Hybrid Spectral/Subspace Clustering of Molecular Dynamics Simulations

Ivan Syzonenko
Center for Computational Science
Middle Tennessee State University
Murfreesboro, Tennessee
is2k@mtmail.mtsu.edu

Joshua L. Phillips
Department of Computer Science
Middle Tennessee State University
Murfreesboro, Tennessee
joshua.phillips@mtsu.edu

ABSTRACT

Data clustering approaches are widely used in many domains including molecular dynamics (MD) simulation. Modern applications of clustering for MD simulation data must be capable of assessing both natively folded and disordered proteins. We compare the performance of the spectral clustering with a more recent subspace clustering approach, and a newly proposed 'hybrid' clustering algorithm which seeks to combine the useful characteristics of both methods on MD data from both protein classes. Results are analysed in terms of accuracy, stability, data density, and other properties. We conclude with what combinations of algorithms/improvements/data density will provide results that are either more accurate or more stable. We find that subspace clustering produces better results than standard spectral clustering, especially for disordered proteins and regardless of input data density or choice of affinity scaling. Additionally, our hybrid approach improves subspace results in most cases and entropic affinity scaling leads to a better performance of both spectral clustering and our hybrid approach.

KEYWORDS

molecular dynamics, spectral clustering, subspace clustering, disordered proteins, entropic affinities, clustering

ACM Reference Format:
Ivan Syzonenko and Joshua L. Phillips. 2018. Hybrid Spectral/Subspace Clustering of Molecular Dynamics Simulations. In *ACM-BCB'18: 9th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, August 29-September 1, 2018, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3233547.3233595

1 INTRODUCTION

Molecular dynamics (MD) simulations are a useful tool for making theoretical predictions for future experimental validation and allows energy landscape exploration for studying meta-stable conformations and the transitions between them [14]. While the problem of capturing meta-stable states may often be successfully resolved within the timescale of the simulation, finding those states is often performed with automated techniques such as clustering. Although there are many clustering algorithms, not all of them can be successfully applied to such high-dimensional models. In particular, recent work in the clustering literature shows that many high-dimensional data sets explore a mixture of independent subspaces and previous clustering studies of MD data have ignored such effects. Here we explore the application of subspace clustering techniques to MD simulation data and compare the performance with traditional spectral clustering (SC) algorithms [12]. By examining multireplicate simulations of two natively-folded proteins (NFP) and two intrinsically disordered proteins (IDP), we hope to show when and why such approaches may be superior to traditional techniques.

ACM-BCB'18, August 29-September 1, 2018, Washington, DC, USA

ACM ISBN 978-1-4503-5794-4/18/08…$15.00
https://doi.org/10.1145/3233547.3233595

2 BACKGROUND

2.1 Clustering Problem

We want to divide a given set of points $X = \{\boldsymbol{x}_j \in \mathbb{R}^d\}_{j=1}^N$ into n groups in such a way that each group contains a subset of points that share similar qualities unique to each particular group. Various approaches to solving this problem have been developed and applied to molecular dynamics simulations [1] [9] [8] [15] [16].

Even though clustering has been a common analysis technique for the field, to our knowledge, recent clustering algorithms such as subspace clustering [4] have not been applied to molecular dynamics simulation data. Subspace methods assume that a mixture of different processes may contribute to an overall data set, and multi-replicate simulations common in the field may exhibit such properties. We explore this possibility in the following sections.

Spectral clustering [18] and Subspace clustering [3, 4] details may be found in supplementary materials https://github.com/fio2003/PYSSC/blob/master/hybrid_pyssc_supplementary_materials.pdf in section 'Extended Methods' while here we will concentrate only on the key features needed for interpetation of methods and results.

