Investigating Protein Structure Populations from Simulation Data using Unsupervised Learning

Gideon K. Gogovi
Department of Mathematics and Statistics
University of Houston Downtown
Houston, USA
gogovig@uhd.edu

Joshua Kiddy K. Asamoah
Department of Mathematics
Kwame Nkrumah University of
Science and Technology
Kumasi, Ghana
jkkasamoah@knust.edu.gh

Gabriel Obed Fosu
Department of Mathematics
Kwame Nkrumah University of
Science and Technology
Kumasi, Ghana
gabriel.of@knust.edu.gh

***Abstract*—Data obtained from molecular dynamics simulation provides important intuition into the dynamical interactions of biological molecules. The chronicles of sequential time-dependent atomic motions of configurations obtained from simulation and the derived properties estimated from molecule's trajectory is specified by this sequence. Therefore, knowing how to efficiently extract representative structures from simulation data is important because often, we will want to identify changes in conformation of a protein structure when simulation is performed. We use unsupervised machine learning techniques to cluster such data and investigated a few of protein structural properties. The algorithms implemented in this paper presents clusters of the simulation data that tends to group frames from an adjacent block of time together, even when sampling at $10\,\mathrm{ps}$ intervals. We found that sampling of conformational space for a shorter run simulation may not be able to completely visit all structures that belong to a specific cluster. But for the sufficiently long simulation, the systems revisit previous clusters repeatedly. Cluster populations change rapidly at the initial stage of the simulations, but became steady before each got to their terminal values, indicating equilibrium attainment. Investigation of protein structure properties also attest the correspondence between clusters of protein structures obtained from the clustering algorithms.**

***Index Terms*—Unsupervised learning, Molecular dynamics, Protein conformational transitions, K-means, Hierarchical agglomerative, DBSCAN.**

I. Introduction

Data from trajectories of molecular history that is obtained from molecular dynamics (MD) simulations provide important intuition into the dynamical interactions of biological molecules such as proteins, polymers, and lipids. [1], [2]. The chronicles of sequential time-dependent atomic motions of molecular configurations obtained from simulation and the larger set of derived properties estimated from trajectory data is specified by this sequence of data [3]. This data can also be used to tally or benchmark reputed force field changes. Hence knowing how to efficiently extract representative structures from MD simulation data is important because often times, we will want to identify changes in conformation of a protein structure when molecular dynamics simulation is performed.

Primarily, MD generates atomic trajectory positions as a function of time and sometimes the atomic velocities. These trajectories provides a representation of the molecule's energetically accessible conformational sample. Since simulations are now on microsecond (μs) time scale level and sampled configurations stored in picosecond (ps), these simulations outputs very large amounts of data. The wide-ranging conformational changes that occur during these simulations can lead to a high variance in the calculation of properties such as estimation of energetics and other structural properties [4] that are time independent.

Unsupervised machine learning (ML) methods provide techniques to make meaning of the information in MD trajectories by way of grouping them. A very important ML task in structural bioinformatics and computational systems biology is clustering protein structures. For example, protein structure predictions often involve a stage where clustering is performed to find the best prediction. Unsupervised learning models include tasks such as clustering and dimensionality reduction where the modeling involves features of a datasets without reference to any label. Clustering algorithm group data points into a disjointed collection of subgroups. The data points within each cluster are similar to each other as compared to data points in another cluster.

Unsupervised ML is becoming important for analyzing the increasingly large trajectory data obtained from atomistic simulation. The idea of using these types of algorithms to group similar molecular conformations proteins visit in time during a simulation has been in existence and has been used as far back in 1993 [5], [6]. Subset of publications that developed and apply unsupervised ML techniques to analyze trajectory data includes some earliest applications to very recent ones [5]–[7]. During clustering of protein configurations from trajectory data, each algorithm is expected to ideally group proteins with similar configurations into distinct clusters. This will give a clarified view of how a molecule conformational space is being sampled and allows characterization of the different states visited during the simulation [8]. Aldo *et al.* [9] provide a thoroughgoing overview of the unsupervised ML that have most frequently been used to investigate this simulation trajectory data. Clustering, together with methods such as dimensionality reduction has also been employed to provide representation of structure and dynamics of protein systems [10]. Partitional and hierarchical clustering algorithms uncovers similarities in

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00199

protein structures based on information held in conformational transitions [11]. To the best of our knowledge, no cluster analysis on sufficiently longer (at least to the microsecond scale) MD simulation data has been performed at the time of completing this manuscript. This presents an if-else case of whether clusters consist of simulation window or frames from a single block of time in the simulation or systems do revisit previous clusters.

