_x ) distances</th>
<th>Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chain C Phe 342, Chain C Asp 442</td>
<td>0.86%</td>
</tr>
<tr>
<td>Chain C Ala 419, Chain C Tyr 423</td>
<td>0.83%</td>
</tr>
<tr>
<td>Chain B Thr 323, Chain B Thr 333</td>
<td>0.76%</td>
</tr>
<tr>
<td>Chain C Cys 136, Chain C Gly 142</td>
<td>0.71%</td>
</tr>
<tr>
<td>Chain B Leu 390, Chain B Gly 545</td>
<td>0.60%</td>
</tr>
</tbody>
</table>

---

**Figure 4.** Random forest classification model using pair-wised \( C_x \) distances in S1 subunit. (A) Classification accuracy using different depths of trees. Depth 7 (shown in grey dashed line) was chosen with 92.18% accuracy. (B) Classification accuracy regarding different depths of trees using pair-wised \( C_x \) distances within chain B and C. Depth 8 (shown in grey dashed line) was chosen with 88.04% accuracy.
representative structures of each macrostate in the most probable channel are shown to present a clear route from the closed state to the partially open state. Transition matrix was calculated to determine the probability of the 8 macrostates with maximum of the summed probability of 9.9% from the macrostate 2 (closed state) to open macrostates. In order to represent the protein motions, the pairwise Cα distances from the amino acid residues located on the S1 subunit were extracted from each frame of simulations. Random forest models were applied to identify the key residues for the structural changes between macrostates based on these Cα distances. The little difference between prediction accuracy results from two random forest models, where one includes the movement of chain A and the other does not, implied a correlation between chain A and two other chains. Yet, whether this correlation originates from the mutual influence among chains or the intrinsic asymmetry in biological functions needs further investigation. Overall, our study quantitatively analyzed the S1 subunit with important Cα distances and residues, which contributes to the research on the states transitions in S protein.

Acknowledgements

Computational time was generously provided by Southern Methodist University’s Center for Research Computing. The authors thank D. E. Shaw for sharing the SARS-CoV-2 spike glycoprotein trajectories. The authors thank Ms. Xi Jiang from the Biostatistics Ph.D. program in the Statistics department of SMU for her help in manuscript.

Disclosure statement

No potential conflict of interest was reported by the author(s).

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.

Data availability statement

The data that support the findings of this study are available at https://www.deshawresearch.com/downloads/download_trajectory_sarscov2.cgi/.

References


---

**Figure 5.** S1 protein structure and accumulated feature importance. (A) Residue sequence with tertiary structure in S1 subunit, referenced to the released structure in the prefusion conformation (Wrapp et al., 2020). NTD (blue), N-terminal domain; RBD (green), receptor-binding domain; SD1 and SD2 (orange), subdomains. The sequence starts with residue Ace 26 to Thr 676. (B-C) The position of top 20 important residues are shown in red color. Accumulated tertiary structure importance in chain B and chain C are shown in numbers, respectively.


A.4. PAPER IV

PAPER

PASSer: prediction of allosteric sites server

Hao Tian 1, Xi Jiang 2 and Peng Tao 1

1 Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, TX, United States of America
2 Department of Statistical Science, Southern Methodist University, Dallas, TX, United States of America

E-mail: ptao@smu.edu

Keywords: protein allostery, XGBoost, graph convolutional neural network

Supplementary material for this article is available online

Abstract

Allostery is considered important in regulating protein's activity. Drug development depends on the understanding of allosteric mechanisms, especially the identification of allosteric sites, which is a prerequisite in drug discovery and design. Many computational methods have been developed for allosteric site prediction using pocket features and protein dynamics. Here, we present an ensemble learning method, consisting of eXtreme gradient boosting and graph convolutional neural network, to predict allosteric sites. Our model can learn physical properties and topology without any prior information, and shows good performance under multiple indicators. Prediction results showed that 84.9% of allosteric pockets in the test set appeared in the top 3 positions. The PASSer: Protein Allosteric Sites Server (https://passer.smu.edu), along with a command line interface (https://github.com/smutaogroup/passerCLI) provide insights for further analysis in drug discovery.

1. Introduction

Allostery is the process in which proteins transmit the perturbation caused by the effect of binding at one site to a distal functional site [1]. The allosteric process is fundamental in the regulation of activity. Compared with non-allosteric drugs, allosteric drugs have many advantages: they are conserved and highly specific [2]; they can either activate or inhibit proteins; they can be used in conjunction with orthosteric (non-allosteric) drugs. Although allosteric drugs are important in the pharmaceutical industry [3], they are still poorly understood [4]. Most allosteric mechanisms remain elusive because of the difficulty of identifying potential allosteric sites [5].

Many allosteric site prediction methods have been developed based on molecular dynamics (MD) simulations [6], normal mode analysis [7], two-state Gō models [8] and machine learning (ML) models [9–11]. Among the existing methods, AllositePro [12], AlloPred [13], SPACER [14] and PARS [15] are available as web servers or open-source packages. These previous studies have shown that it is promising to identify allosteric sites by combining static pocket features with protein dynamics. In these studies, static features are calculated by site descriptors describing physical properties of protein pockets, while the protein dynamics are extracted by MD simulation or perturbation.

ML methods have been shown to be superior in the classification of protein pockets. For example, Allosite [11] and AlloPred [13] used support vector machine (SVM) [16] with optimized features. Chen et al [17] used random forest (RF) [18] to construct a three-way predictive model. With the development of ML, more advanced models have been developed and can contribute to the allosteric site classification. eXtreme gradient boosting (XGBoost) [19] is one of the most powerful ML techniques in classification. It is an implementation of the gradient boosting algorithm with regularized terms to reduce overfitting. Compared with SVM and RF, XGBoost achieved superior predictive performance in the protein–protein interactions [20] and hot spots [21].

Though physical properties are largely contained in many methods, topological information is largely ignored and is considered important in classifying pockets. In order to explore the geometry features, an
atomic graph is constructed for each pocket. Atoms are treated as nodes and the pairwise bond distances are calculated as edges [9]. Graph convolutional neural networks (GCNNs) [22], a popular concept in deep learning, have been applied in biological-related predictions, ranging from chemical reactions [23], molecular properties [24], to drug-target interactions [25].

In this study, protein pockets are predicted using an ensemble learning method, which combines the results of XGBoost and GCNN. This model can learn both physical properties and topology information of allosteric pockets and has been proven to be superior to the single XGBoost and GCNN models. Various performance indicators validated the success of this ensemble learning method compared with previous methods.

2. Methods

2.1. Protein database

The data used in the current work was collected from the Allosteric Database (ASD) [26]. There are a total of 1946 entries information of allosteric sites with different proteins and modulators. To ensure data quality, 90 proteins were selected using previous rules [11]: protein structures with either resolution below 3Å or missing residues in the allosteric sites were removed; redundant proteins in the rest of the data that have more than 30% sequence identity were filtered out. The names and IDs of the 90 proteins extracted from the PDB [27, 28] are listed in table S1 (is available online at stacks.iop.org/MLST/2/035015/mmedia).

2.2. Site descriptors

FPocket algorithm [29] is used to detect pockets from the surface of the selected proteins. A pocket is labeled as either 1 (positive) if it contains at least one residue identified as binding to allosteric modulators or 0 (negative) if it does not contain such residues. Therefore, a single protein structure may have more than one positive label. A total of 2246 pockets were detected with 119 pockets being labeled as allosteric sites. There are 19 features calculated by the FPocket as shown in table S2.

2.3. Pearson correlation coefficient

Pearson correlation coefficient (PCC) measures the linear correlation of two variables. Given a pair of variables $X$ and $Y$ as $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, PCC ($r_{X,Y}$) is calculated as

$$
    r_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}
$$

where $n$ is the sample size and $\bar{x}$, $\bar{y}$ are sample means. PCC has a value between $-1$ and $1$. Positive value represents a positive correlation, and negative value represents a negative correlation. The absolute value indicates the degree of correlation. The larger the absolute value, the stronger the correlation.

2.4. eXtreme gradient boosting

XGBoost is an ensemble learning method that combines several decision trees in sequence.

Let $D = \{(x_i, y_i) | |D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}^n\}$ represents a dataset with $m$ features and $n$ labels. The $j$th decision tree in XGBoost predicts a sample $(x_i, y_i)$ by

$$
    g_j(x_i) = w_j(x_i)
$$

where $w_j$ is the leaf weights of this decision tree. The final prediction of XGBoost is given by the summation of predictions from each decision tree:

$$
    \hat{y}_i = \sum_{j=1}^{M} g_j(x_i)
$$

where $M$ is the total number of decision trees. To overcome overfitting introduced by decision trees, the objective function in XGBoost is composed of a loss function $l$ and a regularization term $\Omega$:

$$
    \text{obj}(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{M} \Omega(f_j)
$$

where $\Omega(f) = \gamma T + \frac{\lambda}{2} \sum_{j=1}^{T} \omega_j^2$ with $T$ represents the number of leaves and $\gamma$, $\lambda$ are regularization parameters.
During training, XGBoost iteratively adds new decision trees. The prediction of the $t$th iteration is expressed as

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + g_i(x_i). \quad (5)$$

Correspondingly, the objective function of the $t$th iteration is

$$\text{obj}^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + g_i(x_i)) + \Omega(f_t). \quad (6)$$

XGBoost introduces both first derivative and second derivative of the loss function. By applying Taylor expansion on the objective function at second order, the objective function of the $t$th iteration can be expressed as

$$\text{obj}^{(t)} \approx \sum_{i=1}^{N} [l(y_i, \hat{y}_i^{(t-1)}) + \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})] f_t + \frac{1}{2} \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)}) f_t^2 + \Omega(f_t). \quad (7)$$

XGBoost can predict the labels of sample data with corresponding probabilities. For one pocket, XGBoost outputs the probability of this pocket being an allosteric pocket. This pocket is labeled as positive (allosteric) if the predicted probability is over 50% or negative otherwise.

There are only 5.3% of positive labels in this binary classification job, which means that the input data is highly imbalanced. To focus more on the limited positive labels, XGBoost uses ‘scale_pos_weight’, a parameter for controlling the balance of positive and negative weights. A typical value of this weight equals to the number of negative samples versus the number of positive samples.

In the current study, the maximum tree depth for base learners and the weights for positive labels were fine-tuned while keeping other parameters as default values. The XGBoost algorithm is implemented using Scikit-learn package \cite{scikit-learn} version 0.23.2.

### 2.5. Graph convolutional neural network

The GCNN in this work follows this formula \cite{GCN}: \[ H^{(l+1)} = \text{ReLU}(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W(l)) \] (8)

where $H^{(l)}$ and $H^{(l+1)}$ represents the $l$th and $l+1$th layer, respectively. $H^{(l)} \in \mathbb{R}^{N \times D}$, where $N$ is the number of nodes and $D$ is the number of features. Rectified linear unit (ReLU($x$) = max(0, $x$)) is used as the activation function. $W(l)$ denotes the weight matrix in the $l$th layer. $D$ and $A$ represent degree matrix and adjacent matrix, respectively, with $\tilde{D}_v = \sum_i \tilde{A}_{vi}$. Renormalization (indicated by the ∼ symbol) is applied for the undirected graph $G$ where each node is added with a self-connection. Therefore, $\tilde{A} = A + I_N$ where $I_N$ is the identity matrix.