2.1.1 Spectral clustering. For representation of relations between points we built similarity graphs that consisted of all vertices connected with a similarity function that encodes all connections. As a similarity function we picked the Gaussian similarity function (GSF) which is one of the most commonly used functions for further refinement of the neighborhood graph and is defined as $s(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma^2)$, where σ is a user-defined parameter which determines the rate of decrease in similarity for all points. Entropic affinities (EA) may be used to select values for σ in a data-driven manner [7, 17]. Similarity graph was built with $k - nearestneighborsgraph$ method. Another vital part of the Spectral clustering method is computation of the approximate normalized cut for which we applied singular value decomposition (SVD) to the graph Laplacian.

2.1.2 Subspace clustering. Key part of the Subspace clustering is a sparse solution of the problem $\boldsymbol{x}_i = X\boldsymbol{c_j}$ described in details in supplementary materials, subsection 'Subspace Clustering' and defined as C. We define S as a a set of m (affine) subspaces $\{S_\ell\}_{\ell=1}^{m}$ in d dimensions.

Please note that Spectral clustering is capable of clustering nonlinear problems, but it was not designed to work with multiple subspaces. On the other hand standard subspace clustering works with affine subspaces but has poor performance on data with nonlinear properties. Both of these statements naturally lead us to an approach which combines the strengths of the two algorithms described above.

2.1.3 Normalized Mutual Information. Since we need some tool to quantitatively measure each algorithm's ability to separate data into clusters, we utilize normalized mutual information (NMI). Full definition can be found in supplementary materials, subsection 'Normalized Mutual Information'. Mutual information reflects the dependence of two variables, in our case - simulation replicate number (R) and cluster number (F).

2.2 Methods

As follows from the descriptions above, the connecting link can be found at the steps just prior to SVD. Among different approaches, we suggest that dot product (SDS) and element-wise (SES) multiplications of both affinity matrices (standard spectral and standard subspace) may resulted in NMI increase. In other words, only weighted connections between points which exist in *both* the standard similarity graph (nonlinear) *and* the subspace-sparse graph (affine subspace) should be preserved. Therefore, a general, efficient algorithm may be defined:

(1) Compute optimization coefficients C.
(2) Compute affinity matrix S.
(3) Construct matrix $M = SC \cdot C$ (SDS) or $M = SC * C$ (SES)
(4) Construct graph Laplacians.
(5) Perform singular vector decomposition.
(6) Run k-means algorithm.

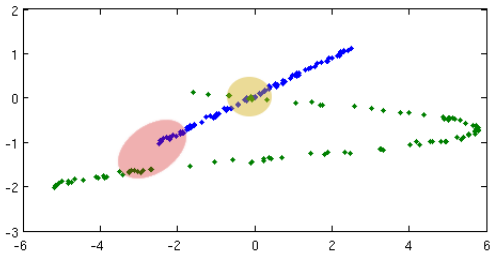

Figure 1: Example of a complex manifold with structures considered challenging for standard clustering algorithms. Red area shows a region hard for subspace algorithms to separate. Yellow area indicates a region hard for spectral clustering to separate.

2.2.1 Geometric Rationale. The standard Gaussian affinity graph will connect points within close proximity to one-another, but exclude those far apart as shown in red (see Figure 1). This property preserves the relationships between points along nonlinear manifolds. The yellow area where the blue and green data sets intersect is however problematic since local connectivity blurs the relationship between different manifolds and will connect the manifolds together. The subspace method treats all blue points and part of the green points (red area) as one subspace since they are observed to lie in the same affine subspace (along the same line in the ambient space). However, the boundary points in the yellow area are more separated between subspaces than when using the Gaussian affinity function. This geometric interpretation suggests that a combination of Gaussian affinities and subspace algorithm coefficients may result in better separation of points in both red and yellow areas. In both approaches, we express all points as vectors of n weights. When both methods agree to connect certain points, such connections result in higher weights in the final connectivity matrix. When they disagree, the connectivity coefficients will be lower and result in breaking unreliable links (spurious connectivity relations created by either the standard spectral or subspace algorithms, independently).

