RESEARCH

A fast parallel algorithm to reduce protein folding trajectories

Luis Garreta[1], Mauricio Martinez[2], Néstor Díaz[3] and Pedro A Moreno[4*]

Abstract

Background: Simulations are among the most important tools for studying the mechanisms underlying protein folding. Protein folding simulations have experienced substantial progress in recent years. Diverse technologies are employed, and simulations are reaching the microsecond timescale and greater, which generate very long trajectories. However, the analysis of these trajectories is complicated, and tools are necessary to simplify them so that both the main events and the temporal order in which they occur are preserved.

Results: We present an algorithm to reduce long protein-folding trajectories in a fast and parallel way. The algorithm divides a trajectory into segments to be processed in parallel and then selects from each segment the most representative conformations using a rapid clustering strategy. This strategy leverages the temporal order of the conformations to compare them locally, avoiding an all-versus-all comparison. The algorithm reduces a trajectory by a high percentage, preserving both the patterns and the structure obtained by other, more complex reduction techniques. In addition, its performance is close to that of other efficient reduction techniques and is improved when executed in parallel using more than one core.

Conclusions: The proposed algorithm quickly reduces a protein folding trajectory by selecting its most representative conformations and preserves both the structure and temporal order of the trajectory. The reduced trajectories can be used as input for more complex analysis techniques and for other reduction techniques that are impractical for use with long folding trajectories. The algorithm is fast and is designed to run in parallel on conventional PCs with multicore technology, which are present in most typical research laboratories.

Keywords: Protein folding simulations; Protein structure comparison; Protein structure clustering

Background

We present a parallel algorithm to reduce protein folding trajectories that quickly obtains representative conformations, conserving both the three-dimensional (3D) structure and temporal order of the trajectories. Proteins play fundamental roles in all living organisms, but to be functional, they must fold from their linear amino acid (AA) sequence into a unique 3D or native state; this process is known as protein folding. Understanding the mechanisms and rules of this process has been one of the most pursued objectives of computational biology, and an important theoretical tool to study these phenomena is simulation. Simulations are used to generate folding trajectories (Figure 8), which describe the sequence of states that proteins follow as a function of time during the folding process.

Most folding simulations use the molecular dynamics (MD) method, which, due to its computational cost, is limited to small proteins (< 100 AA) and very short times (picoseconds or microseconds). However, technological innovations have allowed significant advances in these simulations, both with respect to time scale and the technology employed to execute them.

In 2011, using the Anton supercomputer, specifically designed for protein folding [1], full simulations of 12 proteins were published, several on the order of milliseconds [2]. And more recently, in 2020, several trajectories related to the study of the SARS-Cov2 virus were released by DE Shaw Research [3]. The trajectories, on the order of microseconds, were simulated on Anton 2 [4], a supercomputer up to ten times faster than its predecessor Anton. As an economic alternative, in 2014, graphic processing units

*Correspondence: pedro.moreno@correounivalle.edu.co
[4]Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Santiago de Cali, Colombia
Full list of author information is available at the end of the article
†Equal contributor

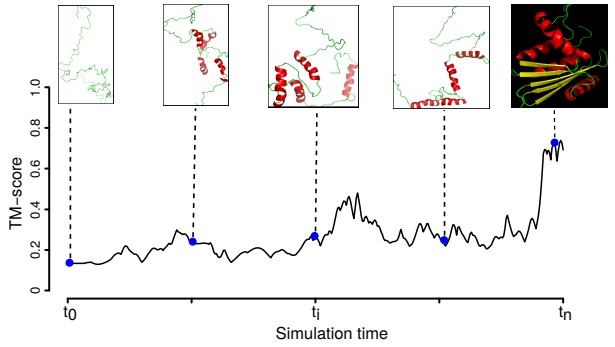

Figure 1 Protein folding trajectory. Some protein conformations with their three-dimensional structure are marked as blue dots along the trajectory. The evolution of folding is measured at each step by comparing the structure at time t_i to the native structure (black image background) using a structure comparison metric; in this work, we use the TM-score as the metric (see Methods). X-axis: Simulation time from t_0 to t_n. Y-axis: TM-score value from 0 (different) to 1 (similar). .

(GPU) were used to simulate the folding of 17 proteins on the order of microseconds, [5]. And years earlier, in 2007, the Folding@home distributed computing platform utilized as many as 250,000 PCs, voluntarily available around the world, to simulate on the order of microseconds the folding of the villin-headpiece protein [6].

These innovations have allowed significant progress in protein folding simulations, both with respect to time scale and the technology used to execute them, resulting in the generation of trajectories with millions of conformations. However, due to the large number of conformations, their processing and analysis with conventional PCs is computationally expensive, and new algorithms are needed to simplify them while preserving as much information as possible.

