

VISUALIZING AND IDENTIFYING CONFORMATIONAL ENSEMBLES IN MOLECULAR DYNAMICS TRAJECTORIES

Using a conformational distance measure based on the changes in intramolecular atom distances, the authors show that a planar map can help visualize molecular dynamics trajectories efficiently such that conformational ensembles appear as well-separated point sets. Cluster analysis can then be used to automatically identify conformations.

Molecular dynamics simulations on large computers have become one of the mainstays for investigating the structure–function relationship in biomolecules. Using statistical algorithms, we get several snapshots (or a *trajectory*) of the molecule. The distribution of the snapshots approximates the expected distribution of molecular shapes in actual biochemical processes. By looking for conformational ensembles—groups of configurations that have a similar geometrical shape—and transition paths between them, biochemists can learn about the molecular bases of biochemical processes. Such an understanding has many practical applications—for example, in designing more efficient medical drugs.

Identifying typical shapes in such a large set of molecular configurations is itself a difficult task, and much literature on the topic exists.^{1–3} In

most simulations, researchers focus on a few characteristic quantities, such as some helical angles in the molecule, and monitor the change of these quantities in the simulation. This approach requires some advance knowledge about the parts of the molecule that are important to the dynamics. From the computational science viewpoint, however, it's more interesting to investigate methods that don't use such a priori knowledge and rely solely on the information inherent in the trajectory. Such an approach not only opens the possibility of a completely automatic analysis but also serves as an example of how to use techniques from information visualization and statistical cluster analysis in molecular simulations.

Our research is part of a larger effort to identify essential degrees of freedom in molecular simulations.⁴ Based on a feature vector that captures the geometrical shape of each molecular configuration in the trajectory, we introduce a measure of conformational distance and use it to visualize the trajectory in a plane and identify clusters of similar configurations.

In this article, we apply our method to a trajectory from an adaptive-temperature hybrid Monte Carlo simulation⁵ of the molecule *adenylyl(3'-5')cytidyl(3'-5')cytidin*—also called

1521-9615/02/\$17.00 © 2002 IEEE

CHRISTOPH BEST

Massachusetts Institute of Technology

HANS-CHRISTIAN HEGE

Zuse Institute, Berlin

r(ACC)—in a vacuum using the GROMOS96 molecular dynamics code extended atom force field.^{4,5} This is a simple tri-ribonucleotide consisting of three residues and 70 (effective) atoms, but it is sufficiently complex to demonstrate our procedure. For the analysis, we chose a subset of 1,000 configurations equidistantly from the trajectory.

Configurations, feature vectors, and configurational distance

Simulating complex biomolecular processes using a molecular dynamics simulation produces a trajectory—a sequence of molecular configurations that models the actual dynamics of the molecules involved in the process. Typically, a trajectory contains tens of thousands of configurations. Each configuration x is described by N vectors $\mathbf{x}_i \in \mathbb{R}^3$, $i = 1, \dots, N$, which specify the 3D Cartesian position of the N atoms that make up a molecule.

To classify the geometry of a trajectory's configurations, we must assign a *feature vector* to each configuration that describes its geometry in such a way that similar configurations have similar feature vectors. However, the set of $3N$ numbers that comprise the atoms' Cartesian positions is unsuitable to this end because identical geometries can appear with different rotations and translations. Although we can easily fix the translational freedom using the center of mass, eliminating the rotational degree of freedom is more difficult. One way to fix it is to rotate the molecule into a preferred position by optimizing the rotation with respect to a reference configuration or by fixing the axis of inertia. Many applications have frequently and successfully used this method, but such fitting procedures could suddenly produce a large rotation when the fit is ambiguous and thus do not guarantee that similar configurations will always have similar feature vectors.