2.3 Implementation

2.3.1 Data Preparation. We studied two general types of proteins: Natively folded proteins (NFP) and Intrinsically disordered proteins (IDP). NFPs fold into stable conformers thus utilizing fewer available degrees of freedom and limiting variation in conformation over time. Proteins for the NFP group were picked in such a way that we have representation of two major structural classes - alpha-helical (Trp-cage [11]) and beta-sheet (GB1 hairpin [6]). Trp-cage and GB1 are well-known and widely used in MD simulation to demonstrate secondary and tertiary structure as well as fast folding. IDPs do not tend to converge to some particular conformer, thus their simulation results in a broad variety of trajectories. NSP1 [5] was picked as an example of relaxed-coil structure which exhibits few meta-stable conformations and Nup116 [5] as an example of compact collapsed-coil structure with many meta-stable conformations [19].

All simulations were performed using GROMACS 4.5.4 [2] using the AMBERff99SBildn [10] forcefield and TIP3P water model with 150 mMol NaCl added to neutralize the system. For each protein we created 10 independent simulations with duration 350ns each, but with different initial velocities. Temperature profile specified in supplementary materials in section 'Temperature profile' and reflects heating each protein into a highly disordered shape and then monitoring its return to the native stable state.

The integration time step was picked as 2 fs. The first 100ns (steps 1-3) were discarded as the equilibration phase. After simulation we extracted the backbone structure from each simulation frame. Prior to clustering the set of coordinate frames was translated into Dihedral angle space and sin-cos embedding of the dihedral angles with the Molecular Dynamics Spectral Clustering Toolkit (MDSCTK) [13].

We created three data sets in order to test how data density would affect our final result: Dense (DN) - frames were taken every

10ps resulting in 2501 points per simulation, then 10 simulations were concatenated to form a complete data set with the size of 25010 points, Sparse (SP) - frames were taken every 100ps resulting in 251 points per simulation, then 10 simulations were concatenated to form a complete data set with size of 2510 points, and Super-sparse (SS) - frames were taken every 1000ps resulting in 25 points per simulation, then 10 simulations were concatenated to form a complete data set with the size of 250 points.

2.3.2 Clustering Setup. K-nearest neighbors were precomputed with MDSCTK and stored for future use. For experiments with plain GSF, sigma values were hand picked to achieve top performance among different proteins, but all results were saved for future analysis. The sigma/perplexity selection strategy used was: the range of values was 'scanned' for the best NMI values and then picked for a finer 'neighborhood' search. Note that not all sigma/perplexity values may be present in all experiments since for each protein different sigma/perplexity values resulted in high NMI. We applied a similar strategy for picking k-nearest neighbors.

For final clustering we ran the k-means++ algorithm 80 times (due to its stochastic nature). Our experience suggests that this was more than enough since the maximum deviation was only 0.042 NMI and the average deviation was only 0.005 NMI. Other implementations of the k-means algorithm may require different number of executions to improve stability or decrease overall computation time. NMI was computed and stored after every iteration in order to derive maximum, minimum, average, and median values for the particular test set. We used only median data for future analysis. Additional results may be found online at: https://github.com/fio2003/PYSSC/tree/master/pyssc_usage_and_raw_results/results_database/results.7z. All code implementation along with more detailed results can be found at: https://github.com/fio2003/PYSSC. Our parallel scheduler which we used for running the analysis can be found at https://github.com/fio2003/PYSSC_scheduler.

2.3.3 Statistical Analysis. Analyses of the clustering results for the above experiments were performed as follows.

Overall performance analysis: We selected the highest NMI values among each group of algorithms, protein types, affinity types, and data densities (categories) and created three Tables (1, 2, and 3) which consist of the intersection of algorithms and proteins.

Graph segmentation analysis: We plotted the relationship between NMI values and perplexity, sigma and k-nearest neighbors for each category to analyze unique properties. Each graph was divided into three segments. Later each segment was classified according to Figures 2 and 3. Two examples of such a classification can be observed in Figures 3 and 4 in supplementary materials.

Segmented graph analysis: We used the previous classification to plot the relationship between the width of graphs described in the previous paragraph and NMI values. Each graph was divided into nine sectors: three for NMI classification and three for thickness classification. For each sector we counted number of segments that fall into each of the sectors to show the relationship between NMI values and variance for each category.