A graph readout is calculated through the average of node features for each graph:

$$h_g = \frac{1}{|V|} \sum_{v \in V} h_v \quad (9)$$

where $h_g$ is the readout result of graph $g$ and $h_v$ is the node feature in node $v$. $V$ represents the nodes in graph $g$.

An example of 1-layer GCNN model is shown in figure 1. A graph is first fed into a convolution layer. The in-degree and out-degree of a node refer to the number of edges coming into and going out from that node, respectively. In-degree of each node was calculated as the node feature. Graph feature is calculated as the average of node features in the readout layer with ReLU activation function. The output was further fed into a linear classification layer $g$, which predicts the probability of being an allosteric pocket. Previous research \cite{GCN} has shown the limitations of 1-layer GCNN. In the current study, atomic graphs of each protein pocket are constructed and fed into a 2-layer GCNN model. This model consists of two graph convolution layers, each with 256 dimensions of a hidden node feature vector, followed by a readout layer and a linear classification layer. The node degree is used as the initial node feature. The in-degree is the same as the out-degree in an undirected atomic graph. Graph representation is calculated as the average of node representations. The linear classification layer outputs the probabilities of pockets being allosteric sites.
Figure 1. Architecture of 1-layer GCNN model. 1-layer GCNN is composed of a graph convolution layer, a readout layer and a linear classification layer. Rectified linear unit is used as the activation function. An atomic graph is constructed for a given pocket and GCNN predicts the probability of this pocket being an allosteric site.

Table 1. Binary classification results. Confusion matrix can visualize and evaluate the performance of classification models. Rows represent the instances in predicted classes while columns represent the instances in real classes. True positive and true negative refer to the results where the model correctly predicts the positive and negative classes, respectively. Similarly, false positive and false negative refer to the results where the model incorrectly predicts the positive and negative classes, respectively.

<table>
<thead>
<tr>
<th>Real positive</th>
<th>Real negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted positive</td>
<td>True positive (TP)</td>
</tr>
<tr>
<td>Predicted negative</td>
<td>False positive (FP)</td>
</tr>
</tbody>
</table>

To overcome the potential limitation in training GCNN with an imbalanced dataset, the ratio between negative labels and positive labels was fine-tuned. Specifically, in each protein, allosteric pockets (positive samples) were fully used, while non-allosteric pockets (negative samples) were partially used. Each negative sample was randomly selected, and the total number of non-allosteric pockets equals the number of allosteric pockets times the ratio value.

In constructing atomic graphs, the threshold of bond distance was also fine-tuned. Each atom was considered as a node in the atomic graph and the pairwise distances between atoms were calculated. If the distance is below a specified threshold, an edge is constructed connecting the two related nodes. Thus, the distance threshold controls the degree of local connectivity.

The GCNN model is implemented using deep graph library package [32] version 0.4.3.

2.6. Performance indicators
For binary classification, the results can be classified as Table 1. Various indicators were used to quantify model performance: precision (TP / (TP + FP)) measures how well the model can predict real positive labels; accuracy ((TP + TN) / (TP + FP + FN + TN)) measures the overall classification accuracy; recall (or sensitivity, TP / (TP + FN)) and specificity (TN / (TN + FP)) together measure the ability to classify TP and TN; F1 score (2 * precision * recall / (precision + recall)) is the weighted average of precision and recall. The higher the values of these indicators, the better the model’s performance.

A receiver operating characteristic (ROC) curve was applied as another indicator to test model performance in binary classification. ROC curve is plotted as TP rate against FP rate with different threshold settings. The area under curve (AUC) is calculated for quantification. The upper limit for AUC value is 1.0. A dummy model should have an AUC value of 0.5.

The ranking information based on the predicted probabilities of protein pockets was also considered as an important indicator. Specifically, for each allosteric pocket, the ranking (predicted probability of being an allosteric site in descending order) was recorded and categorized as the first, second, third or other positions. The ranking result indicates how likely the allosteric pockets can appear in the top positions among all detected pockets in the same protein. A model with good performance should rank an allosteric pocket in the top positions.

3. Results

3.1. Feature exploration
The distribution of 19 features is shown in figure S1. While some exhibited long-tail distributions such as features 1 (Score) and 2 (Druggability score), data normalization is unnecessary since XGBoost does not
require normal distribution. Instead, XGBoost, like other tree-based models, only focuses on the order, and whether features are normalized or not does not affect the prediction results. Violin plots are shown in figure S2 to better distinguish the feature distribution of allosteric sites and non-allosteric sites.

The correlation matrix between these features is shown in figure S3. Several features exhibited high correlations. For example, features 3 (Number of alpha spheres), 4 (Total SASA), 5 (Polar SASA), 6 (Apolar SASA), and 7 (Volume) are highly correlated with each other. Features 17 (Alpha sphere density) and 18 (Center of mass) are also strongly correlated with these five features.

For each feature value in a pocket, the inner ranking refers to the ranking position of this feature among values of other pockets in the same protein. Similar to the definition of inner ranking, the overall ranking refers to the ranking position of this feature value among values of other pockets in the overall dataset. Both sets of ranking features were normalized. The correlations between the original features and these two ranking feature sets were calculated and shown in figure S3. The high negative values indicate strongly negative correlations between the original features and the ranking features. While feature rankings were calculated and applied as additional features in a previous study [13], in the current dataset, the high correlation indicated that the ranking features provided little additional information and thus were discarded.

3.2. Prediction performance of XGBoost
XGBoost model can overcome the limitation of data imbalance by controlling the weight difference between negative labels and positive labels. This parameter was fine-tuned along with the maximum depth of trees. The results are plotted in figure 2. Two sets of parameters reached high F1 scores, and both were selected in the final model. The final XGBoost model is composed of two models, each with one set of parameters. The results in any given pocket are the averaged results predicted by these two models. The results of the fine-tuned XGBoost model are listed in table 2. Compared with the reference results, XGBoost model exhibited higher accuracy, precision, specificity, and ROC AUC values with comparable results in recall and F1 scores. Therefore, XGBoost model performs well in allosteric site prediction.

3.3. Prediction performance of GCNN
Unlike XGBoost, GCNN models suffer from imbalanced dataset. To address this problem, the ratio between negative labels and positive labels was evaluated first. The results are plotted in figure 3(A). A ratio of 2...
Figure 3. Fine-tuned results of ratio and distance threshold parameters in GCNN model. (A) The ratio between number of negative labels and number of positive labels was fine-tuned. Ratio of 2 is considered reaching a balance between recall and precision. (B) The atomic distance threshold was fine-tuned from 7 to 11 Å. There is no significant increase in F1 score after 10 Å, which is selected as the distance cutoff. For each parameter value, GCNN was run 10 times independently.

Table 3. Probabilities of predicting allosteric sites in the top 3 positions. Ensemble learning method can rank an allosteric site in the top 3 positions with a probability of 84.9%, which is higher than previous results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top 1</th>
<th>Top 2</th>
<th>Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARS [15]</td>
<td>44%</td>
<td>62%</td>
<td>73%</td>
</tr>
<tr>
<td>AlloPred [13]</td>
<td>57.5%</td>
<td>70.0%</td>
<td>NA*</td>
</tr>
<tr>
<td>Ensemble learning</td>
<td>60.7%</td>
<td>81.6%</td>
<td>84.9%</td>
</tr>
</tbody>
</table>

*Not available in the reported results.

(number of negative labels: number of positive labels = 2:1) was selected. The distance threshold was further fine-tuned, and the results are plotted in figure 3(B). 10 Å was selected as the distance cutoff when constructing atomic graphs.

The results of the fine-tuned GCNN model with 10 independent runs are listed in table 2. Compared with XGBoost, GCNNs are less effective in classifying allosteric sites. However, it is expected that combining XGBoost and GCNN will result in better performance than either model.

3.4. Prediction performance of model ensembling
The ensemble learning model is composed of both XGBoost model and GCNN model. For a given pocket, physical properties are calculated and fed into the XGBoost model; a representative atomic graph is fed into the GCNN model. The final result is calculated as the averaged probability of these two models. This final model contains both the physical properties and topological features of protein pockets. The combined results are listed in table 2. Compared with the XGBoost model, model ensembling leads to a 6.00% increase in recall, a 0.82% decrease in precision, and a 2.89% increase in F1 score. The AUC ROC value also had a 1.89% increase.

For each protein, the identified pockets are ranked based on the predicted probabilities. Overall, 60.7% of allosteric pockets are predicted as the first position, while 81.6% among the top 2 and 84.9% among the top 3. In other words, if a pocket is an allosteric pocket, there is a probability of 84.9% that it can be predicted in the top 3 among all detected pockets in the same protein. The prediction results, together with values reported in other studies, are listed in table 3. It should be noted that the types and amounts of proteins in the test set are different from each other.

3.5. Novel allosteric sites prediction
To test this ensemble learning method, two proteins not in the dataset (table S1) were used. The predicted allosteric pockets of these two proteins are illustrated in figure 4. These two proteins represent two different types of allosteric proteins. The second PDZ domain (PDZ2) is a dynamics-driven protein in human PTP1E protein which undergoes allosteric process upon binding with peptides [33]. The light-oxygen-voltage domain of *Phaeodactylum tricornutum* Aureochrome 1a (AuLOV) is a conformational-driven allosteric protein [34]. AuLOV is a monomer in the dark state and undergoes dimerization upon blue light perturbation [35]. In both cases, our prediction model ranks the allosteric sites as the top 1 with probabilities of 45.14% and 89.46%, respectively. This indicates that this model is capable of predicting both dynamics-driven and conformational-driven allosteric proteins. Apparently, the probability for the dynamics-driven allosteric protein is much smaller than the one of the conformational-driven allosteric protein, which is not unexpected.
Figure 4. Prediction results of two examples not included in the training set: (A) Dynamics-driven PDZ2 protein in bound states (PDB ID 3LNY); (B) the LOV domain of conformational-driven Phaeodactylum tricornutum Aureochrome 1a in dark state (PDB ID: 5DKK). Red regions are the most probable pockets in the predicted results with probabilities of (A) 45.14% and (B) 89.46% and are also the true allosteric sites.

Figure 5. Web server workflow. User can upload either a PDB ID or a PDB file in the web server. FPocket is used to detect pockets. For each pocket, physical properties are calculated and predicted using a pretrained XGBoost model; while an atomic graph is constructed and fed into a pretrained GCNN model. The final probability is given by averaging results from both models.

3.6. Web and CLI usage
A web server based on the allosteric prediction method developed in this study is implemented using a Python web framework, Django. JSmol [36] is a JavaScript implementation of the Jmol package and is embedded in the web page for protein and pocket visualization. Web pages are rendered using Bootstrap. This server is named as Protein Allosteric Sites Server (PASSer). A workflow of PASSer is outlined in figure 5.