In this paper, clustering algorithms were implemented and used to compare the different conformational transitions obtained from the simulation of X-ray diffraction protein from Monkeypox virus [12] in explicit water for very long run. Caused by the monkeypox virus, monkeypox is a zoonotic disease which belongs to the family Poxviridae, subfamily of chordopoxvirinae, and genus orthopoxvirus [13], [14]. The monkeypox virus is in a close relation with the smallpox virus and results in a smallpox-like disease. The virus was initially discovered in a laboratory in Denmark from monkeys in the late 1950s [15], hence the origin of name monkeypox. The first case of the virus was diagnosed in humans in the Democratic Republic of the Congo (DRC) in 1970 [16]. Thereafter, the virus has become endemic in Congo, and has spread to many Central and West Africa African countries. The first cases reported outside Africa was in 2003 [14].

The clustering algorithms employed in this study are K−means, Hierarchical Agglomerative (bottom-up) and DBSCAN, Density-Based Spatial Clustering of Applications with Noise as implemented in CPPTRAJ in AMBER package [17]. This study is important because, by clustering the protein simulation trajectories into distinct sub-state populations, the variance in simulation structures can be minimized and this will provide more useful information about the ensemble of sampled conformations from the simulation in time. Determining structures that are assuredly distinct and those that are only varies slightly on the same framework is time-consuming and a critical task in regard to computational forage. For these reasons, both supervised and unsupervised ML algorithms have been widely used to analyze existing databases in this context [18].

The rest of the paper is presented as follows. The computational details of the MD simulation of the protein to generate data for the clustering analysis in section II is present. The necessary mathematical and statistics concepts that form the background of clustering algorithms employed in this study is also presented in this section. This is followed by section III where we analyze trajectories data from MD simulations and discuss the findings. The paper is concluded in section IV with a summary from the findings.

II. Materials and Methods

A. Initial Protein Structure

The starting coordinates of the X-ray diffraction structure of the Protein from monkeypox virus were downloaded from the protein data bank (PDB ID: 4QWO) [12]. This protein consist of 132 amino acids (ALA1−ASN132) which together with hydrogen atoms, makes up a total of 2120 atoms. Water molecules attached to the initial X-ray structure were removed and the amino acid, selenomethionine (MSE) was replaced with normal amino acid, methionine (MET) because MD engine used for did not have MSE in it's database. We added acetyl and N−methyl amide capping groups to the protein termini in order to avoid any artificially strong interaction. The N−terminal amino-acid is taken to be a capping acetyl group while the C−terminal, N−methyl amide capping group.

B. Details of MD Simulations and Data Generation

The protein was placed in a cubic box containing SPC/E water model molecules [19] of size $72.5\,\text{Å}$ per side with the box border being at least $0.2\,\text{nm}$ from any protein atom. The choice of SPC/E model is based on a previous MD simulation study [20], [21] which demonstrated how this water model preserves protein structure and mimics experiments well. The ff14SB force field [22] was adapted for the modeling and simulation of the protein. The simulations were all performed with a GPU-enabled CUDA version of the Particle Mesh Ewald Molecular Dynamics (PMEMD) module in AMBER 18 package [23]. Prior to the MD production runs, the each system was energetically minimized before the equilibration. We started the minimization with 500000 steps of steepest descent followed by 300000 conjugate gradient steps to remove possible clashes between atoms that maybe very close to each other. Position restraints were used on heavy atoms during annealing, when the systems were gradually being heated from $0\,\text{K}$ to $298.15\,\text{K}$. This was done in $50\,\text{ps}$ for a total of 6 stages with periodic boundary conditions. Performing the heating in stages allows the systems to equilibrate at desired temperatures, hence reducing the chances of system blow up. We then performed NPT-MD equilibration on each of the five independent systems for $40\,\text{ns}$ at a pressure of $1.013\,25\,\text{bar}$ and $298.15\,\text{K}$ temperatures with isotropic position scaling before the production runs. All the simulations started from the one restart file that was generated after exhaustive energy minimization, systems heating and equilibration. The production run was up to $1.0\,\mu\text{s}$ at constant volume (NVE) and temperature using the weak-coupling algorithm [24] with bonds involving hydrogen constrained. The long-range electrostatics within the particle mesh implementation (PME) [25] are used for Ewald sum calculations with a $14\,\text{Å}$ non-bonded cutoff distance and an integration time step of $2\,\text{fs}$. From the trajectory data, the root-mean square deviations $(RMSD)$ of the protein structures are calculated. We run five independent simulations in order to attain convergence by increasing sampling time. The idea of running many independent duplicates of the same simulation is an acceptable practice we aim at providing a vigorous trajectory data can present results that are in-tune with experiment. All trajectory data were saved for each simulation frame and sampled for the clustering analysis. All post simulation analysis of the data obtained from the trajectories were performed using the CPPTRAJ tool [17] implemented in the AMBER package. We also used in-house written codes and bash scripts in some cases.