Two approaches that have been used to reduce protein folding trajectories are dimensionality reduction [7] and clustering [8]. In the dimensionality reduction approach, conformations are transformed into reduced sets of variables that represent the original conformations as well as possible. Both linear and nonlinear techniques have been used (e.g., principal component analysis (PCA) and multidimensional scaling [9], Isomap [10], diffusion maps [11]). However, the reductions produced by these techniques lose the structural information of the conformations (Figure 9, top), and their interpretation becomes difficult since the new dimensions lack a clear relationship with the initial folding trajectory [12].

In the clustering approach, the conformations are assigned to groups that share similar characteristics (e.g., similarity with the native structure), and from each group, an average representative or its general characteristics can be obtained. Hierarchical and partitioned clustering have been used for this approach (e.g., k-means [13], link [14]). However, the groups lose their temporal order since they can include conformations that occur in very distant times (Figure 9, bottom).

In addition, the techniques of both approaches often require pair-wise comparisons, which are computationally costly when the trajectories are very large.

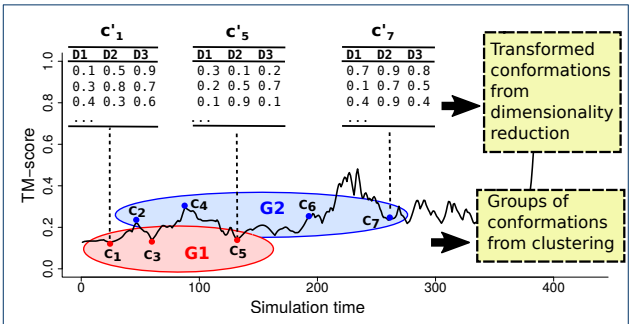

Figure 2 Loss of information in dimensionality reduction and clustering. The dimensionality reduction method (above) transforms the conformations (c_1, c_5, and c_7) into a new set of values (c'_1, c'_5, and c'_7), but their structural information is lost, whereas the clustering method (bottom) forms two groups (G1 and G2), but their temporal order is lost since they contain conformations that overlap in time (c_2, c_3, c_4, c_5).

To reduce a folding trajectory, the proposed algorithm divides the path into segments that are processed in parallel. For each segment, characteristic events are quickly extracted using the rapid clustering strategy of Hobohm and Sander (1992) adapted here for protein folding trajectories. From these results, the most representative events are selected by a strategy of k-medoids [15]. The results of each segment are joined to form the reduced trajectory with the most representative conformations of the original trajectory while retaining both its 3D representation and temporal order.

The algorithm is implemented in the R language except for the function used for pairwise structure comparison, the TM-score [16], this function is executed many times and is implemented in the Fortran language.

Methods

Datasets of protein folding trajectories

We used the folding trajectories of three proteins obtained from different simulation projects. One of the trajectories was that of the trp-cage protein (PDB: 2JOF), simulated with MD using the Anton supercomputer [2], with a simulation time of 208 µs, a time step of 200 ps, and 1044001 conformations. The second

trajectory was that of the apo papain-like protease of SARS-CoV-2 (6WX4), simulated with MD using the Anton 2 supercomputer [3], with a simulation time of 100 µs, a time step of 1 ns, and 100000 conformations. And the third trajectory was that of the villin-headpiece protein (2F4K), also simulated with MD using the Folding@home distributed platform [17], with a simulation time of 8 µs, a time step of 50 ps, and 15201 conformations.

Time steps and folding steps

A time step in MD trajectories is the time length at which conformations are sampled or evaluated during the simulation, whereas a folding step, as employed in the reduced trajectories produced by our algorithm, represents the most likely conformation occurring during a time interval or from a set of likely candidate conformations.

Pairwise comparison of protein structures using the TM-score

In this work, we used the TM-score metric for the pairwise comparison of protein structure [16]. This metric is used in both the proposed algorithm and in the techniques for reduction of protein folding used for benchmarking the algorithm against existing methods. The TM-score is more sensitive to the global topology than local variations; thus, it estimates the pairwise similarity of protein structures much more accurately than the Root Mean Square-Deviation (RMSD), a common metric used for the same purpose. The TM score between two structures ranges from 0 to 1, where 1 is a perfect match. According to a previous analysis [18], a TM score below 0.17 indicates no similarity between the two structures, while a TM score above 0.5 indicates a degree of similarity between them beyond what is expected by chance.

Other techniques for protein folding reduction

nMDS and clustering techniques were used to obtain the intrinsic information of both the original and reduced trajectories, and their results were compared (see Results).

nMDS reductions were carried out using the R-function monoMDS [19], taking as input the dissimilarity matrix obtained from the pairwise comparisons of all the protein conformations of the folding trajectory. The complete-linkage clustering reductions were carried out using the R-function hclust [20], taking as input a matrix with the first two principal components from a PCA analysis. This PCA analysis was carried out using the R-function pca.xyz [21], taking as input a matrix with the 3D coordinates of the $C\alpha$ atoms of all the protein conformations of the folding trajectory.