An example of input and output of PASSer is displayed in figure 6. Users can submit a PDB ID if available or upload a custom PDB file as shown in figure 6(A). By default, all chains in the protein are analyzed. Prediction results are displayed as two parts: top 3 pockets with the highest probability rendered with the protein structure (figure 6(C)) and their probabilities (figure 6(B)). For each pocket, the corresponding residues can be retrieved by clicking the ‘Show Residues’ texts. Protein structure is visualized using JSmol. Each pocket is either displayed upon clicking its ‘Load pocket’ icon or hidden by clicking its ‘Hide pocket’ or overall ‘Reset’ icons.
Figure 6. PASSer web server pages. (A) Users can either submit a PDB ID or a PDB file in the home page. (B) Predicted top 3 pockets are summarized in a table with corresponding probabilities and pocket residues. (C) Protein structures and pocket sites are displayed in an interactive window.

A command line interface (CLI) is provided to facilitate potential developments. Similar with the web usage, this CLI can take either a PDB ID or a local PDB file for predictions.

4. Discussion

The quality of the dataset used for training is critical. Classification models often fail in prediction performance, and lack the generalization with poorly-collected datasets, such as insufficient training data or high similarity between structures. ASD is an online database that provides allosteric proteins and sites with high resolution, bringing opportunities for allosteric site prediction. There are other databases, such as ASBench [37] and sc-PDB [38], which can also be used to improve data quality and model performance.

In order to predict allosteric sites, proper pockets need to be identified on the surface of proteins. Several open-source pocket detection software has been developed. Previous results [29] have shown that, the geometry-based algorithm FPocket is superior to other methods, such as PASS [39] and LIGSITEMP [40], and can cover known allosteric sites. In addition, FPocket is under active development and can be integrated with other methods to build a complete pipeline for site prediction.

Several computational methods have been developed for allosteric site prediction over the past few years. Due to the fast development of ML methods, many models integrate ML methods, such as SVM and RF, for accurate predictions [11, 17]. One critical issue is that, many ML models fail when dealing with imbalanced datasets [41]: in the allosteric site database, negative samples account for a majority of the dataset with a limited proportion of positive samples. Undersampling is one way to rebalance the dataset. For example, Allosite discarded some negative labels and used a ratio of 1:4 between positive and negative labels. However, undersampling leads to insufficient usage of the overall dataset. XGBoost, as a gradient boosting method, overcomes this data imbalance by controlling the relative weights between classes so that the dataset can be fully used. Various performance indicators, as listed in table 2, validated the effectiveness of XGBoost for the identification of allosteric sites.

It is worth noting that some features are highly correlated, as shown in figure S3. This collinearity should be addressed in regression models, which reduces the model precision and thus weakens prediction results. In contrast, XGBoost is free from this problem. When several features are found to be highly correlated to each other, XGBoost will choose one of these features. Therefore, collinearity does not affect the prediction results. Nevertheless, collinearity influences model interpretation, such as feature importance, which should be considered with caution.

Physical properties have been widely used to describe the characteristics of pockets. These features are normally calculated using static protein structures. To probe the dynamical behavior of pockets, normal mode analysis and MD simulations are normally conducted [7, 42]. Results from these methods have shown that models can achieve satisfactory performance through the combination of both static features and protein dynamics. However, pocket geometry is often ignored, which could play an important role in prediction. Therefore, a GCNN is applied to retain the topological information. Specifically, pockets are
represented as undirected graphs at atomic level, and GCNN is designed to learn the local connectivity among atoms. While a previous study [9] included energy-weighted covalent and weak bonds in the prediction of allosteric sites, it should be noted here that: (1) it is assumed that the physical properties are explicitly retained in the site descriptors and GCNN only studies the node degree; (2) GCNN does not require any a priori information about the location of active sites. The ensemble learning method, consisting of GCNN and XGBoost, exhibited higher performance compared with single models.

5. Conclusion

The proposed ensemble learning method involves XGBoost and GCNN, which can learn both the physical properties and topology of protein pockets. The results are comparable with previous studies and have a higher percentage of ranking allosteric sites at top positions. The web server provides a user-friendly interface. Protein structures and top pockets are visualized in an interactive window on the result page. This ensemble learning method, embedded in the PASSer and CLI, can help exploration on protein allostery and drug development.

Code availability

The PASSer server is available at https://passer.smu.edu. The command line interface is available on GitHub at https://github.com/smutaogroup/passerCLI.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

Computational time was generously provided by Southern Methodist University’s Center for Research Computing. 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. The authors would like to thank Kurt Pifer for his help in website development.

Conflict of interests

The authors declare no competing interests.

ORCID iDs

Hao Tian https://orcid.org/0000-0002-0186-9811
Peng Tao https://orcid.org/0000-0002-2488-0239

References

[7] Panjkovich A and Daura X 2012 Exploiting protein flexibility to predict the location of allosteric sites BMC Bioinform. 13 273
[19] Chen T and Guerstrin C 2016 Xgboost: a scalable tree boosting system Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining
[21] Li K, Zhang S, Yan D, Bin Y and Xia J 2020 Prediction of hot spots in protein–DNA binding interfaces based on supervised isometric feature mapping and extreme gradient boosting BMC. Bioinform. 21 1–10
[28] Burley S K et al 2019 RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy Nucleic Acids Res. 47 D464–74
[29] Le Guilloux V, Schmidtke P and Tuffery P 2009 Fpocket: an open source platform for ligand pocket detection BMC. Bioinform. 10 1–11
[34] Heintz U and Schlüchting I 2016 Blue light-induced LOV domain dimerization enhances the affinity of Aureochrome 1a for its target DNA sequence Elife 5 e11860
[40] Huang B and Schroeder M 2006 LIGSITE csc: predicting ligand binding sites using the Connolly surface and degree of conservation BMC. Struct. Biol. 6 19
[41] Zhao X-M, Li X, Chen L and Aihara K 2008 Protein classification with imbalanced data Proteins: Struct. Function Bioinform. 70 1125–32
A.5. PAPER V

Explore Protein Conformational Space With Variational Autoencoder

Hao Tian¹, Xi Jiang², Francesco Trozzi¹, Sian Xiao¹, Eric C. Larson³ and Peng Tao¹*

¹Center for Research Computing, Center for Drug Discovery, Design, and Delivery (C4D), Department of Chemistry, Southern Methodist University, Dallas, TX, United States, ²Department of Statistical Science, Southern Methodist University, Dallas, TX, United States, ³Department of Computer Science, Southern Methodist University, Dallas, TX, United States

Molecular dynamics (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.

Keywords: protein system, conformational space, variational autoencoder, molecular dynamics, deep learning

1 INTRODUCTION

Molecular dynamics (MD) simulations have been applied extensively to understand protein structure, function, and kinetics. Klepeis et al. (2009); Song et al. (2020) Through the development of hardware and software, e.g., graphics processing unit (GPU) Shaw et al. (2009) and OpenMM Eastman et al. (2017), 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. Hartmann et al. (2014) Moreover, the energy landscapes of proteins are discretized with many local energy minima separated by high energy barriers. Krivov (2011) 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) Hamelberg et al. (2004), to expand the landscape. However, some domain knowledge is required to define the essential coordinates, e.g., collective variables (CVs). Maximova et al. (2016) 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 Bowman et al. (2010) 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 (Degiacomi, 2019; Lemke and Peter, 2019; Tsuchiya et al., 2019; Jin et al., 2021; Ramaswamy et al., 2021). 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 being treated as CVs. (Sultan et al., 2018).

In this study, we proved the success of variational autoencoders in protein sampling by using the enzyme adenosine kinase (ADK) as an example. The crystalized ADK is initially in its closed state and undergoes a series of conformational changes to its open state (Schrank et al., 2013). 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) (Berman et al., 2000) 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 (Jorgensen et al., 1983). Na$^+$ and Cl$^-$ 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 (Essmann et al., 1995). Bonds associated with hydrogen atoms were constrained using the SHAKE algorithm (Ryckaert et al., 1977) with 2 fs step size. All simulations were conducted with CHARMM27 force field (Foloppe and MacKerell, 2000) and OpenMM 7 Eastman et al. (2017).

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

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

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 (Baldi, 2012). 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 (Wetzel, 2017; Strub and Mary, 2015). 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) (Doersch, 2016). 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 $p(x, z) = p(x|z)p(z)$. 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 $p_\theta(z|x) = p(x|z)p(z)/\int p_\theta(z|x)p(z)dz$ is approximated using the tractable variational Bayes approach which maximizes the Evidence Lower Bound (ELBO):

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

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 (Abadi et al., 2016) backend v2.4.1.

2.3 Performance Assessment Criteria

Several previous studies (Alam et al., 2020; Alam and Shehu, 2020; Guo et al., 2020) focused on the evaluation of autoencoders on the generation of nonlinear featurization and the learned nonlinear representations of protein tertiary structures. In the current study, a similar strategy was employed to quantify and compare the performance of autoencoders and variational autoencoders. Specifically, four metrics were chosen as:

1) Spearman correlation coefficient, 2) Pearson correlation coefficient (PCC), 3) root-mean-square deviation (RMSD), and 4) discrete optimized protein energy (DOPE):

1) Spearman correlation coefficient. Spearman correlation coefficient is used to quantitatively analyze how well distances between all pairs of points in the original spaces
have been preserved in the reduced dimensions. Spearman correlation coefficient is calculated as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$  \hspace{1cm} (3)

where $d_i$ and $n$ are the difference in paired ranks and number of samples, respectively.

2) Pearson correlation coefficient (PCC). PCC uses L2 distance (also referred as Euclidean distance) to estimate the linear relation between distances in the original space and the reduced space. PCC is calculated as:

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$  \hspace{1cm} (4)

where $n$ is the sample size, $\bar{x}$, $\bar{y}$ are the mean value of distances, respectively. In the current study, $x$ and $y$ are used to represent distances input feature space and latent space, respectively.

3) Root-mean-square deviation (RMSD). RMSD is used to quantify the conformational differences between training and decoded protein conformations. Given a molecular structure $r$ with a reference $r^0$, RMSD is calculated as:

$$\text{RMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (r_i - U r_i)^2}$$  \hspace{1cm} (5)

where $r$ and $r^0$ are coordinates and normally represented in the Cartesian space. $N$ is the number of atoms. $U$ is the transformation matrix for the best-fit alignment between a given structure and its reference structure.

4) Discrete optimized protein energy (DOPE). The DOPE score (Shen and Sali, 2006) has been extensively used in the assessment of both experimentally and computationally generated models (Deka et al., 2015; Khare et al., 2019). The lower the DOPE score, the better the model. DOPE scores were calculated using modeling package MODELLER (Eswar et al., 2006) version 10.1.

The correlation-based metrics have been widely applied in the comparison between dimensionality reduction methods for biomolecules (Tian and Tao, 2020; Trozzi et al., 2021). 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.