C. The Unsupervised Learning Algorithms

The first unsupervised learning algorithm we used in this study is $K-$means.

It has been applied to cluster data in many application domains and was first introduced in 1982 [26]. It is relatively not a very complicated algorithm and has shown to be very functional especially in high-dimensional spaces application because of its speed. This algorithm consists of initially choosing K cluster centers at random from complete trajectory data and calculate distances of each data point to these centers as a distance (D) matrix with the formula;

$$\mathrm{D}^{(n,k)} = \sqrt{\sum_{m=1}^{M} \left(x_j^{(n)} - c_j^{(k)}\right)^2} \qquad \text{for} \quad k = 1 \quad to \quad K \tag{1}$$

An $(n \times n)$ distance matrix where n is the number of points in the datasets is needed for exact clustering. Usually, it is not necessary for the distance matrix to be saved since it allows the clustering of sufficiently large data to be possible at faster rate. At this stage, each of the data points are assigned with the nearest cluster/center:

$$\arg\min_k \mathrm{D}^{(n,k)}$$

Computation of new cluster centroids the next step. This is done by taking current members into consideration. We then revert to the distance calculation step and repeat until convergence. This is non-deterministic algorithm and so, the result depends on initial cluster centroids seed which are randomly selected. The stopping criteria used in implementation of $K-$means in this paper is terminate when 10 clusters are formed and with initial set of protein structure randomization. The RMSD of all the residues in the protein and selecting only atoms C, N, O, CA and CB (without considering hydrogen atoms) for the distance metric calculations.

The second clustering algorithm, hierarchical agglomerative, we considered is also well-known pairwise distance metric clustering algorithm with a wide usage. It is a very slow algorithm with a $\mathcal{O}(n^3)$ time complexity for a general case when dealing with large dataset . The merges and splits in this algorithm are determined in a greedy manner. The bottom-up approach of the hierarchical clustering is used to cluster the protein trajectory data from the simulation. This approach constructs clusters into a hierarchy by merging two smaller clusters into a larger one repeatedly. The clustering was performed on the protein backbone atoms using the average-linkage with a stopping criterion; when either 10 clusters are obtained or when the minimum distance between two clusters is greater than $3.5\,\mathrm{nm}$.

The third clustering algorithm employed in this study is DBSCAN, proposed by Martin *et al* [27]. It is a non-parametric density-based algorithm that identify individual clusters in the trajectory data. It is based on the idea that a cluster in trajectory data space is a neighboring region of high point density and separated from other cluster by low point density in contiguous regions. The method can discover clusters of different formations and sizes from large trajectory data with outliers and noise. The DBSCAN algorithm requires two parameters; a threshold, which is the minimum number of data points clustered together for a region to be considered dense, and also, a distance measure (ϵ) that is used to locate the datasets in neighborhood of any data point. We use a threshold, $minpoints = 4$, and a distance cutoff, $\epsilon = 0.15\,\mathrm{nm}$ for the DBSCAN analysis. How these parameters are selected is explained later in section III.