The reduced trajectories were calculated with our algorithm from the villin-headpiece trajectory with 15201 conformations [6]. The first reduced trajectory had 7197 conformations (representing a reduction of 52%), and the second had 2258 conformations (representing a reduction of 80%).

Implementation

The proposed algorithm reduces a trajectory of protein folding in three steps: partitioning, extraction, and selection. The first step runs only once, whereas the other two runs several times independently, allowing them to run in parallel. Each step involves a strategy to improve the efficiency of the algorithm when working with large protein-folding trajectories. Figure 10 shows the overview of the algorithm, and the steps are given below.

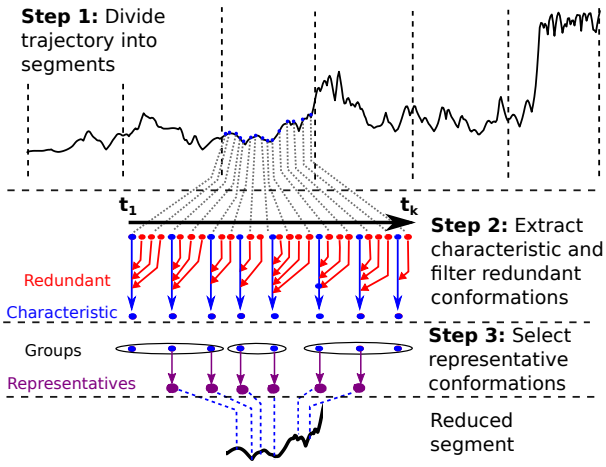

Figure 3 Overview of the proposed algorithm. Step 1: Divide the trajectory into segments (vertical dashed lines) and reduce each individually. Each segment has k conformations from time t_1 to t_k. Step 2: For each segment, extract characteristic conformations (blue dots) and remove redundant ones (red dots). Step 3: Select the most representative conformations among the previous conformations to obtain a reduced segment.

Step 1: Partitioning

In step 1, the trajectory is divided into segments to reduce them locally and in parallel (dotted vertical lines, Figure 10). This process is carried out by dividing the trajectory from the start to the end in segments with k conformations each, where k is an input parameter for the algorithm. Local reductions allow focus on the particular characteristics of each segment that will determine the global characteristics of the trajectory. The parallel reductions allow improved algorithm efficiency when it runs on machines with more than one processor (e.g., multicore computers) (Figure 11).

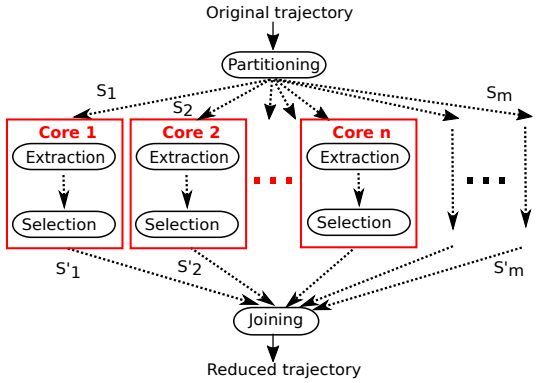

Figure 4 Parallel processing of segments. The trajectory is divided into m segments (S_1, ..., S_m), and each segment is processed separately and in parallel on a machine with n processing cores. The results of the extraction and selection steps (S'_1, ..., S'_m) are independent of each other and are joined to form the reduced trajectory.

Stage 2: Extraction

In step 2, the characteristic conformations of each segment are rapidly extracted, and the redundant ones are eliminated. This step is efficiently carried out by means of a rapid clustering approach that performs few pairwise comparisons and extracts the most dissimilar conformations of a segment instead of grouping similar ones.

Here, we improved the fast clustering algorithm of Hobohm and Sander (1992) to work with a trajectory segment and exploit the implicit order of its conformations given by its simulation time (black horizontal line, Figure 10). Our algorithm selects the initial conformation at time t_1 as the first characteristic. Then, it compares this characteristic with the next conformation. If dissimilar, the last conformation becomes a new characteristic; otherwise, the last conformation is redundant and is removed (red dots, Figure 10). The process continues with the remaining conformations until finishing with the final one at time t_k, thus producing the set of representative characteristics of the segment (purple dots, Figure 10)

Step 3: Selection

In step 3, the conformations of the previously extracted characteristics are clustered to select the most representative characteristics. To identify these representatives, the algorithm uses a k-medoids strategy (PAM algorithm [15]) that calculates the k conformations (medoids) with the minimal average difference between all the other members of the group.

The PAM algorithm needs as input a dissimilarity matrix with the pairwise comparisons of all-versus-all conformations of the trajectory segment; generating this matrix is an intensive computational task when the number of conformations is very large. However, this task is feasible here since the algorithm is working with a reduced set of characteristic conformations (attained in the previous step) of a trajectory segment and not the complete trajectory.