Boxplox analysis: Finally, we used violin plots and boxplots to better describe the distribution of NMI values inside each combination of categories.

All percentages shown are calculated as follows: W + M + N = 100% and "/" + "-" + "\" = 100%. All others (CT, CS, S) show percent of maximum possible value.

	Graph thickness
W	width more than 0.1 NMI.
M	width between 0.05 - 0.1 NMI.
N	width less than 0.05 NMI.
CT	flags that two adjacent segments were classified differently.

Figure 2: Width definitions

	Graph shape behavior
/	grows more than 0.05 NMI per segment
-	does not grow/fall more than 0.05 NMI
\	falls/decreases more than 0.05 per segment.
S	significantly - more than 0.1 NMI per segment.
CS	changes live /\or \/, also called A and V shapes.

Figure 3: Graph shape definitions

3 RESULTS

All tables below contain the median NMI results for the corresponding experiments.

3.1 Overall Performance (Detailed)

For the dense data we used only entropic affinities due to the prohibitive computational cost of exploring the parameter space for fixed sigma.

Entropic affinity				
TRP Cage	GB1	NUP116	NSP1	
0.4495	0.5053	0.7447	0.5114	SC
0.2848	0.2772	0.4157	0.3128	Subspace
0.3892	0.3991	0.5869	0.4148	SDS
0.3752	0.4695	0.5794	0.5052	SES

Table 1: Best NMI values for each protein obtained with all algorithms using entropic affinities and the dense data set.

3.1.1 Entropic Affinities Analysis. Discussion below is in reference to Table 1. Algorithms: For all cases the SC algorithm showed the highest NMI values while the Subspace algorithm showed the lowest results.

Proteins: NFPs (0.4495 for TRP Cage and 0.5053 for GB1) and NSP1 (0.5114) demonstrated similar results while NUP116 demonstrated a significantly higher NMI value (0.7447).

Entropic affinity

TRP Cage	GB1	NUP116	NSP1	
0.4593	0.5048	0.7311	0.5545	SC
0.4740	0.3665	0.6345	0.5182	Subspace
0.4924	0.4662	0.7169	0.5432	SDS
0.5018	0.4901	0.7214	0.5890	SES

Plain affinity

TRP Cage	GB1	NUP116	NSP1	
0.3100	0.2485	0.2864	0.3037	SC
0.4740	0.3665	0.6345	0.5182	Subspace
0.4881	0.4319	0.6360	0.5350	SDS
0.2685	0.3047	0.3395	0.3269	SES

Table 2: Best NMI values for each protein obtained with all algorithms for the sparse data set.

3.1.2 Entropic Affinities Analysis. Discussion below is in reference to Table 2. Algorithms: SES demonstrated high NMI values for all proteins, but SC was slightly better for GB1 and NUP116. Subspace performed the worst among algorithms for IDPs and GB1.

Proteins: TRP Cage and GB1 had almost identical NMI values - 0.518 for TRP Cage and 0.5048 for GB1. NSP1 had slightly higher NMI - 0.589 than both NFPs. NUP116 had the highest NMI among all proteins - 0.7311.

3.1.3 Plain Affinities Analysis. Discussion below is in reference to Table 2. Algorithms: SDS showed the highest NMI among algorithms for all proteins while SC had the lowest NMI values for GB1 and IDPs. SES showed the lowest NMI value for TRP Cage. Proteins: NFPs had lower NMI values, while IDPs had higher NMI values.

Entropic affinity

TRP Cage	GB1	NUP116	NSP1	
0.4508	0.4592	0.5941	0.4797	SC
0.4150	0.3861	0.6586	0.4685	Subspace
0.4170	0.4017	0.6727	0.4624	SDS
0.4224	0.4095	0.6509	0.5128	SES

Plain affinity

TRP Cage	GB1	NUP116	NSP1	
0.3496	0.3181	0.2989	0.1575	SC
0.4150	0.3861	0.6586	0.4685	Subspace
0.4273	0.4159	0.6759	0.4685	SDS
0.3487	0.3490	0.3210	0.3016	SES

Table 3: Best NMI values for each protein obtained with all algorithms for the super-sparse data set.