Moreover, to evaluate the quality of deep learning models, two distance-based metrics, maximum mean discrepancy and earth mover’s distance, were applied to compare the training and generated distributions. Following the strategy from a previous study (Alam and Shehu, 2020), RMSDs were calculated as a proxy variable representing the protein tertiary structures.

1) Maximum mean discrepancy (MMD). MMD is a statistical analysis to represent distances between projected distributions using mean embeddings of features. MMD is defined by a feature map $\varphi: \mathcal{X} \rightarrow \mathcal{H}$ where $\mathcal{H}$ is a reproducing kernel Hilbert space. MMD is calculated as:

$$\text{MMD}(P, Q) = \| \mathbb{E}_{X \sim P} [\varphi(X)] - \mathbb{E}_{Y \sim Q} [\varphi(Y)] \|_H$$  \hspace{1cm} (6)

In the current study, MMD is used for the purpose of model comparison and selection. A good model is expected to generate distributions similar to the training sets, leading to small MMD values.

2) Earth mover’s distance (EMD). EMD is a measurement to evaluate dissimilarity between two multi-dimensional probability distributions. It is also known as the Wasserstein metric in mathematics. Analogically, two
distributions on a two-dimensional surface could be considered as two piles of a certain amount of earth (dirt). EMD is the least amount of work needed of transforming one pile into the other. EMD is calculated using SciPy v1.5.2 (Virtanen et al., 2020).

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 (Scherer et al., 2015) 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 (Kubitzki and de Groot, 2008), 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 Figures 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 (Arora and Brooks, 2007; Hanson et al., 2007; Formoso et al., 2015). 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 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 10 models (5 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.

![Figure 2](https://example.com/figure2.png)

**FIGURE 2** | Protein structures of ADK in (A) open state and (B) closed state. The LID, NMP, and CORE regions are colored as orange, blue, and grey, respectively. Four vectors, V1-V2 for CORE-NMP angle and V3-V4 for CORE-LID angle, are used to characterize protein conformations. The related residues are illustrated in Table 1.
and the mean value of each metric was calculated. The results of performance assessment metrics are plotted in Figure 4.

The variational autoencoder with 4 hidden layers performed the best with high Spearman and Pearson coefficients and low RMSD.

In terms of the encoder (Figures 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), variational autoencoders lead to a lower deviation between the training and decoded protein structures. Based on the elbow criteria, 4-layer VAE was selected as the final model with good performance and short training time. The convergence of the training process was evaluated using Spearman correlation coefficient, Pearson correlation coefficient, and RMSD as metrics with regard to the number of training iterations (epochs) used in the training process of 4-layer VAE. The convergence of these values is apparent when approaching 200 epochs (Figure 4D).

**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>1AKE-4AKE</th>
<th>1DVR</th>
<th>2AK3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tail</td>
<td>Head</td>
<td>Tail</td>
</tr>
</tbody>
</table>

**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>2</td>
<td>128</td>
<td>4980</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></td>
<td>16, 128, 1024</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1024, 256, 64, 16</td>
<td></td>
<td>16, 64, 256, 1024</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2048, 512, 128, 32, 8</td>
<td></td>
<td>8, 32, 128, 512, 2048</td>
<td></td>
</tr>
</tbody>
</table>
To further evaluate the model performance in decoding protein conformations, two distance-based metrics were calculated and listed in Table 3. For both cases, the variational autoencoder with 4 hidden layers reached the lowest values. This indicates that 4-layer VAE is capable of generating protein conformations that are closer to the training distribution. In addition to the comparison of distributions, the quality of decoded protein conformations is quantified by DOPE score. The DOPE scores of decoded protein conformations were calculated with their distributions represented as boxplots in Figure 5. The mean DOPE score decreases from 1-layer VAE to 4-layer VAE, indicating the increased ability to generate protein conformations that have lower potential energy. This ability does not increase further with more complex model structure as shown by the 5-layer VAE.

To summarize, the above results suggest that a variational autoencoder with 4 hidden layers in both of the encoder and decoder modules exhibited the best performance in terms of learning a meaningful latent space and decoding physically plausible proteins conformations. Therefore, 4-layer VAE was chosen to conduct further analysis.

Two decoded ADK structures in the open and the closed states through the selected 4-layer VAE are illustrated in Figure 6. The mean RMSD between the training and decoded structures is 1.03 Å. The learned latent space is plotted in Figure 7A. 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 Figures 7C–F with the NMP-CORE and LID-CORE angles plotted in Figure 7B.

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 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 8 to show the conformational spaces from the 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 simulation. 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 9. It is shown that the short combined simulations in Figure 8B exhibited similar implied timescales as the reference trajectory.

4 DISCUSSION

The protein energy landscapes could be divided into many local energy minima which are represented as the metastable conformational states. These conformations are separated by free energy barriers that are much higher than $k_B T$. (He et al., 2003) Due to this reason, MD simulations are often trapped in a local minimum for a long simulation time before jumping to another. In this study, we aim to accelerate this inefficient process by directly taking protein structures from the less sampled regions as the initial structures for additional MD simulations. However, protein structures are high-dimensional data with the degrees of freedom as 3N in the Cartesian space. Unlike ADK
protein which has known intrinsic collective variables (NMP-LID angles) to characterize protein conformational space, most proteins do not have such representations.

To overcome this problem, 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 (Das et al., 2006; Tribello et al., 2012), this leads to an improved ability to learn the movement of covalently bonded atoms (Degiacomi, 2019).

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 several 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 dimensionality reduction technique and these two indicators have been widely used in such tasks. The performance of decoder modules was defined as the resemblance between the training and decoded protein conformations. RMSD and the DOPE score were used to quantify structure and system energy differences, respectively. The DOPE score has been used in the assessment of computationally generated models (Deka et al., 2015; Khare et al., 2019). A good model is expected to have a low DOPE score. We systematically compared the DOPE scores of generated protein conformations in different VAE settings. It is shown that a complicated model architecture with more hidden layers can generate protein conformations with lower DOPE scores, while this is converged after 4 hidden layers. We further compared the RMSD distributions between training and generated protein conformations with MMD and EMD indicators. Under both cases, 4-layer VAE achieved the lowest scores and was considered the best in representing protein conformational spaces.

Sampling efficiency is compared between the 20 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 8 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.

Through the angle plot (Figure 8) and the estimated timescale calculation (Figure 9), it is demonstrated that short MD simulations including trajectories starting from the generated conformations in the latent space could achieve the sampling efficiency comparable to a single long MD simulation. This suggests that iteratively conducting short MD simulations starting from conformations generated in the learned conformational space could serve as an alternative approach to extensive MD simulations.

![Figure 9](https://example.com/figure9.png)

**FIGURE 9** Estimated timescales with different lag times. The subplots (A) and (B) correspond to the trajectories in Figures 8A,B, respectively. Top 8 timescales were selected and each was plotted with 95% confidence interval.
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.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

REFERENCES


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. EL and PT also helped write the article. PT supervised the project. 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.


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.


**Conflict of Interest:** 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.

**Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Tian, Jiang, Trozzi, Xiao, Larson and Tao. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.
A.6. PAPER VI

ADMETboost: a web server for accurate ADMET prediction

Hao Tian¹ · Rajas Ketkar² · Peng Tao¹

Received: 13 September 2022 / Accepted: 31 October 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
The absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties are important in drug discovery as they define efficacy and safety. In this work, we applied an ensemble of features, including fingerprints and descriptors, and a tree-based machine learning model, extreme gradient boosting, for accurate ADMET prediction. Our model performs well in the Therapeutics Data Commons ADMET benchmark group. For 22 tasks, our model is ranked first in 18 tasks and top 3 in 21 tasks. The trained machine learning models are integrated in ADMETboost, a web server that is publicly available at https://ai-druglab.smu.edu/admet.

Keywords
ADMET · Machine learning · XGBoost · Web server

Introduction
Properties such as absorption, distribution, metabolism, excretion, and toxicity (ADMET) are important in small molecule drug discovery and therapeutics. It was reported that many clinical trials fail due to the deficiencies in ADMET properties [13, 15, 17, 28]. While profiling ADMET in the early stage of drug discovery is desirable, experimental evaluation of ADMET properties is costly with limited available data. Moreover, computational studies of ADMET in the clinical trial stage can serve as an efficient design strategy that can allow researchers to pay more attention to the most promising compounds [8].

Recent developments in machine learning (ML) promote research in chemistry and biology [23, 26, 32] and bring new opportunities for ADMET prediction. ADMEITlab [6] provides 31 ADMET endpoints with six machine learning models and further advanced to 53 endpoints using a multi-task graph attention network [29]. vNN [22] is a web server that applies the variable nearest neighborhood method to predict 15 ADMET properties. admetSAR [4] and admetSAR 2.0 [31] are also ML-based web servers for drug discovery or environmental risk assessment with random forest, support vector machine, and k-nearest neighbor models. As a fingerprint-based random forest model, FP-ADMET [27] evaluates over 50 ADMET and ADMET-related tasks. In these ML models, small molecules are provided in SMILES representations and further featureized using fingerprints, such as extended connectivity fingerprints [21] and Molecular ACCess System (MACCS) fingerprints [7]. Beside these, there are many other fingerprints and descriptors that can be used for ADMET prediction, such as PubChem fingerprints and Mordred descriptors. Taking advantage of all possible features enables a sufficient learning process for machine learning models.

One common issue is that many machine learning models in previous work are trained on different datasets, which leads to an unfair comparison and evaluation of ML models. As a curated dataset, Therapeutics Data Commons (TDC) [11] unifies resources in therapeutics for systematic access and evaluation. There are 22 tasks in the TDC ADMET benchmark group, each with small molecule SMILES representations and corresponding ADMET property values or labels.

Extreme gradient boosting (XGBoost) [3] is a powerful machine learning model and has been shown to be effective in regression and classification tasks in biology and chemistry [2, 5, 24, 25]. In this work, we applied XGBoost to learn a feature ensemble, including multiple fingerprints and descriptors, for accurate ADMET prediction. Our model performs well in the TDC ADMET benchmark group with 11 tasks ranked first and 19 tasks ranked top 3.
Methods

Therapeutics Data Commons

Therapeutics Data Commons (v0.3.6) is a Python library with an open-science initiative. It holds many therapeutics tasks and datasets including target discovery, activity modeling, efficacy, safety, and manufacturing. TDC provides a unified and meaningful benchmark for fair comparisons between different machine learning models. For each ADMET prediction task, TDC splits the dataset into the predefined 80% training set and 20% test set with scaffold split, which simulates the real-world application scenario. In practice, a well-trained machine learning model would be used to predict ADMET properties on unseen and structurally different drugs.

Fingerprints and descriptors

Six featurizers from DeepChem [20] were used to compute fingerprints and descriptors:

- MACCS fingerprints are common structural keys that compute a binary string based on a molecule’s structural features.
- Extended-connectivity circular fingerprints compute a bit vector by breaking up a molecule into circular neighborhoods. They are widely used for structure-activity modeling.
- PubChem fingerprints consist of 881 structural keys that cover a wide range of substructures and features. They are used by PubChem for similarity searching.
- Mordred descriptors [18] calculate a set of chemical descriptors such as the count of aromatic atoms or the count of all halogen atoms.
- RDKit descriptors calculate a set of chemical descriptors such as molecular weight and the number of radical electrons.