Alternatively, we can choose a feature vector that is invariant under translations and rotations by considering the set of all intramolecular distances⁶

$$\{d_{ij}(x) = |\mathbf{x}_i - \mathbf{x}_j|, \quad i, j \in 1, \dots, N\}. \quad (1)$$

The price to pay is that instead of $3N$ elements, this vector now has $N(N-1)/2$. However, geometrically similar configurations have similar feature vectors, and the Cartesian distance in $N(N-1)/2$ -dimensional space is a natural mea-

sure of conformational distance of two configurations x and y :

$$D(x, y) = \sqrt{\frac{1}{N(N-1)/2} \sum_{i>j} (d_{ij}(x) - d_{ij}(y))^2}. \quad (2)$$

The feature vector's $N(N-1)/2$ components are not independent and will thus lie on a $(3N-6)$ -dimensional submanifold of this space. However, reconstructing the original coordinates from the pairwise distances, known as the *graph realization* or *molecule problem* in computer graphics, is an NP-hard problem.

Another common way to choose a feature vector is to use the dihedral angles between certain atoms as basic degrees of freedom. This is a natural choice because dihedral angles are the main degrees of motion in simulations (atomic distances and bond angles are usually much more rigid). But, the potential energy that determines the molecule's dynamics depends on the spatial distance of the atoms; the relation between dihedral angles and spatial distances is complicated at best. In particular, small changes in a pivotal angle could produce a large change of geometry.

Intramolecular distances correctly capture the motion inside a molecule. If a group of atoms in the molecule is rigid, that group will always be characterized by the same set of intramolecular distances between these atoms, regardless of any change in the group's location, even though the atoms' Cartesian positions change. This reinforces the importance of choosing a feature vector that represents all the molecule's possible geometrical variations.

The feature vector in Equation 1 is vast compared to the number of degrees of freedom. In our example molecule, it has 2,415 elements as compared to $70 \times 3 - 6 = 204$ degrees of freedom. Some of its elements will show little or no fluctuation (specifically the ones associated with the lengths of chemical bonds) and thus contribute little to the conformational distance, others will fluctuate thermally, and still others will assume different values in different fluctuations and thus exhibit a double-peaked distribution.

Here, we are only interested in conformational distance (see Equation 2) between feature vectors, not in the feature vectors themselves. To reduce the computational requirements, we analyze the feature vector's elements statistically over a large set of configurations and select those whose distribution has the largest width and thus the largest contribution to the conformational

distance. Because thermal fluctuations are smaller than conformational changes, this also includes the distances most affected by conformational changes. The conformational distance obtained from a subset of 500 intramolecular distances is a good approximation to Equation 2 and simultaneously reduces thermal noise. Researchers have used a similar procedure to identify essential degrees of freedom in Cartesian coordinate space.²

Mapping conformational distance

Using the conformational distance defined by Equation 2, we can now characterize the trajectory ($x^{(n)}$, $n = 1 \dots N$) by a matrix $D_{nm} = D(x^{(n)}, x^{(m)})$ of conformational distances between the configurations in it. To capture the trajectory's major properties, we try to represent these distances approximately in a low-dimensional space. This means we assign each configuration a point in d dimensions such that the geometrical similarity between configurations is reproduced as faithfully as possible by the distance between the representative points. For $d = 2$, this produces a 2D map of the trajectory that captures the geometrical similarity and dissimilarity of many configurations. Visualizing an arbitrary similarity matrix between a set of N objects is a general problem, and the method we present here thus applies to other problems as well.

Obviously, different choices exist of how to reduce the information from the $N(N - 1)/2$ dimensions in which the feature vector lives to d dimensions. Assume that the n th configuration is assigned a representative $\mathbf{a} \in \mathbb{R}^d$ and that the configurational distance between the n th and m th configuration is D_{nm} . One choice is to require that the mean quadratic deviation of D_{nm} from the d -dimensional distance between the representatives a_n and a_m