3.1.4 Entropic Affinities Analysis. Discussion below is in reference to Table 3. Algorithms: SC performed the best for NFP group with NMI values of 0.4508 and 0.4592 for TRP Cage and GB1 respectively, but demonstrated the worst NMI value of 0.5941 for NUP116. Subspace demonstrated the lowest NMI values for the NFP group resulting in 0.4150 and 0.3861 for TRP Cage and GB1 respectively. SDS demonstrated the highest NMI value for NUP116 - 0.6727, but the lowest NMI value for NSP1. SES demonstrated the highest value for NSP1 - 0.5128. Proteins: Both NFPs showed similar values and within the IDP group, NUP116 had the highest NMI value - 0.6727.

3.1.5 Plain Affinities Analysis. Discussion below is in reference to Table 3. Algorithms: SC demonstrated the worst NMI results for the IDPs and GB1. SDS demonstrated the highest NMI values for all proteins. Subspace demonstrated the same (highest) NMI value for NSP1. SES demonstrated the worst NMI values for TRP Cage protein. Proteins: Like in the Sparse data case, the NFP group demonstrated similar results and in the IDP group, NUP116 had the highest value - 0.6759.

3.2 Overall Performance (General)

Based on the detailed results above, it is clear that SES depends more on SC while SDS depends more on Subspace. Entropic affinities generally demonstrated better NMI values for all algorithms with one exception: the NMI value obtained for NUP116 in the super-sparse data with SDS and plain affinities was not significantly higher (0.6759) than the same combination with entropic affinities (0.6727). SC combined with entropic affinities is the best for all proteins in the dense data, GB1 and NUP116 for the sparse data, and the NFP group for the super-sparse data. SES demonstrated the best results for TRP Cage and for the sparse data and NSP1 for the sparse and super-sparse data sets. For plain affinities SDS with sparse and super-sparse data showed the best results among all algorithms for all proteins. In general also, SC was almost always the worst algorithm to use with plain affinities.

3.3 General Graph Segmentation Results

3.3.1 Sparsity. Graph thickness: *For knn*, the sparser the data, the more narrow graphs are produced. *For perplexity/sigma* we see the opposite trend. Both show more CT with denser data. Graph shape: *For knn*, all results were pretty much identical, but the denser data contained more CS. Angles were also smaller. *For perplexity/sigma*, denser data contained slightly fewer straight parts.

3.3.2 Algorithms. Graph thickness: *For knn*, SDS and SES showed thinner shapes than SC. *For perplexity/sigma* SC showed thinner shapes, SDS second, and SES was the last in this regard, but SES thickness variation was less (30% for SES and 45% for SC). Graph shape: *For knn*, SC had the most (83%) straight lines, while SES had the least number (43%). There was the opposite situation with curvature, where SC had small curvature and SES had sharp curvature. *For perplexity/sigma*, there were no significant differences except that SC had the highest number of CS, but SES had the smallest.

3.3.3 Affinity. Graph thickness: *For knn*, entropic affinities exhibited twice as many narrow parts compared to plain affinities and a very similar situation with regard to changes, so entropic affinities were more stable. *For perplexity/sigma* the situation was the same, 68% vs 7% narrow parts for entropic and plain affinities, but 56% vs 17% for changes. The situation with changes can be explained since often there was just an even distribution of points that did not give any information, but was not treated as a thickness change. Graph shape: *For knn*, entropic affinities produced more straight lines, less CS and significantly less curvature than plain

affinities. *For perplexity/sigma* entropic affinities contained slightly fewer straight regions.