Extreme gradient boosting

Extreme gradient boosting is a powerful machine learning model. It boosts model performance through an ensemble that includes decision tree models trained in sequence.

Let \( D = \{(x_i, y_i) | D = n, x_i \in R^m, y_i \in R^a\} \) represent a training set with \( m \) features and \( n \) labels. The \( j \)th decision tree in an XGBoost model makes a prediction for sample \( (x_i, y_i) \) by \( g_j(x_i) = w_q(x_i) \), where \( w_q \) is the leaf weights. The final prediction of the XGBoost model is the sum of all \( M \) decision tree predictions with \( \hat{y}_i = \sum_{j=1}^{M} g_j(x_i) \).

The objective function consists of a loss function \( l \) and a regularization term \( \Omega \) to reduce overfitting:

\[
\text{obj}(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{M} \Omega(f_j)
\]

where \( \Omega(f_j) = \gamma T + \frac{1}{2} \sum_{l=1}^{T} \omega_l^2 \). \( T \) represents the number of leaves while \( \gamma \) and \( \lambda \) are parameters for regularization.

During training, XGBoost iteratively trains a new decision tree based on the output of the previous tree. The prediction of the \( r \)th iteration \( \hat{y}_i(t) = \hat{y}_i(t-1) + g_t(x_i) \). The objective function of the \( r \)th iteration is:

\[
\text{obj}(t) = \sum_{i=1}^{N} l(y_i, \hat{y}_i(t-1) + g_t(x_i)) + \Omega(f_t)
\]

XGBoost introduces the first and second derivatives of this objective function, which can be expressed as follows by applying Taylor expansion at second order:

\[
\text{obj}(t) \approx \sum_{i=1}^{N} l(y_i, \hat{y}_i(t-1) + \partial_g \hat{y}_i(t-1) f_t(x_i) + \frac{1}{2} \partial^2_g \hat{y}_i(t-1) f^2_t(x_i)) + \Omega(f_t)
\]

A total of seven parameters were fine-tuned with selected value options and are listed in Table 1. Default values were used for other parameters.

Performance criteria

For regression tasks, mean absolute error (MAE) and Spearman’s correlation coefficient are considered to evaluate model performance:

- MAE is used to measure the deviation between predictions \( y_i \) and real values \( x_i \) with \( n \) sample size.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Fine-tuned XGBoost parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and description</td>
<td>Values</td>
</tr>
<tr>
<td>n_estimators: number of gradient boosted trees</td>
<td>[50, 100, 200, 500, 1000]</td>
</tr>
<tr>
<td>max_depth: maximum tree depth</td>
<td>[3, 4, 5, 6, 7]</td>
</tr>
<tr>
<td>learning_rate: boosting learning rate</td>
<td>[0.01, 0.05, 0.1, 0.2, 0.3]</td>
</tr>
<tr>
<td>subsample: subsample ratio of instances</td>
<td>[0.5, 0.6, 0.7, 0.8, 0.9, 1.0]</td>
</tr>
<tr>
<td>colsample_bytree: subsample ratio of columns</td>
<td>[0.5, 0.6, 0.7, 0.8, 0.9, 1.0]</td>
</tr>
<tr>
<td>reg_alpha: L1 regularization weights</td>
<td>[0, 0.1, 1, 5, 10]</td>
</tr>
<tr>
<td>reg_lambda: L2 regularization weights</td>
<td>[0, 0.1, 1, 5, 10]</td>
</tr>
</tbody>
</table>
MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \quad (4)

- Spearman’s correlation coefficient $\rho$ measures the correlation strength between two ranked variables. Where $d_i$ represents the difference in paired ranks,

$$
\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (5)
$$

For binary classification tasks, area under curve (AUC) is calculated with receiver operating characteristic (ROC) and precision-recall curve (PRC). For both metrics, a higher value indicates a more powerful model.

- AUROC is the area under the curve where the x-axis is the false positive rate and the y-axis is the true positive rate.
- AUPRC is the area under the curve where the x-axis is the recall and the y-axis is the precision.

### Results and Discussion

#### Model Performance

We first used a random seed to split the overall dataset into a training set (80%) and a test set (20%). The XGBoost model was trained with the training set using 5-fold cross-validation (CV). A randomized grid search CV was applied to optimize hyperparameters. The parameter set with the highest CV score was used, and the model performance was evaluated on the test set. We repeated this process five times with varying random seeds from zero to four following the TDC guidelines. The evaluation results are

<table>
<thead>
<tr>
<th>Task</th>
<th>Metric</th>
<th>Method</th>
<th>Method Score</th>
<th>XGBoost Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Absorption</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caco2</td>
<td>MAE</td>
<td>RDKit2D + MLP</td>
<td>0.393 ± 0.024</td>
<td>0.288 ± 0.011</td>
<td>1st</td>
</tr>
<tr>
<td>HIA</td>
<td>AUROC</td>
<td>AttrMasking</td>
<td>0.978 ± 0.006</td>
<td>0.987 ± 0.002</td>
<td>1st</td>
</tr>
<tr>
<td>Pgp</td>
<td>AUROC</td>
<td>AttrMasking</td>
<td>0.929 ± 0.006</td>
<td>0.911 ± 0.002</td>
<td>4th</td>
</tr>
<tr>
<td>Bioav</td>
<td>AUROC</td>
<td>RDKit2D + MLP</td>
<td>0.672 ± 0.021</td>
<td>0.700 ± 0.010</td>
<td>1st</td>
</tr>
<tr>
<td>Lipo</td>
<td>MAE</td>
<td>ContextPred</td>
<td>0.535 ± 0.012</td>
<td>0.533 ± 0.005</td>
<td>1st</td>
</tr>
<tr>
<td>AqSol</td>
<td>MAE</td>
<td>AttentiveFP</td>
<td>0.776 ± 0.008</td>
<td>0.727 ± 0.004</td>
<td>1st</td>
</tr>
<tr>
<td><strong>Distribution</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBB</td>
<td>AUROC</td>
<td>ContextPred</td>
<td>0.897 ± 0.004</td>
<td>0.905 ± 0.001</td>
<td>1st</td>
</tr>
<tr>
<td>PPBR</td>
<td>MAE</td>
<td>NeuralFP</td>
<td>9.292 ± 0.384</td>
<td>8.251 ± 0.115</td>
<td>1st</td>
</tr>
<tr>
<td>VDss</td>
<td>Spearman</td>
<td>RDKit2D + MLP</td>
<td>0.561 ± 0.025</td>
<td>0.612 ± 0.018</td>
<td>1st</td>
</tr>
<tr>
<td><strong>Metabolism</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYP2C9 inhibition</td>
<td>AUPRC</td>
<td>AttentiveFP</td>
<td>0.749 ± 0.004</td>
<td>0.794 ± 0.004</td>
<td>1st</td>
</tr>
<tr>
<td>CYP2D6 inhibition</td>
<td>AUPRC</td>
<td>AttentiveFP</td>
<td>0.646 ± 0.014</td>
<td>0.721 ± 0.003</td>
<td>1st</td>
</tr>
<tr>
<td>CYP3A4 inhibition</td>
<td>AUPRC</td>
<td>AttentiveFP</td>
<td>0.851 ± 0.006</td>
<td>0.877 ± 0.002</td>
<td>1st</td>
</tr>
<tr>
<td>CYP2C9 substrate</td>
<td>AUPRC</td>
<td>Morgan + MLP</td>
<td>0.380 ± 0.015</td>
<td>0.387 ± 0.018</td>
<td>1st</td>
</tr>
<tr>
<td>CYP2D6 substrate</td>
<td>AUPRC</td>
<td>RDKit2D + MLP</td>
<td>0.677 ± 0.047</td>
<td>0.648 ± 0.023</td>
<td>3rd</td>
</tr>
<tr>
<td>CYP3A4 substrate</td>
<td>AUPRC</td>
<td>CNN</td>
<td>0.662 ± 0.031</td>
<td>0.680 ± 0.005</td>
<td>1st</td>
</tr>
<tr>
<td><strong>Excretion</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half life</td>
<td>Spearman</td>
<td>Morgan + MLP</td>
<td>0.329 ± 0.083</td>
<td>0.396 ± 0.027</td>
<td>1st</td>
</tr>
<tr>
<td>CL-Hepa</td>
<td>Spearman</td>
<td>ContextPred</td>
<td>0.439 ± 0.026</td>
<td>0.420 ± 0.011</td>
<td>2nd</td>
</tr>
<tr>
<td>CL-Micro</td>
<td>Spearman</td>
<td>RDKit2D + MLP</td>
<td>0.586 ± 0.014</td>
<td>0.587 ± 0.006</td>
<td>1st</td>
</tr>
<tr>
<td><strong>Toxicity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD50</td>
<td>MAE</td>
<td>Morgan + MLP</td>
<td>0.649 ± 0.019</td>
<td>0.602 ± 0.006</td>
<td>1st</td>
</tr>
<tr>
<td>hERG</td>
<td>AUROC</td>
<td>RDKit2D + MLP</td>
<td>0.841 ± 0.020</td>
<td>0.806 ± 0.005</td>
<td>3rd</td>
</tr>
<tr>
<td>Ames</td>
<td>AUROC</td>
<td>AttrMasking</td>
<td>0.842 ± 0.008</td>
<td>0.859 ± 0.002</td>
<td>1st</td>
</tr>
<tr>
<td>DILI</td>
<td>AUROC</td>
<td>AttrMasking</td>
<td>0.919 ± 0.008</td>
<td>0.933 ± 0.011</td>
<td>1st</td>
</tr>
</tbody>
</table>

*a*Only models that have been evaluated on most of the tasks are considered.
listed in Table 2. In each task, there are at least seven other models or featurization methods being compared with our model, including DeepPurpose [10], AttentiveFP [30], ContextPred [9], NeuralFP [16], AttrMasking [9], and a graph convolutional network [14]. For all 22 tasks, XGBoost is ranked first for 18 and top 3 for 21 out of 22 tasks, demonstrating the success of XGBoost models in predicting ADMET properties.

The superior prediction results of XGBoost are explainable. As shown in Table 2, previously, there are 13 tasks which the top models are trained using descriptors (RDKit 2D + MLP model) or fingerprints (Morgan + MLP model and AttentiveFP). Inspired by this, XGBoost was trained using a combination of fingerprints and descriptors. These featurization methods cover both structural features (MACCS, extended-connectivity, Mol2Vec, and PubChem fingerprints) and chemical descriptors (Mordred and RDKit descriptors) for each given SMILES representation. For a specific property prediction task, XGBoost can take all possible molecular features into consideration, select the best set of them for prediction, and avoid over-fitting by controlling tree complexity. Together, these would boost the prediction performance of XGBoost to be superior to other models.

