Self Supervised Solution for Analysis of Molecular Dynamics Simulations In-Situ

H. Sahni, T. Estrada (University of New Mexico) H. Carrillo-Cabada (Intuit) E. Kots, M. Cuendet, H.Weinstein (Weill Cornell Medical Center)
E. Deelman (University of Southern California) S. Caino-Lores, M. Taufer (University of Tennessee)

THE UNIVERSITY OF NEW MEXICO

Abstract

With modern technology and High Performance Computing (HPC), Molecular Dynamics (MD) simulations can be task and data parallel. That means, they can be decomposed into multiple independent tasks (i.e., trajectories) with their own data, which can be processed in parallel. Analysis of MD simulations includes finding specific molecular events and the conformation changes that a protein undergoes. However, traditional analysis rely on global decomposition of all the trajectories for a specific molecular system, which can be performed only in a centralized way. **We propose a lightweight self-supervised machine learning technique to analyse MD simulations In-Situ.** That is, we aim to speedup the process of finding molecular events in the protein trajectory at run-time, without having to wait for the entire simulation to finish. This allows us to scale the analysis with the simulation.

Introduction

- A Protein is a long chain of amino acid residues. This chain decides the overall structure of the protein.
- MD is used to simulate folding process of proteins over a period of time (Fig-2).
- Each time step which is termed as a Frame, defines the state of the protein. The chain or collection of frames build a trajectory.
- Changes in factors like ph, temperature and composition of a solution can make a protein go through a number of molecular events which induce conformational changes in the protein trajectory.
- Structural Rearrangements (Fig-1), Binding Events, Protein Associations are some of the conformational changes.
- We use domain knowledge to determine the residues of interest ‘r’ (i.e., residues that will be tracked to determine if a molecular event occurs).

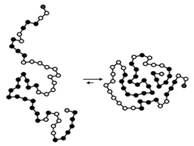


Fig 1: Structure of the protein and structural rearrangements

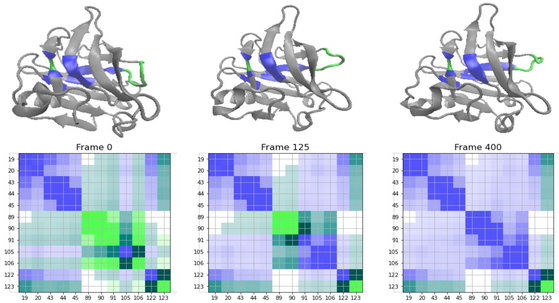


Fig 2: Example of Frames in a trajectory with residues of interest in (top) and GEM encoding of residues of interest (bottom) intensity of the matrix is the Euclidean distance, color represents their secondary structure (red: helices, blue: sheets, green: coils)

Our Method

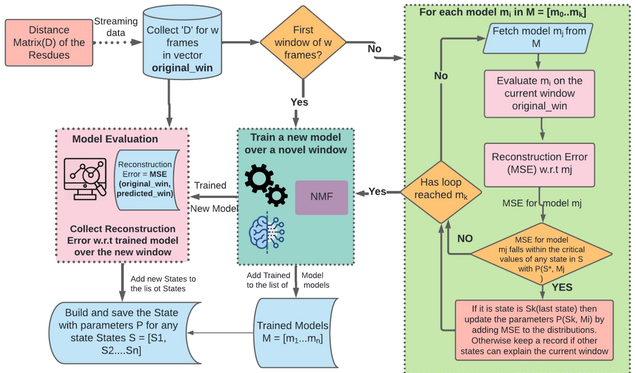


Fig 3: Block diagram explaining the main method

For each frame we calculate the Euclidean distance between each pair of residues of interest and build a vector X of size |r|^2 * 1. We train NMF model (M0) with first ‘w’ frames and construct the first State (S0). We build a state for every new molecular event that is detected in the trajectory. Every state comprises of parameters P(Sk,Mj) associated with the “normal” behavior of the protein in that state. P(Sk,Mj) consists of running mean, running standard deviation, as well as the critical values for a 80% level of confidence (p<0.2) for a two tailed z-test. New states are built when the existing ones cannot explain the data.