3.3.4 Protein type. Graph thickness: *For knn*, both were similar, but NFP produced more narrow pieces and significantly less changes. *For perplexity/sigma*, big difference was only with changes 33% vs 50% for for NFP and IDP. NFP had slightly more narrow parts. Graph shape: *For knn*, NFP contained more straight regions and less CS. For perplexity/sigma NFP still contained a little more straight regions and less A and V shapes.

More detailed analysis of graph segmentation can be found in the supplementary materials.

3.4 Segmented Graph Analysis

Graphs displayed in Figures 4, 5, 6, 7, 8 have only categorical meaning. The points inside sectors were slightly (randomly) displaced along the x-axis to provide perspective concerning the number of points with similar NMI values.

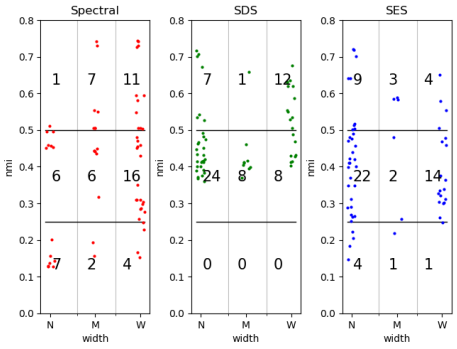

Figure 4: Relationship between NMI values and variation for the SC (left), SDS (middle), and SES (right) algorithms for the k-nearest neighbors batch. Numbers on the graph indicate the number of points in that sector.

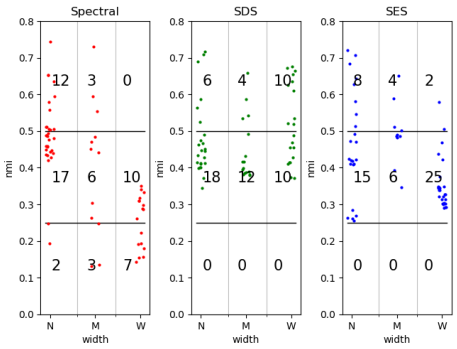

Figure 5: Relationship between NMI values and variation for the SC (left), SDS (middle), and SES (right) algorithms for the perplexity/sigma batch. Numbers on the graph indicate the number of points in that sector.

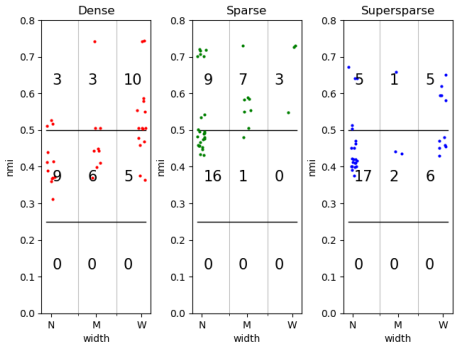

Figure 6: Relationship between NMI values and variation for the Dense (left), Sparse (middle), and Super-sparse (right) data sets. Numbers on the graph indicate the number of points in that sector.

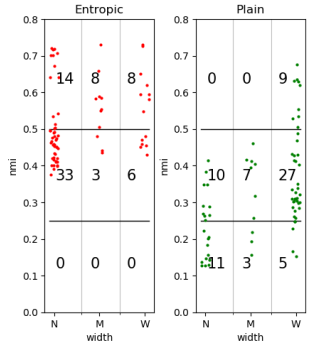

Figure 7: Relationship between NMI values and variation for Entropic (left) and Plain (right) affinities. Numbers on the graph indicate the number of points in that sector.

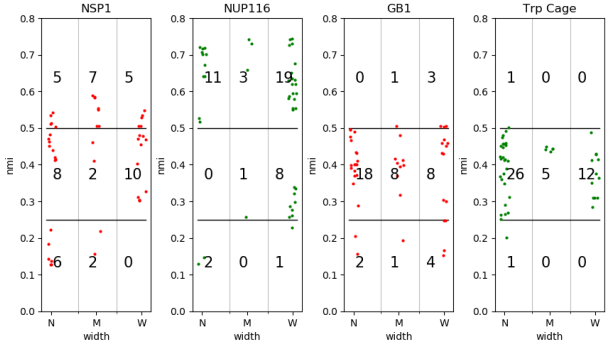

Figure 8: Relationship between NMI values and variation for IDPs: NSP1, NUP116 (left) and NFPs: GB1, TRP Cage (right). Numbers on the graph indicate the number of points in that sector.