III. Results and Discussion

We extract representative protein structures from the simulation data using cluster analysis as a method for determining structure populations. This section presents results from the clustering analysis. The analysis were performed on only the backbone atoms of the protein. Table I presents results from some metrics used to compare the three algorithms applied to the trajectory data with each corresponding to $1\,\mu\mathrm{s}$. For each of the methods, we calculated the pseudo F statistic (pSF), Davies-Bouldin Index (DBI) and R-squared (SSR/SST) values. The DBI is defined as the average, for all clusters X, of f, where $f(X) = \max$, across other clusters Y, of

$$\frac{(Cx + Cy)}{dXY}.$$

Where Cx and Cy are average distances from points in X and Y to the centroid respectively, and dXY is the distance between cluster centroids. Smaller DBI values represent a better clustering [28]. This metric aims at identifying clusters that are compact and well-separated and it is a good idea to compare DBI values for different clustering algorithms only when we have similar cluster sizes since this is affected by cluster count.

From the analysis, DBSCAN produced the smallest DBI value as compared to the other two algorithms. Another metric for measuring clustering which intends to measure the rigidity of clusters is the pseudo$-F$ statistic (pSF). As shown in equation 2 below, the (pSF) is a ratio of the mean sum of squares between clusters to the mean sum of squares within clusters. It is based on a comparison of intra-cluster variance to the residual variance over all trajectory data points [29]. This metric is estimated from a regression model's coefficients of the sum of squares regression (SSR) and the sum of squares error (SSE) through the ratio for all trajectory data points n and clusters g.

$$pSF = \frac{SSR/(g-1)}{SSE/(n-g)}. \tag{2}$$

The n in equation (2) is the number of trajectory data points (thus frames or protein structures over time) and g, the number of clusters. The R^2 value which is the ratio of sum of squares regression to the sum of squares error represents the fraction of variance explained by the trajectory data.

Since K$-$means algorithm is deterministic, both performance and time for convergence was refined by selecting the initail

seed carefully. We adopted Arthur and Vassilvitskii's [30] suggestion on the initial cluster centroids selection by sampling seeds depending on the distance to all seeds selected. All three clustering methods in this paper used the same approach of first calculating similarities and then using it to cluster the trajectory data into clusters. As mentioned earlier, the DBSCAN algorithm requires values for for the minimum number of trajectory data points to form a cluster and the cutoff distance for a cluster to be formed. Usually, an idea of these parameters are determined from the creation of a K−distance plot (see Figure 1) which shows for each trajectory frame the K^{th} farthest distance and sorted in terms of descending distance measure. In view of this, K−distance analysis was then performed in the 132 residues considering only back-bone atoms and no hydrogen atoms considered. A total of six runs was conducted and a plot of the six resulting outputs shown in Figure 1. Literature suggests that the shape of the distance curve does not change significantly beyond $K = 4$ [31] and this is evident in Figure 1. We observe from the figure that there is a steep slope from the beginning and the curves then levels out at ≈ 1.5 Å. The K−distance values did not show significant difference between them. Hence, we incorporate the minimum number of points $minpoints = 4$ and a distance cutoff, $\epsilon = 0.15\,\mathrm{nm}$ into the input of the DBSCAN algorithm and present the results together with the K−means, and hierarchical agglomerative algorithms in Figure 2.

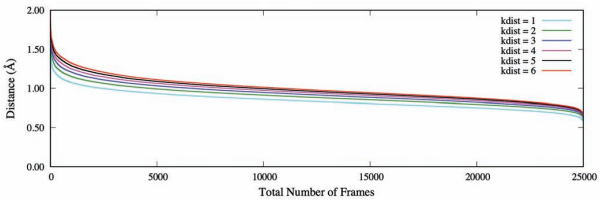

Fig. 1. Kdist analysis performed on the residues.

The high pSF and low DBI values are signaling a better clustering. We can infer from Table I that hierarchical agglomerative algorithm and DBSCAN performed very similar with low DBI values and higher percentage of variance explained (R^2). The cluster size for clusters representing more than 10% herein referred as most populated clusters are generally similar for all the three algorithms but very similar for K−means and DBSCAN. We could suggest that DBSCAN is somewhat more suitable than the hierarchical agglomerative algorithm for clustering the simulation datasets, even though it has the lower pSF.