To further understand the importance of each fingerprint and descriptor for each ADMET task, averaged feature importance was calculated for each feature set and is plotted in Fig. 1. It is shown that Mordred descriptors are consistently the most important feature in all tasks, followed by Mol2Vec and Circular fingerprints. The MACCS keys fingerprint set is the least important among the five groups of features. As Mordred descriptors are considered significantly more important than other features, we retrained the models in each task using only this feature set. The results are listed in Table 3. XGBoost with only Mordred outperformed the base model in three tasks (HIA, Aqsol, and PPBR). However, metabolism and excretion predictions were not improved using XGBoost with Mordred alone but were comparable in other tasks.

Searching for the parameter set with the best validation performance is necessary. However, there are over 100,000 parameter combinations in the current search space, and it could grow exponentially with additional features being considered. It is challenging to iterate over all possible parameter sets to find the best one. In the current study, a randomized grid search CV was used. It should be noted that the randomized grid search CV does not necessarily lead to the global optimum parameter set due to its random nature. In recent decades, Bayesian optimization has been developed to search in the hyperparameter space, such as hyperopt [1], which might be promising for such tasks.

It should be noted that while TDCommons provides a benchmark dataset useful for evaluating and comparing different machine learning models, there are some limitations when applying this dataset. First, in the ADMET prediction scenario, all classification and regression predictions are single-instance: we only predict one value for each task. Thus, multitask learning is not feasible under the current framework. Moreover, the dataset has not been consistently updated. The dataset version should be mentioned when reporting related results.

![Fig. 1 Average feature importance of fingerprints and descriptors in absorption (A), distribution (B), metabolism (C), elimination (D), and toxicity tasks (E)](image-url)
Table 3  Performance comparison of XGBoost models trained with all features and with only Mordred

<table>
<thead>
<tr>
<th>Task</th>
<th>Metric</th>
<th>XGBoost with all features</th>
<th>XGBoost with Mordred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absorption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Caco2</td>
<td>MAE</td>
<td>0.288 ± 0.011</td>
<td>0.301 ± 0.008</td>
</tr>
<tr>
<td>HIA</td>
<td>AUROC</td>
<td>0.987 ± 0.002</td>
<td>0.990 ± 0.002</td>
</tr>
<tr>
<td>Pgp</td>
<td>AUROC</td>
<td>0.911 ± 0.002</td>
<td>0.909 ± 0.005</td>
</tr>
<tr>
<td>Bioav</td>
<td>AUROC</td>
<td>0.700 ± 0.010</td>
<td>0.692 ± 0.016</td>
</tr>
<tr>
<td>Lipo</td>
<td>MAE</td>
<td>0.533 ± 0.005</td>
<td>0.538 ± 0.003</td>
</tr>
<tr>
<td>AqSol</td>
<td>MAE</td>
<td>0.727 ± 0.004</td>
<td>0.720 ± 0.003</td>
</tr>
<tr>
<td>Distribution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBB</td>
<td>AUROC</td>
<td>0.905 ± 0.001</td>
<td>0.900 ± 0.001</td>
</tr>
<tr>
<td>PPBR</td>
<td>MAE</td>
<td>8.251 ± 0.115</td>
<td>7.897 ± 0.061</td>
</tr>
<tr>
<td>VDss</td>
<td>Spearman</td>
<td>0.612 ± 0.018</td>
<td>0.610 ± 0.005</td>
</tr>
<tr>
<td>Metabolism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYP2C9 inhibition</td>
<td>AUPRC</td>
<td>0.794 ± 0.004</td>
<td>0.781 ± 0.002</td>
</tr>
<tr>
<td>CYP2D6 inhibition</td>
<td>AUPRC</td>
<td>0.721 ± 0.003</td>
<td>0.694 ± 0.005</td>
</tr>
<tr>
<td>CYP3A4 inhibition</td>
<td>AUPRC</td>
<td>0.877 ± 0.002</td>
<td>0.862 ± 0.002</td>
</tr>
<tr>
<td>CYP2C9 substrate</td>
<td>AUPRC</td>
<td>0.387 ± 0.018</td>
<td>0.334 ± 0.004</td>
</tr>
<tr>
<td>CYP2D6 substrate</td>
<td>AUPRC</td>
<td>0.648 ± 0.023</td>
<td>0.594 ± 0.034</td>
</tr>
<tr>
<td>CYP3A4 substrate</td>
<td>AUPRC</td>
<td>0.680 ± 0.005</td>
<td>0.649 ± 0.013</td>
</tr>
<tr>
<td>Excretion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half life</td>
<td>Spearman</td>
<td>0.396 ± 0.027</td>
<td>0.373 ± 0.008</td>
</tr>
<tr>
<td>CL-Hepa</td>
<td>Spearman</td>
<td>0.420 ± 0.011</td>
<td>0.378 ± 0.020</td>
</tr>
<tr>
<td>CL-Micro</td>
<td>Spearman</td>
<td>0.587 ± 0.006</td>
<td>0.576 ± 0.010</td>
</tr>
<tr>
<td>Toxicity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD50</td>
<td>MAE</td>
<td>0.602 ± 0.006</td>
<td>0.602 ± 0.006</td>
</tr>
<tr>
<td>hERG</td>
<td>AUROC</td>
<td>0.806 ± 0.005</td>
<td>0.763 ± 0.007</td>
</tr>
<tr>
<td>Ames</td>
<td>AUROC</td>
<td>0.859 ± 0.002</td>
<td>0.856 ± 0.002</td>
</tr>
<tr>
<td>DILI</td>
<td>AUROC</td>
<td>0.933 ± 0.011</td>
<td>0.928 ± 0.003</td>
</tr>
</tbody>
</table>

Web server

The trained machine learning models are hosted on the Southern Methodist University high performance computing cluster at https://ai-druglab.smu.edu/admet. A SMILES representation is required for ADMET predictions. On the result page, molecule structures in both 2D and 3D are displayed using Open Babel [19]. A table is present to summary prediction results under 22 tasks. The optimal levels are referenced from Drug-Like Soft Rule and empirical ranges from ADMETLab 2.0 [29]. For each prediction, green, yellow, and red colors are used to indicate whether the prediction lies in optimal, medium, or poor ranges, as suggested in ADMETLab 2.0. The web server has been tested rigorously to respond within seconds.

Conclusion

In this study, we applied XGBoost for ADMET prediction. XGBoost can effectively learn molecule features ranging from fingerprints to descriptors. For the 22 tasks in the TDC ADMET benchmark, our model is ranked first in 18 tasks with all tasks ranked in top 5. The web server, ADMETboost, can be freely accessed at https://ai-druglab.smu.edu/admet.

Acknowledgements  Computational time was generously provided by Southern Methodist University’s Center for Research Computing. The preprint version of this work is available on arXiv with DOI number 2204.07532 under CC BY-NC-ND 4.0 license.
Author contribution HT and RK conducted the experiment. HT plotted figures. All authors revised the manuscript.

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.

Availability of data and materials The data used in this study is publicly available in TDC ADMET benchmark group [https://tdcommons.ai/benchmark/admet_group/overview/]. The dataset can be downloaded through the TDC Python package (v0.3.6). The default training and testing data were used for model training. We shared the related codes, model parameters for each task, and the ready-to-use featurization results on GitHub at https://github.com/smu-tao-group/ADMET_XGBoost. The web server can be accessed at https://ai-druglab.smu.edu/admet.

Declarations

Ethics approval Not applicable.

Competing interests The authors declare no competing interests.

References


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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.
A.7. PAPER VII

ABSTRACT: Molecular dynamics (MD) simulation is widely used to study protein conformations and dynamics. However, conventional simulation suffers from being trapped in some local energy minima that are hard to escape. Thus, most of the computational time is spent sampling in the already visited regions. This leads to an inefficient sampling process and further hinders the exploration of protein movements in affordable simulation time. The advancement of deep learning provides new opportunities for protein sampling. Variational autoencoders are a class of deep learning models to learn a low-dimensional representation (referred to as the latent space) that can capture the key features of the input data. Based on this characteristic, we proposed a new adaptive sampling method, latent space-assisted adaptive sampling for protein trajectories (LAST), to accelerate the exploration of protein conformational space. This method comprises cycles of (i) variational autoencoder training, (ii) seed structure selection on the latent space, and (iii) conformational sampling through additional MD simulations. The proposed approach is validated through the sampling of four structures of two protein systems: two metastable states of Escherichia coli adenine kinase (ADK) and two native states of Vivid (VVD). In all four conformations, seed structures were shown to lie on the boundary of conformation distributions. Moreover, large conformational changes were observed in a shorter simulation time when compared with structural dissimilarity sampling (SDS) and conventional MD (cMD) simulations in both systems. In metastable ADK simulations, LAST explored two transition paths toward two stable states, while SDS explored only one and cMD neither. In VVD light state simulations, LAST was three times faster than cMD simulation with a similar conformational space. Overall, LAST is comparable to SDS and is a promising tool in adaptive sampling. The LAST method is publicly available at https://github.com/smu-tao-group/LAST to facilitate related research.

1. INTRODUCTION

Molecular dynamics (MD) simulation has a wide application on the study of protein conformations and dynamics. Recent developments in biocomputing, such as Anton, AMBER, and OpenMM, have enabled the simulation time scale to milliseconds, which promotes the research in sampling protein motions and structure landscapes. However, the time scales of many protein functions exceed the time scales achievable through traditional MD simulations. Moreover, protein sampling suffers from being trapped within local energy minima, proving difficult to escape. As a result, most of the computational time is typically spent in sampling previously visited regions, which hinders the efficient exploration of protein conformational space.

Many enhanced sampling methods have been developed to address this issue. These methods can be classified into two types. In the first type, biasing potentials are introduced to expand the sampling space, such as metadynamics and Gaussian-accelerated MD. In the second type, seed structures are selected as restarts for iterative MD simulations. This is referred to as adaptive sampling, and numerous methods have been proposed that differ in seed selection methods. Markov state models have been applied to cluster conformations into microstates; parallel cascade selection MD (PaCS-MD) and nontargeted PaCS-MD calculate the root-mean-square deviation (RMSD) to select top snapshots; frontier expansion sampling conducts dimensionality reduction with principal component analysis and Gaussian mixture models to select frontier structures; structural dissimilarity sampling (SDS) selects new seeds based on principal component analysis.

Recent innovations in deep learning have provided new insights into sampling protein conformational space. Autoencoders (AEs) and variational autoencoders (VAEs) are a class of deep learning models that learn a representation (encoding), which can capture the key features of input data.
Several studies have demonstrated the success of AEs and VAEs in their applications to protein conformations and functions. In our previous work, we showed that VAEs are capable of learning a low-dimensional representation (referred to as the latent space) of protein systems. Through a quantitative study, the learned latent space is shown to be able to represent conformational characteristics. This property indicates that the larger differences the two protein conformations have, the farther their corresponding latent points are from each other.