$$A^2 = \sum_{n>m} (|\mathbf{a}_n - \mathbf{a}_m| - D_{nm})^2 \quad (3)$$

is minimized, so

$$\frac{\partial A^2}{\partial \mathbf{x}_n} = \sum_m \frac{\mathbf{a}_n - \mathbf{a}_m}{|\mathbf{a}_n - \mathbf{a}_m|} (|\mathbf{a}_n - \mathbf{a}_m| - D_{nm}) = 0 \quad (4)$$

vanishes. You can visualize this equation as a set of springs that connect all pairs of points and whose natural length is given by the desired distance between the points. Thus the force exerted on point n from point m is proportional to the

difference between actual and desired distance and directed along the vector $\mathbf{x}_n - \mathbf{x}_m$. Equation 4 expresses a balance of forces at point n .

We can minimize Equation 3 numerically by using the conjugate–gradient method. However, there is no guarantee of finding the absolute minimum with this method.

It is instructive to see how the positions of a change as the dimension d of the visualization space increases. Suppose we have a solution of Equation 4 in $d - 1$ dimensions. We can embed this solution in d dimensions by setting $x_{n,D} = 0$ for all n , which still satisfies Equation 4, and is thus an example of a false minimum. However, if we change $x_{n,D}$ infinitesimally, the resulting change of A^2 is given by the second derivative

$$\frac{\partial^2 A^2}{\partial x_{n,D} \partial x_{n,D}} = - \sum_{k \neq n} \frac{d_{nk} - |\mathbf{x}_n - \mathbf{x}_k|}{|\mathbf{a}_n - \mathbf{a}_k|} \quad (5)$$

If this quantity is negative for any n , we can reduce A^2 by moving a_n out of the $(d - 1)$ -dimensional subspace. This happens when the desired distances d_{nk} are, on a distance-weighted average, larger than the distances $|\mathbf{a}_n - \mathbf{a}_k|$ of the representatives—for example, if a_n is “pushed” out of the $(d - 1)$ -dimensional subspace by its neighbors.

Let's contrast this low-dimensional representation to another widely used one, namely principal-component analysis based on the feature matrix's singular-value decomposition (SVD). Let M_{ni} be a feature matrix of $n = 1, \dots, N$ objects, with $i = 1, \dots, N_f$ features each. (In our example, N is the number of configurations whereas N_f is the number of intramolecular distances.) The SVD expresses this matrix as a series

$$M_{ni} = \sum_k^{\text{rank } M} \lambda_k \mathbf{u}_n^{(k)} \mathbf{v}_i^{(k)} \quad (6)$$

where $\mathbf{u}^{(k)}$ and $\mathbf{v}^{(k)}$ are N - and N_f -dimensional orthonormalized basis vectors, and λ_k are the singular values. The number of terms in the series is the matrix's rank—at most, the lower of n and m .

We can find a d -dimensional approximation \tilde{M} to M by truncating the series after d terms, including only the largest λ_k . This approximation defines d -dimensional representatives \mathbf{a}_n of the objects using the values of the first d basis vectors

$$\mathbf{a}_n = \left(\lambda_k \mathbf{u}_n^{(k)} \right)_{k=1, \dots, d} \quad (7)$$

because the distances \tilde{D}_{nm} between two rows of \tilde{M} are exactly reproduced by these representatives:

$$\begin{aligned}
\tilde{D}_{mm}^2 &= \sum_i^{N_f} (\tilde{M}_{mi} - \tilde{M}_{mi})^2 = \sum_i^{N_f} \left(\sum_k^d \lambda_k (u_n^{(k)} - u_m^{(k)}) v_k^{(i)} \right)^2 \\
&= \sum_{kl}^d \lambda_k \lambda_l (u_n^{(k)} - u_m^{(k)}) (u_n^{(l)} - u_m^{(l)}) \sum_i^{N_f} v_i^{(k)} v_i^{(l)} \\
&= \sum_k^d \lambda_k^2 (u_n^{(k)} - u_m^{(k)})^2 = (\mathbf{a}_n - \mathbf{a}_m)^2
\end{aligned}
\tag{8}$$

(where we used the fact that the vectors $\mathbf{v}^{(k)}$ are orthogonal). However, the SVD and thus the basis vectors $u_i^{(k)}$ do not depend on d : The representatives in the SVD approach are found by orthogonal projection to a d -dimensional subspace. Conversely, in the approximation obtained from minimizing Equation 3, the nonlinearity that the square root introduces also redistributes some of the “lost” distance in the remaining dimensions. We can thus argue that the representatives obtained from Equation 3 should give a better representation of the feature vector than those obtained from SVD.