In Figure 2, it can be observe that at the cluster populations are changing quickly at beginning of the trajectory over the frames. As the simulations progress, the cluster inhabitants stabilizes gently until they get to their final frame values of 25000, the total number of frames from the all independent sampled trajectories data. This indicates that the cluster populations are approaching equilibrium as the simulation extends. These equilibrium results may be suitable for other analysis such as thermodynamics of protein folding. The number of clusters obtained from the analysis are 10 for K−means, 14 for hierarchical and 13 for DBSCAN. The most populated cluster (Pop:0) for the K−means includes 5094 protein structures representing 20.4% of the total number of frames processed with average distance between proteins in this cluster being 2.196 Å (with 0.529 of standard deviation).

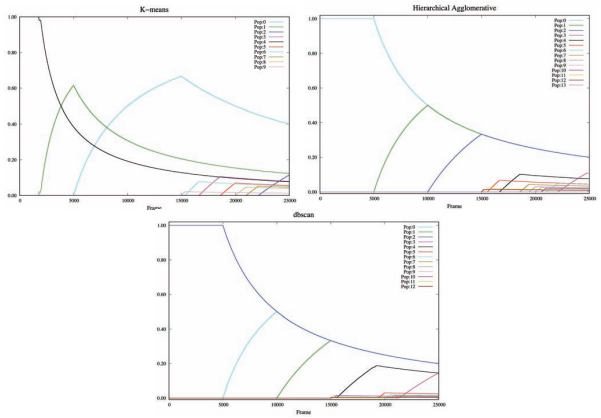

Fig. 2. Protein cluster populations versus time.

At frame 8839 the protein structure with the lowest cumulative distance to every other structures can be found with an average distance to every other cluster being 4.477 Å. Similarly for the hierarchical and DBSCAN algorithms, the Pop:0's includes 5000 frames each, representing 20% each for the total frames processed and with average distances between cluster proteins being 2.732 Å (with 0.709 of standard deviation) and 2.134 Å (with 0.494 of standard deviation) respectively. Also, in frame 2612 of the hierarchical algorithm, the protein structure with the smallest cumulative distance to every other protein structure is found with average distance to every other cluster being 5.375 Å.This is the same for frame 8839 with an average distance of 5.375 Å for DBSCAN. It is worth mentioning that the hierarchical clustering and DBSCAN algorithms produced very similar results, where clusters (Pop:0 and Pop:2), (Pop:1 and Pop:0) and (Pop:2 and Pop:1) from hierarchical clustering and DBSCAN respectively are almost the same population. This results are better visualized by monitoring cluster formation on RMSD along time. The plots shown in Figure 3 depict the clusters formed from each of the three algorithms with the colors representing the cluster numbering in Figure 2.

A. Structural Properties of the Protein from the clustering populations

One of the many possible protein size measures is the radius of gyration, R_g. This can be calculated with the formula

$$R_g^2 = \sum_{i=1}^{N} (\mathbf{r}_i - \mathbf{r}_{cm})^2 / N.$$

TABLE I
METRICS OF CLUSTER ALGORITHM QUALITY

ML Algorithm	pSF	DBI	SSR/SST	Cluster size	Cluster fraction (> 0.10)
K−means	5494.79	1.46	0.66	5094, 5000, 5000, 3520, 2753	0.20, 0.20, 0.20, 0.14, 0.11
Hierarchical	5327.27	1.26	0.73	5000, 5000, 5000, 2719	0.20, 0.20, 0.20, 0.11
DBSCAN	4159.50	1.17	0.69	5000, 5000, 4999, 3652, 3626	0.20, 0.20, 0.20, 0.15, 0.15

where the $\mathbf{r}_i$'s are position vectors of the atoms with reference to center of mass of the protein. While $\mathbf{r}_{cm}$ is the center of mass position vector and N is the number of atoms in the protein. The radius of gyration can also be defined as a measure of the distribution of atoms in a protein around its axis. That is, distance between the point it is rotating and where the transfer of energy has the utmost effect. We measured the protein sizes within each cluster for the three methods by calculating the average radius of gyration for the protein structures in each cluster and the result presented in Figure 4. In general, R_g values increases as cluster size decreases (the cluster sizes decreases from Pop:0, Pop:1, ..., Pop:n). The first three highly populated clusters were also found to be similar in terms of molecular shape measured by the R_g.