Results and Discussion

Three tests were carried out to evaluate the capacity and performance of the proposed algorithm. In the first test, reduction of three trajectories was performed using the proposed algorithm. In the second, the intrinsic information preserved by the reductions was compared between our algorithm and two other folding reduction techniques. In the third, the performance of the methods was compared.

Reduction of protein folding trajectories

Figure 12 shows the TM-score plots of the reduced trajectories produced by the proposed algorithm. The plots show that the algorithm attempts to find the most representative conformations from the original trajectories while conserving two fundamental properties: the structure and the temporal ordering of the original conformations. The other folding reduction methods investigated lose these properties in their reductions, as described in the next section.

As a result, these reduced trajectories become a summary of the original ones and can be used as inputs for more complex analyses or other reduction methods that require pairwise comparisons but are impractical for large trajectories.

Comparison with other methods

We compared how the intrinsic information captured from a folding trajectory by other folding reduction techniques is also preserved in the reductions produced by our algorithm. First, two reduced trajectories were computed from the original trajectory of the villin-headpiece protein using our algorithm, and then the intrinsic information was computed on these trajectories using nMDS and clustering reductions (Figure 13) (see Methods for the details of the trajectory and techniques).

As shown in Figure 13, the pattern of circles of points from nMDS and the structure of the three groups from clustering are consistent between the original and reduced trajectories. This result shows that the reductions produced by the proposed algorithm largely preserve the intrinsic information observed in the original trajectory. Furthermore, the proposed algorithm has several advantages. First, contrary to nMDS and clustering, our algorithm avoids the calculation of the