In this study, we proposed a new adaptive sampling method, latent space-assisted adaptive sampling for protein trajectories (LAST), to accelerate the exploration of protein conformational space. Initially, a short MD simulation is conducted starting from the crystal structure. Afterward, the following steps are repeated iteratively until certain criteria are met. First, a VAE is trained using sampled protein conformations. Then, seed structures are selected in the learned latent space. Finally, starting from these selected seed structures, additional simulations are conducted to sample more protein conformations that will be used in the next round. To quantify the performance, we applied LAST on four conformations in two protein systems: two metastable states of *Escherichia coli* adenosine kinase (ADK) and two native states of Vivid (VVD). To better explore the protein conformational space, ADK conformations sampled from the simulation were projected onto its two intrinsic angles, and VVD conformations were projected onto the space using two RMSD values with reference to the two native structures in dark and light states, respectively. These collective variables are unrelated and unknown to the VAE models. Our results showed that seed structures were consistently located on the boundary of the sampled conformational distributions in all four conformations regardless of protein projection methods. We further compared the sampling efficiency among LAST, SDS, and conventional MD (cMD). In both systems, large conformational changes were observed in a shorter time in LAST simulations. To be specific, LAST explored two transition paths toward two stable states, while SDS explored one and cMD neither in the metastable ADK simulations. In VVD simulations, LAST only took one-third of cMD simulation time to discover a similar conformational space.

2. METHODS

2.1. Variational Autoencoder. An autoencoder is a type of deep learning model that aims to encode a high-dimensional input to a low-dimensional latent space through an encoder module and decode it back to the original dimensions through a decoder module. By minimizing the differences between inputs and outputs, known as reconstruction loss, the latent space is expected to learn a low-dimensional representation of the input space. However, the latent space in an AE is not well constrained and leads to unsatisfying results when sampling in the latent space. To overcome this issue, variational autoencoders add an optimization constraint on the latent space to follow a certain distribution.

The encoder module \( q_\phi(z|x) \) is an inference model that transforms data \( x \) into output latent variable \( z \), being parameterized with \( \phi \). In reverse, the decoder module \( p_\theta(x|z) \) is a generative model that transforms latent variable \( z \) into output data \( \hat{x} \), being parameterized with \( \theta \). Both models are trained simultaneously with a joint distribution as \( p(x, z) = p_\phi(x|z)p(z) \). \( p(z) \) is the constraint distribution for latent space and typically is chosen as a normal distribution. The tractable variational Bayes approach is used to approximate the intractable posterior \( p_\phi(z|x) = p_\phi(x|z)p(z)/(\int p_\phi(x|z)p(z)dz) \) by maximizing the evidence lower bound (ELBO)

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

where \( \text{KL} \) is the Kullback–Leibler divergence.

In our implementation, the VAE model is developed using Keras package with Tensorflow backend.
2.2. Molecular Dynamics Simulations. The initial structures of four conformations in two protein systems, two metastable adenosine kinase (ADK) and two native states (PDB ID 2PD7 and 3RH8) of Vivid (VVD), were taken from the Protein Data Bank (PDB). Simulation files were generated using tleap with the AMBER ff14SB force field. NVT Langevin MD simulations (100 ps) were carried out, followed by 200 ps of NPT simulations at 1 atm and 300 K. In each round of LAST method, one 100 ps MD simulation was conducted for each seed structure. Particle mesh Ewald (PME) algorithm was used to calculate long-range electrostatic interactions. The simulation time step was set as 2 fs. All simulations were conducted with OpenMM 7.

2.3. Latent Space-Assisted Adaptive Sampling for Protein Trajectories. LAST method includes three steps, and its workflow is shown in Figure 1. First, a variational autoencoder is trained using all previous simulations. Second, the lowest-probability samples are selected on the latent space and their corresponding protein structures are treated as seeds. Third, additional MD simulations are conducted from seed structures.

2.3.1. VAE Training. In each iteration, some preprocessing procedures are needed. The simulation trajectories are first aligned to the first frame, and heavy atoms are extracted with Cartesian coordinates being expanded as a feature vector (Figure 1A,B). Then, each feature is transformed to a range of 0–1 using min–max linear scaling, which is used to construct a data set for VAE training. The architecture of the VAE model is shown in Figure 1C. In the current study, we design the encoder model being composed of three hidden layers with a size of 512, 128, and 32 and the decoder model with a size of 32, 128, and 512. The number and size of hidden layers can be adjusted based on the size of proteins. The dimension of latent space is set as two for simplicity and ease of visualization.

2.3.2. Seed Selection. Appropriate seed selection method is needed to expedite the sampling of protein conformational space. In LAST, seeds are selected on the two-dimensional learned latent space of VAE, which has two important characteristics to enable an efficient seed selection. First, as demonstrated in our previous work, the distance between two data points on the latent space is meaningful. Two structurally similar proteins have a shorter distance between their corresponding latent vectors. Second, the sampling distribution of latent space in the VAE is similar but does not strictly follow a normal distribution. It is likely that the KL divergence term in the loss function contributes to the normal distribution, and the reconstruction loss component in the loss function may contribute to the deviation from the normal distribution. As for the distribution of the VAE latent space of protein conformations, the most common protein structures are encoded in the center of the latent space, while structurally different proteins are encoded on the boundary. In a data distribution, samples with the lowest probabilities refer to those points that differ significantly from other data. Based on the above two points, it is reasonable to treat the lowest-probability samples on the latent space as seeds to accelerate conformational space exploration, as their conformations deviate from the majority of the sampled ones.

To implement the seed selection method, we propose three improvements to make LAST computationally efficient:

1. Latent space of VAE is not strictly normal after optimization even though the normality is incentivized in the loss function. Therefore, a nonparametric multivariate kernel density estimator, instead of multivariate normal density function, is used to fit the latent space. The estimator is developed in Python statsmodels library.

2. Latent space distribution might be skewed so that the top N lowest-probability samples with the smallest probability densities tend to gather on one side of the distribution. To avoid the above issue, the cumulative distribution function (CDF) of the fitted nonparametric multivariate kernel density estimator on the latent space, instead of probability density, is applied to guarantee that samples from both sides of CDF (values close to 0 and 1) are equally selected. In this case, the first order of the density estimator was accumulated in the latent space.

3. Protein conformations corresponding to the lowest-probability samples can be located and selected based on data index. These protein conformations might be similar to each other, resulting in sampling repeated conformational space from MD simulations starting from these conformations. Thus, to further boost sampling efficiency, we require new seed structures to have at least 1 Å RMSD with all previously selected seeds.

One example of seed selection result is shown in Figure 1D, where seeds are highlighted in red stars in the latent space visualization.

2.3.3. Additional MD Simulations. Short MD simulations are conducted in each round. In the current study, 10 seeds are selected in one round and a 100 ps simulation is done starting from each seed. Thus, the total simulation time in each round is 1 ns. The detail of these simulations is described in Section 2.2.

The above three steps are iteratively done until convergence. Here, we design the convergence criterion by calculating the mean RMSD of Cα atoms with regard to the starting protein structure. The iterative sampling process is terminated once the mean RMSD stops to increase for successive five rounds or reaches the maximum round number.

Algorithm 1: Latent space assisted adaptive sampling for protein trajectories

Prepare simulation files.
Conduct 100 ps NVT and 200 ps NPT simulations while iteration is not reaching the maximum round do
Align trajectories and extract Cartesian coordinates.
Train a VAE model.
Fit latent space with a non-parametric multivariate kernel density estimator.
Select top 10 lowest-probability samples based on CDF and get seed structures.
Conduct 100 ps simulation for each seed.
if mean RMSD is converged then
Stop iteration.
end if
end while

The LAST algorithm is summarized in Algorithm 1 with codes that are freely available at the GitHub site of https://github.com/smu-tao-group/LAST.

2.4. Structural Dissimilarity Sampling. Structural dissimilarity sampling (SDS) is an efficient method to quickly expand protein conformational distributions toward unvisited conformational spaces. Similar to LAST, SDS iterates between (1) arrangement of seed structures for a diverse distribution in the frontiers of conformational regions and (2) conduction of
additional MD simulations based on these selected structures. In this work, SDS was applied to each protein system and the sampled protein conformational spaces were compared with the LAST and cMD results under the same simulation time. The SDS was implemented using scripts from Zhang and Gong\textsuperscript{17} under https://github.com/Gonglab-THU-MD/Frontier-Expansion-Sampling.

3. RESULTS

Four structures of two protein systems (ADK and VVD) were prepared for MD simulations, as described in Section 2.2. For each protein structure, 100 ps of NVT and 200 ps of NPT simulations were conducted. During the iterative process, all previous simulations were aligned to the first frame with Cartesian coordinates of heavy atoms being extracted as a feature vector to represent protein conformation. Afterward, a variational autoencoder model was trained. Ten seed structures were selected with an additional 100 ps simulation starting from each of them. Therefore, each iteration takes a 1 ns simulation time.

ADK protein is composed of a rigid CORE domain, a lid-shaped ATP-binding domain (LID), and an AMP-binding domain (NMP). Many computational studies have shown ADK to carry out large conformational transitions between the

Figure 2. Structures of (A) ADK and (B) VVD. ADK is composed of a CORE domain, an LID domain, and an NMP domain. LID−CORE and NMP−CORE angles are calculated by four vectors to represent protein conformations. Both proteins are colored at the secondary structure level using ChimeraX.

Figure 3. Seed structure distribution on the low-dimensional protein representations. (A, B) ADK protein conformations are represented in LID−CORE and NMP−CORE angle vectors. (C, D) VVD protein conformations are represented in RMSDs with regard to the native dark and native light states. Seed structures are represented in red stars. The analysis in these plots was carried out after seven rounds of LAST simulations for illustration purposes.
closed state to the open state during the ATP−ADP catalyzation process.\textsuperscript{33,34} Four vectors that form NMP−CORE and LID−CORE angles, as shown in Figure S1, have been widely used to characterize ADK protein conformation. VVD is a light-oxygen-voltage domain that undergoes global conformational changes upon perturbation. VVD is shown to be flexible in the native light state and relatively stable in the native dark state.\textsuperscript{34} ADK and VVD structures are illustrated using ChimeraX.\textsuperscript{35} (Figure 2).

Proper low-dimensional protein representations are needed to evaluate the quality of seed selection. In the current study, ADK protein structure is projected to LID−CORE and NMP−CORE two-dimensional (2D) angle plots. We followed the same residue selection rule to calculate vectors and angles.\textsuperscript{24} For the VVD structure, 2D root-mean-square deviation (RMSD) with reference to the native dark and light structures was used to show the sampled protein conformational space.

Both the angle plot in ADK and RMSD plot in VVD were used to display the protein conformation of seed structures (Figure 3). In each subplot, seed structures are highlighted as red stars. In two metastable ADK conformations (Figure 3A,B), seed structures are mainly located in the less sampled regions with small or large LID/NMP angles. This indicates that the variational autoencoder can capture the structural
differences of protein conformations within the learned latent space. In the native dark and native light VVD conformations (Figure 3C,D), seed structures are also shown to be evenly distributed in the boundary of protein conformational space defined by RMSD to two native VVD structures.