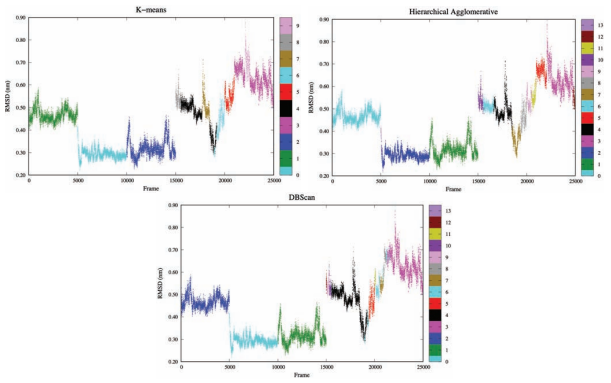

Fig. 3. Distribution of clusters along RMSD of trajectory data. RMSD values are colored based on their cluster memberships along the frames.

Other useful structural properties examined are the protein end-to-end distance (D_{ee}), and solvent-accessible surface area (SASA). D_{ee} describes the flexibility of a protein and is defined as the distance between centers of mass of the two end amino acids in the protein. Because the protein chain can take any shape during the simulation, D_{ee} values may not necessarily follow any trend. However, the smaller values gives an idea on how close the two end amino acids are to each other and larger D_{ee} values represent structures with end-amino acids being far apart. The average D_{ee} values from the proteins in each cluster and across the ML algorithms are presented in Table II. SASA measures protein behavior and is controlled by the interactions or non-interactions of hydrophobic and hydrophilic residues with water. The solvent molecules in the system creates a surface tension near the protein-solvent interface which affects the dynamics and structure of the protein. We adopted the Linear Combinations of Pairwise Overlaps (LCPO) method [32] to approximate the SASA for the protein structures obtained from the clustering algorithms, LCPO estimates the SASA of each atom by calculating an overlap between atoms and their neighbors. If the an atom in the protein is highly overlapped by another, then the less atom exposed to solvent. Similarities observed between the algorithms from other properties and metrics are also observed for D_{ee} and SASA especially for the densely populated clusters (see Table II).

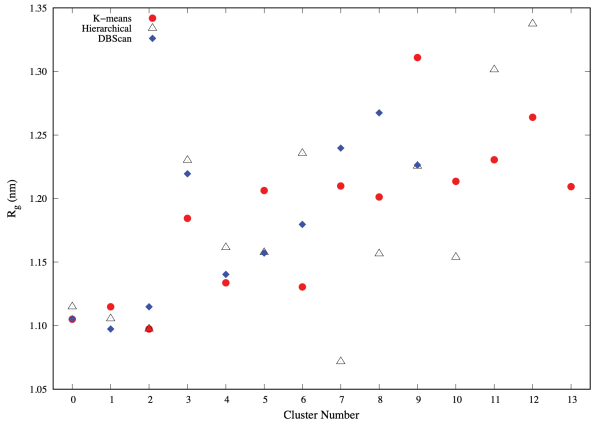

Fig. 4. Average Radius of Gyration of the Protein Structures in each Cluster

IV. CONCLUSIONS

In this paper, we have presented unsupervised machine learning algorithms namely, K−means, hierarchical agglomerative and DBSCAN clustering to investigate protein structure populations from extended molecular dynamics simulations. We used a protein from monkeypox virus for this study. We observed from the study that, all three algorithms implemented presents clusters of the simulation trajectory data that tends to cluster frames from a adjacent block of time together, even when sampling at pico second time intervals.

A peculiar observation from the results is that, at the initial stage of the simulations, the cluster populations are changing quickly over the frames. When the simulation progressed, cluster populations stabilizes slowly until they got to the total number of frames of all independent trajectories. This gives an indication that the cluster populations are getting to an equilibrium and such results may be suitable for studies involving thermodynamic analysis. Also, the study observed similarities between the protein structures obtained from hierarchical clustering, and DBSCAN algorithms results,

TABLE II
PROTEIN SOLVENT-ACCESSIBLE SURFACE AREA ($SASA$) AND END-TO-END DISTANCE (D_{ee}) OF STRUCTURES IN EACH CLUSTERS.