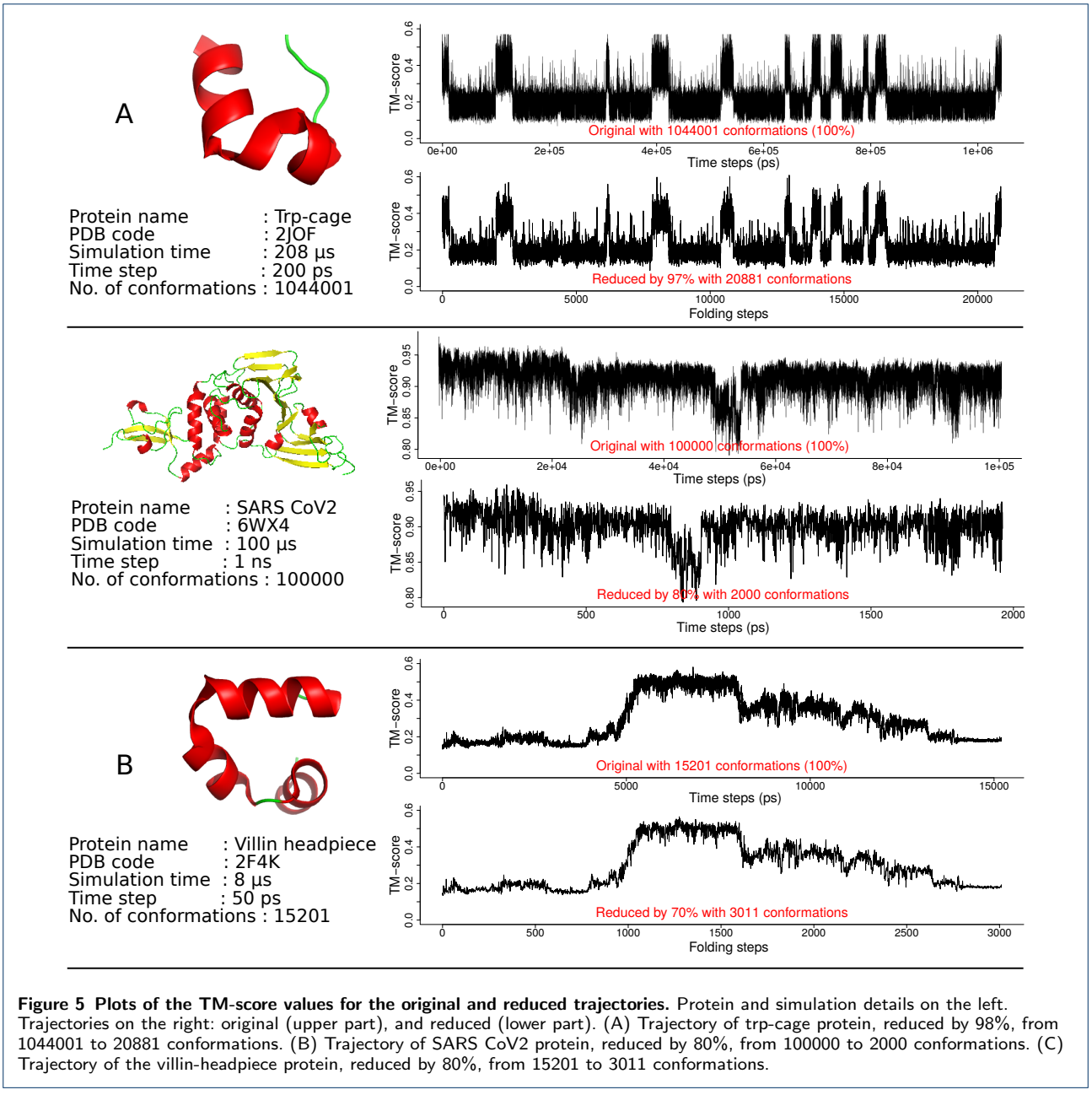

Figure 5 Plots of the TM-score values for the original and reduced trajectories. Protein and simulation details on the left. Trajectories on the right: original (upper part), and reduced (lower part). (A) Trajectory of trp-cage protein, reduced by 98%, from 1044001 to 20881 conformations. (B) Trajectory of SARS CoV2 protein, reduced by 80%, from 100000 to 2000 conformations. (C) Trajectory of the villin-headpiece protein, reduced by 80%, from 15201 to 3011 conformations.

dissimilarity matrix, which is a computationally expensive task for medium to large trajectories. Second, its reductions are a set of protein conformations, in contrast to the transformations produced with other techniques, such as nMDS, PCA, Isomap or diffusion maps [9, 22, 11], which, in addition to losing the structural information of the conformations, the reductions can only be interpreted with the knowledge or experience in the intrinsic algorithm used by the technique [12].

Third, in the proposed algorithm, the temporal ordering of conformations is conserved, in contrast to clustering methods [8] that merge configurations from different simulation times into clusters.

Algorithm performance

The performance of the proposed algorithm was tested via two tests: a runtime comparison of the proposed algorithm with the three other typical techniques for folding reduction, i.e., nMDS, clustering, and PCA (Figure 14.A); and a parallel comparison in which the algorithm was run with several cores (Figure 14.B). For both tests, the trajectory corresponded to the first

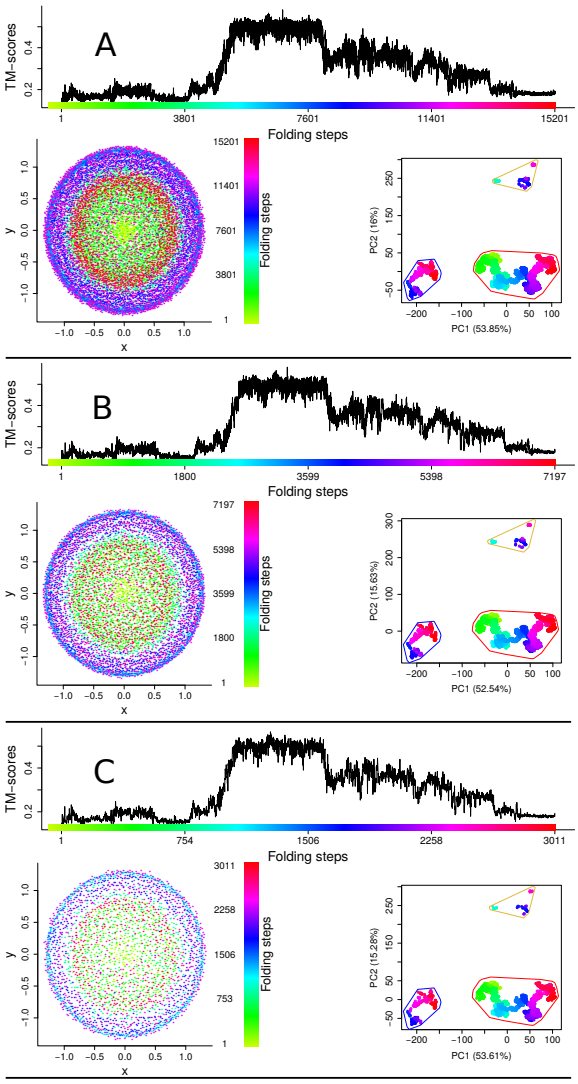

Figure 6 Comparison with other methods. Intrinsic information captured from the original trajectory of the villin-headpiece protein is largely preserved in the reduced trajectories produced by our algorithm. TM-score plot, nMDS pattern and clustering groups are presented at the top, bottom left, and bottom right of each panel. (A) Original trajectory. (B and C) Trajectories reduced by 52% and 80%, respectively. The nMDS produces a set of circular patterns associated with conformations occurring in different times of the simulation, whereas clustering produces a structure of three clusters associated with the first two principal components of the C_α atoms of the conformations.

100k conformations from the full trajectory of the *trp-cage* protein, described above in Methods.