3.4.1 Algorithm. We found that despite differences in behavior between the k-nearest neighbors and perplexity/sigma batches, we still can see a clear difference between algorithms inside each

group. In the k-nearest neighbors case (Figure 4) SDS produced generally higher NMI values than SC and SES produced more consistent results, most of which were described as 'narrow'. In the perplexity/sigma batch we may observe that both SDS and SES demonstrate generally higher NMI than SC, but also have more variation which can be observed in figure 5.

3.4.2 Density. Both sparse and super-sparse plots showed similar behavior for both k-nearest neighbors and perplexity/sigma. Denser data produced slightly higher NMI, which can be seen on Figure 6.

3.4.3 Affinity. We found that there is a clear difference between plain and entropic affinities. Entropic affinities demonstrate more narrow variation and higher NMI values than plain affinities (see Figure 7). Graphs for perplexity/sigma (data not reported here, but can be found inside the git repository mentioned above) showed even stronger difference, describing most points with plain affinities as wide and most points with entropic affinities as narrow.

3.4.4 Proteins. Results for the k-nearest neighbors and perplexity/sigma batches were consistent and showed clear differences between IDP and NFP groups. The NFP group contained moderate NMI values while the IDP group contained higher NMI values. Inside the NFP group both proteins show similar behavior, while inside the IDP group NUP116 demonstrated significantly higher NMI values than NSP1 which can be found in Figure 8. Consistency inside the NFP group was expected since they have very similar structure and tend to fold fast producing similar trajectories during each simulation. Inconsistency inside the IDP group can also be explained when we look closer at their known physical properties: NUP116 tends to have many semi-folded shapes resulting in different trajectories that are easier to separate. On the other hand, NSP1 tends to have many large-amplitude movements and no particular semi-folded shapes which results in producing more chaotic trajectories that are harder to separate.

4 DISCUSSION

We have performed a thorough analysis of the clustering results produced by SC, Subspace, SES, and SDS and their EA improvements on variable-density MD simulation data. The results section shows that EAs significantly improve clustering quality and should be used instead of plain affinities for all algorithms. Hybrid solutions such as SES and SDS in most cases either improve clustering accuracy or stability of the clustering results.

We found that increasing data density significantly increases clustering time, but did not always produce better clustering accuracy. Since the entropic affinities approach is not necessarily the standard approach used in the field, our results indicate that the subspace clustering algorithm and both SDS and SES produced higher (55% more) NMI values than SC. Therefore, our approach of reusing results of the convex optimization solution is a geometrically well-motivated method for dealing with data displaying both subspace and nonlinear components. However, the EA results attest to the fact that much of the issue with clustering MD simulation data is due to nonuniform sampling. Additionally, it was clear that IDPs were easier to cluster than NFPs which was not surprising due to lack of simulation convergence. This result only bolsters the need for better clustering approaches such as SDS and SES. Although we concentrated on MD simulations, SDS and SES improvements should be similar for other data with similar properties. This can lead to better clustering results in areas that intensively use clustering techniques, such as text recognition, image processing, data science, etc. Although higher k generally resulted in higher NMI values, it also required more computational time. Analysis of minimum NMI results may be of interest since algorithmic stability may be more important for computationally demanding data sets.

REFERENCES

[1] Christoph Best and H-C Hege. 2002. Visualizing and identifying conformational ensembles in molecular dynamics trajectories. *Computing in Science & Engineering* 4, 3 (2002), 68–75.

[2] Robert B Best, Xiao Zhu, Jihyun Shim, Pedro EM Lopes, Jeetain Mittal, Michael Feig, and Alexander D MacKerell Jr. 2012. Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain $\chi 1$ and $\chi 2$ dihedral angles. *Journal of chemical theory and computation* 8, 9 (2012), 3257–3273.