	$D_{ee}(nm)$			$SASA(nm^2)$		
Pop:#	K−means	Hierarchical	DBSCAN	K−means	Hierarchical	DBSCAN
0	0.620	0.344	0.622	6177.07	5846.72	6131.38
1	0.344	0.622	0.380	5846.39	6131.30	5347.55
2	0.380	0.380	0.344	5347.55	5348.39	5845.96
3	1.556	1.681	1.618	5249.45	2766.89	3745.48
4	0.804	1.133	0.862	6825.28	5068.85	6962.67
5	0.997	0.916	0.575	4302.48	4740.62	4117.23
6	0.489	1.384	1.451	5718.08	3124.95	3812.62
7	1.248	0.529	0.664	4043.43	9965.02	2166.20
8	1.246	0.575	1.391	4133.85	4149.05	2484.57
9	1.657	1.289	1.308	1793.69	2961.42	2357.25
10		0.770	1.112		2955.56	4194.22
11		1.230	0.630		2474.16	1835.53
12		1.401	1.390		1809.90	1066.32
13		1.467			2208.91	

where cluster (Pop:0 and Pop:2), (Pop:1 and Pop:0) and (Pop:2 and Pop:1) from hierarchical clustering, and DBSCAN respectively being almost the same population. With recurrent sampling each frame in the simulation is close to an adjoining frame and hence the anticipation of this result. Investigation of structural properties such as radius of gyration, protein solvent-accessible surface area and end-to-end distance further attests the similarities observed between clusters obtained from the three algorithms.

The first half of the total number of frames (Figure 3) processed generally consist of frames from a lone block of time but as the simulation is extended, a reverse trend is observed. This implies that the sampling of conformational space for a short simulation may not be able to completely visit all protein structures that belong to a particular cluster. However, for the sufficiently longer simulation, it is observed that the systems revisited previous clusters repeatedly. For example, in figure 3, members of the cluster located around frames $\approx$5000−10000 appeared in the neighborhood of frames $\approx$18500.

ACKNOWLEDGMENT

The authors would like to thank Dr. Silayi Swabir (Office Research Computing, George Manson University, Fairfax, VA) for his outstanding contributions to the finishing of this paper.

REFERENCES

[1] M. Karplus and G. A. Petsko, "Molecular dynamics simulations in biology," *Nature*, vol. 347, no. 6294, pp. 631–639, 1990.

[2] M. Karplus and J. A. McCammon, "Molecular dynamics simulations of biomolecules," *Nature structural biology*, vol. 9, no. 9, pp. 646–652, 2002.

[3] J. Shao, S. W. Tanner, N. Thompson, and T. E. Cheatham, "Clustering molecular dynamics trajectories: 1. characterizing the performance of different clustering algorithms," *Journal of chemical theory and computation*, vol. 3, no. 6, pp. 2312–2334, 2007.

[4] P. A. Kollman, I. Massova, C. Reyes, B. Kuhn, S. Huo, L. Chong, M. Lee, T. Lee, Y. Duan, W. Wang, *et al.*, "Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models," *Accounts of chemical research*, vol. 33, no. 12, pp. 889–897, 2000.

[5] M. E. Karpen, D. J. Tobias, and C. L. Brooks III, "Statistical clustering techniques for the analysis of long molecular dynamics trajectories: analysis of 2.2-ns trajectories of ypgdv," *Biochemistry*, vol. 32, no. 2, pp. 412–420, 1993.

[6] P. S. Shenkin and D. Q. McDonald, "Cluster analysis of molecular conformations," *Journal of computational chemistry*, vol. 15, no. 8, pp. 899–916, 1994.

[7] G. K. Gogovi, S. Silayi, and A. Shehu, "Computing the structural dynamics of rvfv l protein domain in aqueous glycerol solutions," *Biomolecules*, vol. 11, no. 10, p. 1427, 2021.

[8] M. Poncin, B. Hartmann, and R. Lavery, "Conformational sub-states in b-dna," *Journal of molecular biology*, vol. 226, no. 3, pp. 775–794, 1992.

[9] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, "Unsupervised learning methods for molecular simulation data," *Chemical Reviews*, vol. 121, no. 16, pp. 9722–9758, 2021.