Figure 1 shows a 2D map of 1,000 configurations of $r(\text{ACC})$ we found by minimizing Equation 3. The computational requirements here are dictated by the need to store the (symmetric) $1,000 \times 1,000$ distance matrix; storing each molecule’s feature vector is not required. Finding the minimum of Equation 3 thus takes a matter of minutes on a typical workstation.

A line connects the points in the same se-

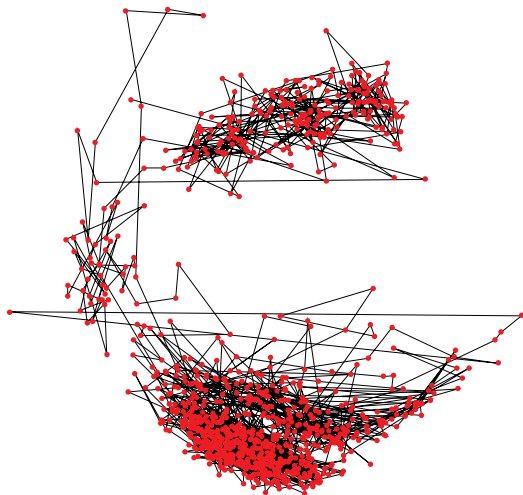


Figure 1. A two-dimensional map of 1,000 configurations chosen from a molecular dynamics trajectory. Points that appear close in this plane represent configurations with a similar geometry.

quence as they are generated in the simulation. This information does not enter in placing the points in the plane, so the fact that the line segments generally are short indicates that the algorithm recognizes adjacent points in the trajectory as geometrically similar. The one pair of lines that crosses nearly the whole plane horizontally is a transient effect from the first three configurations before the molecule equilibrated.

We can perceive at least three clearly separated groups of points. They constitute conformations in a geometrical sense—for example, sets of configurations that have similar geometrical properties. That they are also dynamic conformations is visible from the fact that the connecting line of the points only rarely crosses from one subset to another. This confirms that the 2D representatives the algorithm chose correctly represent the system’s dynamics.

Figure 2 shows a similar map that plots the distance of the centers of two of the residues for each configuration. Because the system consists of only three residues, we can assume that most of the conformational dynamics are in the position of their centers. The picture is similar to Figure 1 in that approximately three distinct point groups appear, but the separation of the groups is less clear than in Figure 1. This indicates that the conformational dynamics is not simply the motion of the residues’ centers, but also includes smaller rearrangements correlated to the large-scale motions. By considering several intramolecular distances, all those small rearrangements enter and reinforce the separation of conformations in the map.

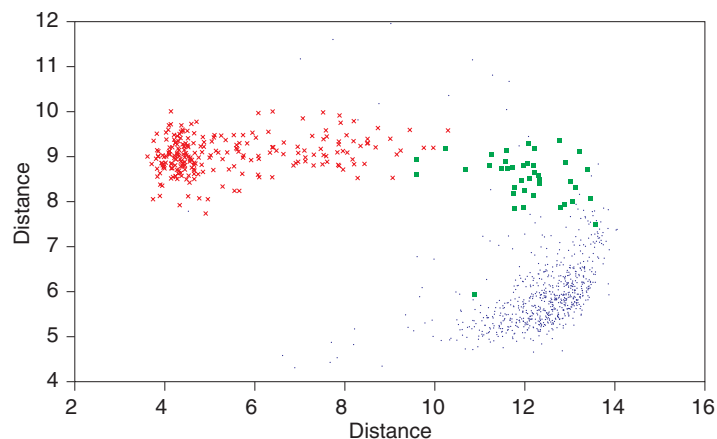


Figure 2. A map of the trajectory in the plane spanned by the distance between the centers of two residues in the molecule. The different symbols represent configurations from the three clusters in Figure 1.