For the comparison among techniques (Figure 14.A), several subtrajectories of different lengths were reduced by all the techniques. When the techniques were run with one core, PCA showed the most efficient runtimes, followed by our proposed algorithm, whereas nMDS and clustering became impractical when used with short to medium trajectories. Furthermore, in contrast to the other techniques, the proposed algorithm has the advantage of being able to run easily in parallel. When it was run using two cores, the runtime was reduced by half and was shorter than that achieved with PCA.

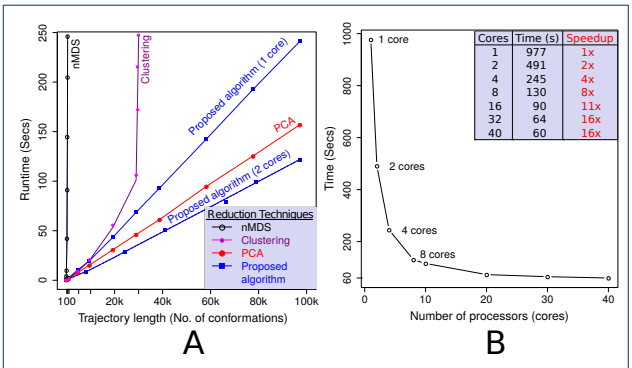

Figure 7 Performance of the algorithm. (A) Runtime comparisons for four folding reduction techniques: nMDS, clustering, PCA, and our proposed algorithm (running with one and two cores). With one core, PCA was the most efficient algorithm, followed by our algorithm. However, when the proposed algorithm was run with two cores, its runtime was reduced by half, being more efficient than PCA. (B) Evaluation of algorithm parallelization. When the number of processing cores is multiplied by two, the processing time reduces by half with a quasi-linear speedup that is maintained for up to ˜8 cores (8x).

To test how parallelization improves the algorithm performance, the full dataset of 100k conformations was reduced by the algorithm using a different number of cores. The runtimes are shown in Figure 14.B, which reveals a good acceleration, with the processing time reduced by half with every doubling of core number. This speedup is maintained up to ˜8 cores and decreases to a minimum after ˜30 cores.

These results show that the algorithm has good performance compared with that of the other techniques and that this performance is improved when the algorithm is run in parallel using more than one core. As a consequence, the speedup of the algorithm scales quasilinearly with the number of processing cores to almost 8x, and with 32 cores, the algorithm achieves a speedup of 16x. Considering that multicore technology is commonplace even for desktop computers, the proposed algorithm has the capacity to take advantage of this technology to reduce large protein folding trajectories in a fast parallel manner, with runtimes closer or better than those of other techniques commonly used for this task.

Conclusions

Although the progress in long timescale simulations of protein folding has enabled the generation of large fold-

ing trajectories, a new challenge lies in their analysis. Due to the millions of conformations they can contain, their processing and analysis is difficult or impractical.

Here, we have proposed a fast and parallel algorithm to simplify large protein-folding trajectories. The algorithm reduces a trajectory by splitting it into segments and then reducing each in parallel using a fast clustering strategy that avoids the pairwise comparison of all structures.

According to the results, our algorithm can summarize a folding trajectory with high data compression and preserve its main conformations and intrinsic information. This was confirmed by comparing the patterns and groups obtained in the original trajectory and the reduced trajectories through the use of nMDS and clustering techniques. Besides, the algorithm runtime shows a linear trend with a performance close to that of PCA, the technique with the best performance among the three evaluated in this work. And, when the algorithm utilizes two or more processing cores, its performance is much better, with quasi-linear acceleration maintained up to ˜8 cores (8x).

Nevertheless, the proposed algorithm is limited to creating a summary of the main events of a protein folding trajectory without performing any kind of analysis, as other techniques do, such as nMDS and clustering However, these summarized trajectories can be used as input to these and other techniques that serve the same purpose and but which were not designed to handle large protein- folding trajectories.

Availability and requirements

- **Project name:** FastFoldingReduction
- **Operating System**: Platform-independent
- **Programming Languajes**: R and Fortran (Source code to work with many OS systems)
- **License**: GNU GPL

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

LG and MM designed the algorithm. LG implemented the algorithm and wrote the manuscript. NM carried out the data collection and analysis. PM guided the study and helped to draft the manuscript. All authors read and approved the final manuscript.

Availability of data and materials

Software is freely available at https://github.com/luisgarreta/foldingreduction. Simulation datasets of the trp-cage protein (PDB: 2JOF) can be requested from D.E. Shaw Research. Simulations datasets of the apo papain-like protease of SARS-CoV-2 (PDB 6WX4) are freely available at https://www.deshawresearch.com/downloads/download_trajectory_sarscov2.cgi/, project DESRES-ANTON-11441075. Simulation datasets of the the villin-headpiece protein (PDB 2F4K) are freely available at https://simtk.org/projects/foldvillin, project 3036.

Acknowledgements

The authors would like to thank the Bioinformatics research group at Universidad del Valle for use of the computer facilities. The authors would also like to thank D.E. Shaw Research for providing the all-atom Molecular Dynamics simulation data.