[3] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, New York, NY, USA.

[4] Ehsan Elhamifar and René Vidal. 2012. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *CoRR* abs/1203.1005 (2012). http://arxiv.org/abs/1203.1005

[5] André Goffeau, Bart G Barrell, Howard Bussey, RW Davis, Bernard Dujon, Heinz Feldmann, Francis Galibert, JD Hoheisel, Cr Jacq, Michael Johnston, et al. 1996. Life with 6000 genes. *Science* 274, 5287 (1996), 546–567.

[6] Angela M Gronenborn, David R Filpula, Nina Z Essig, Aniruddha Achari, Marc Whitlow, Paul T Wingfield, and G Marius Clore. 1991. A Novel, Highly Stable Fold of the Immunoglobulin Binding Domain of Streptococcal Protein G. *Science* 253, 5020 (1991), 657–661.

[7] Geoffrey E Hinton and Sam T Roweis. 2003. Stochastic neighbor embedding. In *Advances in neural information processing systems*. 857–864.

[8] Rao Huang, Li-Ta Lo, Yuhua Wen, Arthur F Voter, and Danny Perez. 2017. Cluster analysis of accelerated molecular dynamics simulations: A case study of the decahedron to icosahedron transition in Pt nanoparticles. *The Journal of chemical physics* 147, 15 (2017), 152717.

[9] Mary E Karpen, Douglas J Tobias, and Charles L Brooks III. 1993. Statistical clustering techniques for the analysis of long molecular dynamics trajectories: analysis of 2.2-ns trajectories of YPGDV. *Biochemistry* 32, 2 (1993), 412–420.

[10] Kresten Lindorff-Larsen, Stefano Piana, Kim Palmo, Paul Maragakis, John L Klepeis, Ron O Dror, and David E Shaw. 2010. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Structure, Function, and Bioinformatics* 78, 8 (2010), 1950–1958.

[11] Jonathan W Neidigh, R Matthew Fesinmeyer, and Niels H Andersen. 2002. Designing a 20-residue Protein. *Nature Structural and Molecular Biology* 9, 6 (2002), 425.

[12] Andrew Y Ng, Michael I Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*. 849–856.

[13] Joshua L. Phillips, Edmond Y. Lau, Michael E. Colvin, and Shawn Newsam. 2008. *Analyzing dynamical simulations of intrinsically disordered proteins using spectral clustering*. 17–24. https://doi.org/10.1109/BIBMW.2008.4686204

[14] Joshua Lee Phillips. 2012. Validation of computational approaches for studying disordered and unfolded protein dynamics using polymer models. (2012).

[15] Joshua L Phillips, Michael E Colvin, and Shawn Newsam. 2011. Validating clustering of molecular dynamics simulations using polymer models. *BMC bioinformatics* 12, 1 (2011), 445.

[16] Sarah Rauscher and Régis Pomès. 2010. Molecular simulations of protein disorder. *Biochemistry and cell biology* 88, 2 (2010), 269–290.

[17] Max Vladymyrov and Miguel Carreira-perpinan. 2013. Entropic Affinities: Properties and Efficient Numerical Computation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Sanjoy Dasgupta and David Mcallester (Eds.), Vol. 28. JMLR Workshop and Conference Proceedings, 477–485. http://jmlr.org/proceedings/papers/v28/vladymyrov13.pdf

[18] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[19] Justin Yamada, Joshua L. Phillips, Samir Patel, Gabriel A Goldfien, Alison Calestagne-Morelli, Hans Huang, R. de la Reza, Justin F. Acheson, Viswanathan Krishnan, Shawn D. Newsam, Ajay Gopinathan, Edmond Y. Lau, Michael E. Colvin, Vladimir N. Uversky, and Michael F. Rexach. 2010. A bimodal distribution of two distinct categories of intrinsically disordered structures with separate functions in FG nucleoporins. *Molecula* 9 10 (2010), 2205–24.