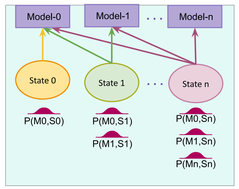


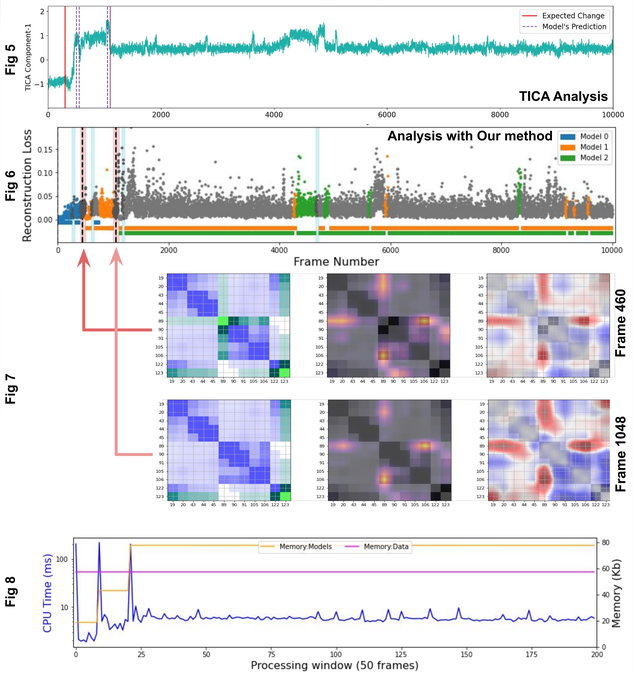
Fig 4: Every state Sk comprises of k+1 associations with previously built NMF models.

Evaluation and Results

We tested our method on trajectories from different proteins and mutants listed below.

Protein	#Traj	#Res	Residues of Interest	# of Frames	TP	FP	FN
Bovine Beta-lactoglobulin	6	162	12	2000-10000	6	2	0
Wild Type: B cell translocation gene (BTG1) mutant	23	129	8	1200-10000	20	5	2
E50K BTG1 mutan	18	129	8	1000-14000	11	3	1
R68L BTG1 mutant	6	129	8	14000	4	2	0
Opsin	1	326	21	2000	3	1	0

Fig 5
Fig 6
Fig 7
Fig 8

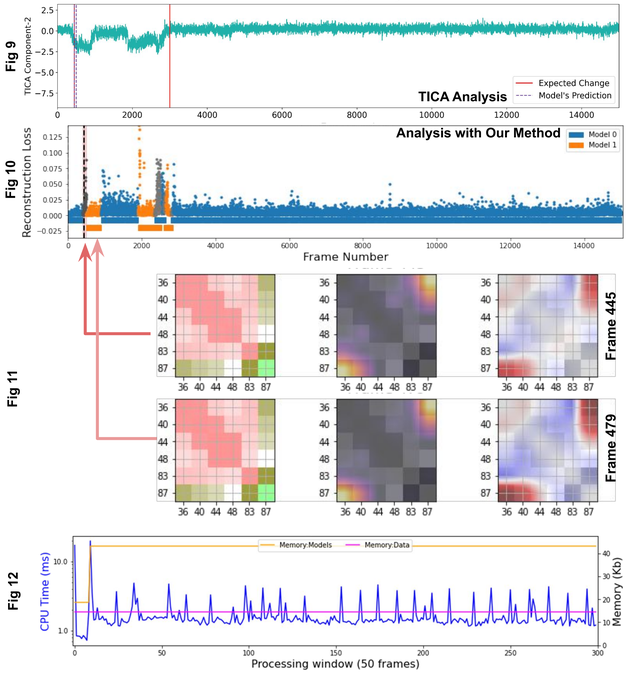


We validated our results with respect to TICA and with the expected conformation changes determined by domain scientists. Figures 6-8 demonstrates our results for trajectories from Globulin and Figs 9-12 for E50K.

Figures 6 and 10 show the states identified by our method. Colored horizontal bars represent states that are able to explain the data (e.g., blue is Model 0 explains the data, orange is Model 1, and green is Model 2). Red vertical bars indicate when a new state was created. Cyan vertical bars are regions of uncertainty that did not create a new state. Figures 7 and 11 show residue interactions that explain the detected molecular events. Every matrix represents the interaction between each pair of residues. On the left, the GEM encoding depicts changes in secondary structure. The middle figure highlights pairs of residue interaction that contributed the most to the detection of the event. Finally, the right matrix shows whether residue pairs moved closer (red) or farther away (blue) than expected.

We perform performance profiling (see figure 8, 12) to measure memory and CPU usage, which are in the order of kilobytes and milliseconds respectively, making our method ideal for lightweight in-situ analysis.

Fig 9
Fig 10
Fig 11
Fig 12



Conclusions

Our method shows a behavior that is consistent with TICA. But unlike TICA, the analysis can be performed in the same node as the simulations or run concurrently on a different node; saving time and computational resources. We train an ensemble of light-weight ML models that do not require the entire view of the protein to determine the relevant changes.

- Our algorithm can monitor molecular events in a protein trajectory In-Situ.
- With this knowledge, we identify different states of the protein through its trajectory.
- Additionally, we are able to explain which residue-pairs contributed the most for a detected conformational change in the protein trajectory.
- Our method is robust regardless of the type of protein, sampling rate, number of residues , as well as other simulation properties.
- Our method is efficient (i.e., execution per window takes a few milliseconds) and requires only constant memory (e.g., 50 frames at a time)