Author details

[1]Corporación Colombiana de Investigación Agropecuaria - AGROSAVIA, CI Tibaitatá, Kilómetro 14, Vía a Mosquera, Colombia. [2]The European Bioinformatics Institute (EMBL-EBI), Hinxton, Cambridgeshire, UK. [3]Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca, Popayán, Colombia. [4]Escuela de Ingeniería de Sistemas y Computación, Universidad del Valle, Santiago de Cali, Colombia.

References

1. Shaw, D.E., Chao, J.C., Eastwood, M.P., Gagliardo, J., Grossman, J.P., Ho, C.R., Lerardi, D.J., Kolossváry, I., Klepeis, J.L., Layman, T., McLeavey, C., Deneroff, M.M., Moraes, M.A., Mueller, R., Priest, E.C., Shan, Y., Spengler, J., Theobald, M., Towles, B., Wang, S.C., Dror, R.O., Kuskin, J.S., Larson, R.H., Salmon, J.K., Young, C., Batson, B., Bowers, K.J.: Anton, a special-purpose machine for molecular dynamics simulation. Communications of the ACM **51**(7), 91 (2008). doi:10.1145/1364782.1364802
2. Lindorff-Larsen, K., Piana, S., Dror, R.O., Shaw, D.E.: How fast-folding proteins fold. Science **334**(6055), 517–520 (2011). doi:10.1126/science.1208351. arXiv:1011.1669v3
3. Molecular Dynamics Simulations Related to SARS-CoV-2 (2020). https://www.deshawresearch.com/resources.html Accessed 2020-08-17
4. Shaw, D.E., Grossman, J.P., Bank, J.A., Batson, B., Butts, J.A., Chao, J.C., Deneroff, M.M., Dror, R.O., Even, A., Fenton, C.H., Forte, A., Gagliardo, J., Gill, G., Greskamp, B., Ho, C.R., Ierardi, D.J., Iserovich, L., Kuskin, J.S., Larson, R.H., Layman, T., Lee, L.-S., Lerer, A.K., Li, C., Killebrew, D., Mackenzie, K.M., Mok, S.Y.-H., Moraes, M.A., Mueller, R., Nociolo, L.J., Peticolas, J.L., Quan, T., Ramot, D., Salmon, J.K., Scarpazza, D.P., Schafer, U.B., Siddique, N., Snyder, C.W., Spengler, J., Tang, P.T.P., Theobald, M., Toma, H., Towles, B., Vitale, B., Wang, S.C., Young, C.: Anton 2: Raising the Bar for Performance and Programmability in a Special-Purpose Molecular Dynamics Supercomputer. In: Shaw2014 (ed.) SC14: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 41–53. IEEE, Los Alamitos, CA, USA (2014). doi:10.1109/SC.2014.9. http://ieeexplore.ieee.org/document/7012191/
5. Nguyen, H., Maier, J., Huang, H., Perrone, V., Simmerling, C.: Folding simulations for proteins with diverse topologies are accessible in days with a physics-based force field and implicit solvent. Journal of the American Chemical Society **136**(40), 13959–13962 (2014). doi:10.1021/ja5032776
6. Larson, S.M., Snow, C.D., Shirts, M., Pande, V.S.: Folding@Home and Genome@Home: Using distributed computing to tackle previously intractable problems in computational biology (2009). 0901.0866
7. Duan, M., Fan, J., Li, M., Han, L., Huo, S.: Evaluation of Dimensionality-reduction Methods from Peptide Folding-unfolding Simulations. Journal of chemical theory and computation **9**(5), 2490–2497 (2013). doi:10.1021/ct400052y
8. Peng, J.-h., Wang, W., Yu, Y.-q., Gu, H.-l., Huang, X.: Clustering algorithms to analyze molecular dynamics simulation trajectories for complex chemical and biological systems. Chinese Journal of Chemical Physics **31**(4), 404–420 (2018). doi:10.1063/1674-0068/31/cjcp1806147
9. Rajan, A., Freddolino, P.L., Schulten, K.: Going beyond clustering in MD trajectory analysis: An application to villin headpiece folding. PLoS ONE **5**(4), 9890 (2010). doi:10.1371/journal.pone.0009890
10. Das, P., Moll, M., Stamati, H.: Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. Proceedings of the ... **103**(26) (2006)
11. Kim, S.B., Dsilva, C.J., Kevrekidis, I.G., Debenedetti, P.G.: Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. The Journal of Chemical Physics **142**(8), 85101 (2015). doi:10.1063/1.4913322