[10] M. Ceriotti, "Unsupervised machine learning in atomistic simulations, between predictions and understanding," *The Journal of chemical physics*, vol. 150, no. 15, p. 150901, 2019.

[11] M. Teletin, G. Czibula, and M.-I. Bocicor, "Using clustering models for uncovering proteins' structural similarity," in *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 185–190, IEEE, 2019.

[12] F. M. Giorgi, D. Pozzobon, A. Di Meglio, and D. Mercatelli, "Genomic characterization of the recent monkeypox outbreak," *bioRxiv*, 2022.

[13] K. M. Ajmera, L. Goyal, T. Pandit, and R. Pandit, "Monkeypox-an emerging pandemic," *IDCases*, p. e01587, 2022.

[14] E. M. Bunge, B. Hoet, L. Chen, F. Lienert, H. Weidenthaler, L. R. Baer, and R. Steffen, "The changing epidemiology of human monkeypox—a potential threat? a systematic review," *PLoS neglected tropical diseases*, vol. 16, no. 2, p. e0010141, 2022.

[15] P. v. Magnus, E. K. Andersen, K. B. Petersen, and A. Birch-Andersen, "A pox-like disease in cynomolgus monkeys," *Acta Pathologica Microbiologica Scandinavica*, vol. 46, no. 2, pp. 156–176, 1959.

[16] J. G. Breman, M. Steniowski, E. Zanotto, A. Gromyko, I. Arita, *et al.*, "Human monkeypox, 1970-79," *Bulletin of the World Health Organization*, vol. 58, no. 2, p. 165, 1980.

[17] D. R. Roe and T. E. Cheatham III, "Ptraj and cpptraj: software for processing and analysis of molecular dynamics trajectory data," *Journal of chemical theory and computation*, vol. 9, no. 7, pp. 3084–3095, 2013.

[18] S. De, F. Musil, T. Ingram, C. Baldauf, and M. Ceriotti, "Mapping and classifying molecules from a high-throughput structural database," *Journal of cheminformatics*, vol. 9, no. 1, pp. 1–14, 2017.

[19] P. Mark and L. Nilsson, "Structure and dynamics of the tip3p, spc, and spc/e water models at 298 k," *The Journal of Physical Chemistry A*, vol. 105, no. 43, pp. 9954–9960, 2001.

[20] G. K. Gogovi, "Structural Exploration of Rift Valley Fever Virus L Protein Domain in Implicit and Explicit Solvents by Molecular Dynamics," in *Advances in Computer Vision and Computational Biology*, pp. 759–774, Springer, 2021.

[21] G. K. Gogovi, *Polymers and Biomolecules in Solvents: A Molecular Dynamics Study*. PhD thesis, George Mason University, 2020.

[22] J. A. Maier, C. Martinez, K. Kasavajhala, L. Wickstrom, K. E. Hauser, and C. Simmerling, "ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb," *Journal of chemical theory and computation*, vol. 11, no. 8, pp. 3696–3713, 2015.
[23] D. Case, I. Ben-Shalom, S. Brozell, D. Cerutti, T. Cheatham III, V. Cruzeiro, T. Darden, R. Duke, D. Ghoreishi, M. Gilson, *et al.*, "Amber 2018," *University of California, San Francisco*, 2018.
[24] H. J. Berendsen, J. v. Postma, W. F. Van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *The Journal of chemical physics*, vol. 81, no. 8, pp. 3684–3690, 1984.
[25] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, "A smooth particle mesh ewald method," *The Journal of chemical physics*, vol. 103, no. 19, pp. 8577–8593, 1995.
[26] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
[27] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *kdd*, vol. 96, pp. 226–231, 1996.
[28] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
[29] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[30] S. Vassilvitskii and D. Arthur, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2006.
[31] D. A. Case, H. M. Aktulga, K. Belfon, I. Ben-Shalom, S. R. Brozell, D. S. Cerutti, T. E. Cheatham III, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, *et al.*, *Amber 2021*. University of California, San Francisco, 2021.
[32] J. Weiser, P. S. Shenkin, and W. C. Still, "Approximate atomic surfaces from linear combinations of pairwise overlaps (lcpo)," *Journal of Computational Chemistry*, vol. 20, no. 2, pp. 217–230, 1999.