To compare the effectiveness of LAST to conventional molecular dynamics simulations, the sampled protein conformational space in each round of the LAST method is displayed together with cMD sampled conformations. Figure 4 shows the protein conformations in 1, 5, 10, and 15 ns for both LAST and cMD. It is shown that under the same simulation time, LAST can explore more protein conformations than cMD. Moreover, the trained variational autoencoder can consistently learn a low-dimensional protein representation in the latent space, regardless of the growing number of simulations and changing shape of conformational space and guide MD simulations to explore less sampled regions. In contrast, there are limited new conformations being explored in cMD simulations from 10 to 15 ns, indicating that it might be trapped in a local energy minimum.

We continued the LAST simulation of ADK until the convergence of LAST. For comparison, both SDS and cMD simulations were conducted under the same simulation time. The sampled protein conformational spaces are shown in Figure 5A. The LAST sampling method took 22 iterations (22 ns simulation time) and explored two paths from the metastable state to the two native states. This aligns with the computational finding that ADK protein undergoes conformational transitions between the open and the closed states. Moreover, the sampled conformational space in LAST spans in the intermediate regions between the closed and open states, with some coverage in the open state and no coverage in the closed state. Meanwhile, SDS only explored one path toward the open state, and cMD mostly sampled the overlap of LAST and SDS methods. The sampled two transition pathways align well with a previous study, in which a 200 ns AMBER simulation was conducted, showing that the LID-open NMP-closed metastable ADK structure could visit both native open and closed states. The same experimental setting was applied to the open and closed states of ADK protein. While these two states are stable, LAST can still cover the majority of cMD results and sample more conformations when compared with SDS simulations, as shown in Figure S2. The sampled conformations in the LID-open NMP-closed metastable ADK structure were also projected using the first two components in PCA (Figure S3). In contrast to Figure 5A, the SDS sampled conformations do not fully overlap with LAST. Instead, both methods sampled different conformational regions and are complementary to the cMD results.

There are 120 ADK structures in the PDB. The minimum RMSDs in LAST and SDS produced trajectories that were calculated with reference to each ADK structure and are listed in Table S2. More than two-thirds (84 out of 120) of minimum RMSDs in LAST are less than those in SDS. On average, the minimum RMSD in LAST is 0.07 Å less than that in SDS. These indicate that the LAST method is comparable to SDS and allows the structural integrity of protein to be reasonably maintained.

For the VVD system, LAST simulation took 30 iterations (30 ns simulation time) to converge. The conformational space is illustrated in Figure 5B. SDS and LAST methods sampled similar conformational spaces and both covered a majority of cMD sample regions. To compare the efficiency of LAST and cMD methods, this cMD simulation was continued while this 2D RMSD map was being monitored. It took 100 ns simulation time for cMD simulation to have a similar space shape to LAST. The initial 30 and 100 ns simulations are displayed in Figure S4B. In terms of the MD simulation time, LAST was three times faster than cMD. Considering the VAE training time, the overall time cost for LAST was around 40% of cMD, with all computations carried out on a Tesla P100 GPU node.

The mean RMSDs with regard to the starting protein structure in each iteration were calculated for both ADK and VVD systems and are shown in Figure 6. Mean RMSDs are presented with black lines, and the standard deviation is shown in red lines for each round. The maximum and minimum RMSD values are shown as the upper and lower bounds in the colored regions. Currently, we set the patience as 5: the iteration loop stops if the maximum mean RMSD does not increase in five consecutive rounds. For the simulation in the ADK system, RMSD starts with 2 Å, gradually increases to 3.5 Å, and stops at iteration 22. In contrast, the RMSDs in the VVD system are smaller and the total simulation lasts longer with a total of 30 iterations.

4. DISCUSSION

In this study, we proposed a new adaptive sampling method to explore protein conformational space. LAST iteratively trains a VAE model using previous simulations, selects seeds that are structurally different from the sampled conformations, and uses them to initiate additional short MD simulations. LAST differs from previous methods in seed selection design, where the lowest-probability samples are selected and treated as seeds on
the latent space of VAE. VAE has been demonstrated to be effective in learning a low-dimensional protein representation in the latent space.\textsuperscript{50,37,38} The embeddings in the latent space are known to keep a distance similarity: if two protein structures are similar in structure, their embeddings in the latent space are close to each other. With this nature, the lowest-probability samples on the latent space are worth further exploration through MD simulations, as their corresponding protein structures deviate from the most common structures. In LAST, these low-probability samples are treated as seed structures to conduct additional MD simulations.

The normality of latent space provides a new opportunity for seed selection. However, the latent space does not strictly follow a normal distribution, as shown in Table S1 and Figure S5. This is mainly because of the relatively strong emphasis on reconstruction loss and lesser emphasis on KL divergence during VAE training. The reconstruction loss term controls the quality of latent space data reconstruction (how well the VAE can reconstruct a protein structure), and KL divergence term constrains the distribution of the latent space (to what degree the latent space needs to follow a normal distribution). Therefore, to have a well-constructed and normal regularized latent space, appropriate weights are needed to be set for both terms. This is a challenging task with fine tuning by hand, as the sample size keeps growing linearly with additional MD simulations in each round. Therefore, instead of trying to find weights to balance the reconstruction loss and KL divergence, we allow the latent space to not strictly follow a normal distribution and use a nonparametric multivariate kernel density estimator to fit the latent space.

One potential problem is that the distribution of the latent space might be skewed or kurtotic. In such cases, one side of probability density function will have a long tail with low values. This could lead to the situation that all selected seed structures lie on the long tail side, and the corresponding protein structures of these seeds might be similar to each other. Seed gathering on one side of latent space distribution decreases the chance to explore more structurally different conformations and thus leads to a less efficient protein sampling process. To partially overcome this issue, we used the cumulative distribution function to select the lowest-probability samples: data points on the two sides of the CDF are evenly selected. This improvement, as shown in Figure S6AB, prevents sampling similar seeds on the boundary of protein conformational spaces.

Still, seed structures might be similar to each other. Nontargeted PaCS-MD proposed a nonredundant selection rule, which calculates pairwise RMSDs between the current simulation cycle and seeds selected in all of the past cycles.\textsuperscript{39} Protein configurations with large RMSDs are then selected as new seeds in the current cycle. We took reference from this idea when selecting seeds. The lowest-probability samples from two ends of the estimated CDF are picked sequentially, while the pairwise RMSDs to previously selected seeds are calculated. We set the RMSD threshold as 1 Å and require that the RMSD values of the newly selected seeds should be greater than the threshold. If not, LAST discards this sample and moves to the next. The effect of this improvement can be seen through the comparison of Figure S6B,C. Moreover, LAST is a memory method: the selected seed structures are stored for RMSD calculation in future iterations, which avoids repeated sampling in the same conformational region and further improves the sampling efficiency.

For ADK, two angles with prior knowledge of its conformational dynamics were chosen to reveal the sampling efficiency. Similarly, RMSD values with reference to VVD native dark and light structures, respectively, were used for the same purpose. These preselected order parameters do not reduce the generality of LAST method because they were not used to develop VAE models. In the other words, the VAE models are “unaware” and do not require this information.

There are some tuning parameters in the LAST sampling scheme, including the dimensions of the latent space, the number of seed structures, the RMSD threshold in seed selection, the architecture of VAE model, and the number of rounds in convergence. In LAST method, the seed structures need to be selected in the frontier regions of conformational space, which has been sampled. These so-called frontier regions could not be easily identified in the Cartesian coordinates. On the contrary, after being projected onto a low-dimensional latent space, the frontier regions of the conformational space representing existing simulations could be easily identified based on the distribution of existing simulations. Consequently, the seed structures for further simulations could be chosen in these frontier regions in the low-dimensional latent space. The latent space is one of the hidden layers in a VAE model. Typically, its dimension is much lower than the input dimension and is considered the bottleneck. In this study, the latent space was set as 2D to visualize, project, and compare high-dimensional protein conformations. The performance of higher dimensions in the latent space is worth further study. For the number of seed structures, 10 seeds are selected in each round. This could be changed under different protein systems and is subjected to the available computing resources. Also, the MD simulation time starting from seeds, currently set as 100 ps, can be adjusted accordingly. However, it should be noted that this simulation time should match the RMSD threshold: the simulation time should not be too short with a large RMSD threshold. Given that the conformational space of selected seeds is not likely to be visited again, it is expected to have a reasonable simulation time to fully explore the conformations in each additional MD run. Besides, the number of hidden layers in the VAE model is important to learn a useful latent space. Our previous finding suggests that a VAE model with three hidden layers is sufficient to learn the ADK protein conformations. Larger model architectures do not have a significant improvement but instead will lead to longer training time. The proper architecture of VAE, in terms of the number of hidden layers and the number of dimensions in the latent space, is worth studying to provide general guidelines when dealing with different protein families. In general, LAST method is applicable in all protein systems. The implementation of LAST method is similar regardless of whether the protein systems contain nonprotein components. However, the user needs to obtain appropriate force field parameters for the system under simulation. Lastly, it is worth noting that the convergence criterion used in this study does not represent the “true” convergence of protein systems. The notion of “true” convergence, as discussed in previous studies,\textsuperscript{50–53} is elusive in simulations. More appropriate criteria are needed for the convergence signal in adaptive sampling, through either numerical indicators or self-consistency checks.
5. CONCLUSIONS
In this study, we present an adaptive sampling method, latent space-assisted adaptive sampling for protein trajectories, to accelerate the exploration of protein conformational spaces. LAST iterates through variational autoencoder training, seed selection, and additional short MD simulations. LAST differs from previous methods in seed selection where the lowest-probability samples in the learned latent space are selected and treated as seed structures. LAST method is compared with SDS and cMD using ADK and VVD protein systems, each with different low-dimensional representations. In both systems, LAST can capture the key protein characteristics and select seeds that lie in the boundary of conformational space. For ADK simulations, LAST explored two transition paths that are in agreement with previous findings. For VVD simulations, LAST is three times faster than conventional MD for exploring the same conformational regions. To conclude, LAST provides an alternative method for efficient adaptive sampling.

ASSOCIATED CONTENT
Data Availability Statement
The LAST algorithm is publicly available on GitHub at https://github.com/smu-tao-group/LAST. The SDS algorithm is available at https://github.com/Gonglab-THU-MD-Frontier-Expansion-Sampling. All simulation trajectories generated in this study are available from the corresponding author without restriction.

Supporting Information
The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c01213.

AUTHOR INFORMATION
Corresponding Author
Peng Tao — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75206, United States; orcid.org/0000-0002-2488-0239; Email: ptao@smu.edu

Authors
Hao Tian — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75206, United States
Xi Jiang — Department of Statistical Science, Southern Methodist University, Dallas, Texas 75206, United States
Sian Xiao — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75206, United States
Hunter La Force — Department of Chemistry, Center for Research Computing, Center for Drug Discovery, Design, and Delivery (CD4), Southern Methodist University, Dallas, Texas 75206, United States
Eric C. Larson — Department of Computer Science, Southern Methodist University, Dallas, Texas 75206, United States

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c01213

Notes
The authors declare no competing financial interest.

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

REFERENCES