12. Cavallo, M., Demiralp, Ç.: A Visual Interaction Framework for Dimensionality Reduction Based Data Exploration. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–4 (2018)
13. Doerr, S., Ariz-Extreme, I., Harvey, M.J., De Fabritiis, G.: Dimensionality reduction methods for molecular simulations (2017). 1710.10629
14. Shao, J., Tanner, S.W., Thompson, N., Cheatham, T.E.: Clustering Molecular Dynamics Trajectories: 1. Characterizing the Performance of Different Clustering Algorithms. Journal of chemical theory and computation **3**(6), 2312–34 (2007). doi:10.1021/ct700119m
15. Rousseeuw, P.J., Kaufman, L., Rousseeuw, P.J.: Finding groups in data. Hoboken: Wiley Online Library (1990)
16. Zhang, Y., Skolnick, J.: Scoring function for automated assessment of protein structure template quality. Proteins: Structure, Function, and Bioinformatics **68**(4), 1020 (2007). doi:10.1002/prot.21643
17. Ensign, D.L., Kasson, P.M., Pande, V.S.: Heterogeneity even at the speed limit of folding : Large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. Journal of molecular biology **374**(3), 806–816 (2007)
18. Xu, J., Zhang, Y.: How significant is a protein structure similarity with TM-score = 0.5? Bioinformatics **26**(7), 889–895 (2010). doi:10.1093/bioinformatics/btq066
19. Oksanen, J., Blanchet, F.G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, P., Stevens, M.H.H., Szoecs, E., Wagner, H.: vegan: Community Ecology Package (2019). https://cran.r-project.org/package=vegan
20. R Core Team: R: A Language and Environment for Statistical Computing, Vienna, Austria (2018). https://www.r-project.org
21. Grant, B.J., Rodrigues, A.P.C., ElSawy, K.M., McCammon, J.A., Caves, L.S.D.: Bio3D: An R package for the comparative analysis of protein structures. Bioinformatics **22**(21), 2695–2696 (2006). doi:10.1093/bioinformatics/btl461
22. Duan, M., Han, L., Rudolph, L., Huo, S., Carlson, G.H.: Geometric Issues in Dimensionality Reduction and Protein Conformation Space. (2014)

Figures

Figure 8 Protein folding trajectory. Some protein conformations with their three-dimensional structure are marked as blue dots along the trajectory. The evolution of folding is measured at each step by comparing the structure at time t_i to the native structure (black image background) using a structure comparison metric; in this work, we use the TM-score as the metric (see Methods). X-axis: Simulation time from t_0 to t_n. Y-axis: TM-score value from 0 (different) to 1 (similar). .

Figure 9 Loss of information in dimensionality reduction and clustering. The dimensionality reduction method (above) transforms the conformations (c_1, c_5, and c_7) into a new set of values (c'_1, c'_5, and c'_7), but their structural information is lost, whereas the clustering method (bottom) forms two groups (G1 and G2), but their temporal order is lost since they contain conformations that overlap in time (c_2, c_3, c_4, c_5).

Figure 10 Overview of the proposed algorithm. Step 1: Divide the trajectory into segments (vertical dashed lines) and reduce each individually. Each segment has k conformations from time t_1 to t_k. Step 2: For each segment, extract characteristic conformations (blue dots) and remove redundant ones (red dots). Step 3: Select the most representative conformations among the previous conformations to obtain a reduced segment.

Figure 11 Parallel processing of segments. The trajectory is divided into m segments (S_1, ..., S_m), and each segment is processed separately and in parallel on a machine with n processing cores. The results of the extraction and selection steps (S'_1, ..., S'_m) are independent of each other and are joined to form the reduced trajectory.

Figure 12 Plots of the TM-score values for the original and reduced trajectories. Protein and simulation details on the left. Trajectories on the right: original (upper part), and reduced (lower part). (A) Trajectory of trp-cage protein, reduced by 98%, from 1044001 to 20881 conformations. (B) Trajectory of SARS CoV2 protein, reduced by 80%, from 100000 to 2000 conformations. (C) Trajectory of the villin-headpiece protein, reduced by 80%, from 15201 to 3011 conformations.

Figure 13 Comparison with other methods. Intrinsic information captured from the original trajectory of the villin-headpiece protein is largely preserved in the reduced trajectories produced by our algorithm. TM-score plot, nMDS pattern and clustering groups are presented at the top, bottom left, and bottom right of each panel. (A) Original trajectory. (B and C) Trajectories reduced by 52% and 80%, respectively. The nMDS produces a set of circular patterns associated with conformations occurring in different times of the simulation, whereas clustering produces a structure of three clusters associated with the first two principal components of the $C_\alpha$atoms of the conformations.

Figure 14 Performance of the algorithm. (A) Runtime comparisons for four folding reduction techniques: nMDS, clustering, PCA, and our proposed algorithm (running with one and two cores). With one core, PCA was the most efficient algorithm, followed by our algorithm. However, when the proposed algorithm was run with two cores, its runtime was reduced by half, being more efficient than PCA. (B) Evaluation of algorithm parallelization. When the number of processing cores is multiplied by two, the processing time reduces by half with a quasi-linear speedup that is maintained for up to ˜8 cores (8x).