Cluster analysis

Cluster analysis is a statistical method to partition a set of objects into disjoint subsets (clusters) with the property that the objects in a cluster are in some sense more similar to each other than to the remaining objects. There are several different ways to make this statement mathematically precise. Here, we choose the notion of minimum residual similarity between clusters, which leads to a natural formulation of the problem in terms of eigensystem analysis and to a heuristic algorithm for its solution. This spectral method goes back to works by Wilm Donath and Alan Hoffmann⁷ on graph partitioning in computer logic and Miroslav Fiedler⁸ on acyclic graphs, and was later picked up by Bruce Hendrickson.⁹ Researchers have also applied other cluster analysis methods based on neural networks or fuzzy clustering to molecular dynamics simulations.^{10–12}

To be as flexible as possible, assume that a similarity measure

$$0 \leq A_{nm} \leq 1, \quad n, m = 1, \dots, N \quad (9)$$

is given between N objects, where $A_{nm} = 0$ indicates complete dissimilarity and $A_{nm} = 1$ complete identity of objects n and m . The residual similarity $R(C)$ of a cluster $C \subset \{1, \dots, N\}$ characterizes the similarity of cluster elements to elements outside the cluster and can be expressed by summing the similarity of all objects in C to all objects not in C :

$$R(C) = \sum_{n \in C, m \notin C} A_{nm}. \quad (10)$$

The elementary step in cluster analysis is to partition the set into two subsets such that this quantity is minimized. Let a_i be this partition's characteristic vector, meaning $a_i = 1$ if $i \in C$, and $a_i = -1$ otherwise. Then we can rewrite the residual similarity as an expectation value

$$\begin{aligned} R(C) &= \frac{1}{4} \sum_{nm} (a_n - a_m)^2 A_{nm} \\ &= \frac{1}{2} \sum_{nm} a_n \left(\sum_k A_{nk} \delta_{nm} - A_{nm} \right) a_j \\ &= (a, Ma) \end{aligned} \quad (11)$$

with the matrix

$$M_{nm} = \begin{cases} -A_{nm} & \text{if } n \neq m \\ \sum_k A_{nk} & \text{if } n = m \end{cases}. \quad (12)$$

This matrix is a generalized Laplacian operator

on a graph with sites n and vertices A_{nm} ; if A_{nm} were the next-neighbor connectivity matrix of a square lattice ($A_{nm} = 1$ if points n and m are next neighbors), it would reduce to the ordinary discrete Laplacian operator.

Minimizing $R(C)$ now resembles the linear algebra problem of finding the (normalized) vector a that minimizes (a, Ma) —basically, finding the lowest eigenvector. However, our a is restricted to values ± 1 , and finding the minimum of (a, Ma) over such vectors is a hard combinatorial problem. Instead of tackling this problem directly, we use a heuristic based on the eigenvalue problem: we assume that we can infer a good (but not optimal) solution of the hard combinatorial problem from the easier linear algebra problem.

The lowest eigenvector of M is the constant eigenvector $a_n \equiv 1$ with eigenvalue 0. The second-lowest eigenvector is normalized, orthogonal to the constant vector, and thus satisfies

$$\sum_n a_n^2 = N \quad \text{and} \quad \sum_n a_n = 0, \quad (13)$$

which guarantees that it will contain both positive and negative values. In graph theory, this eigenvector is called the *characteristic valuation* of a graph. In particular, we can show that if the graph described by the connectivity matrix A_{nm} consists of two unconnected subgraphs, the characteristic valuation will be $a_n = +1$ for nodes n on one subgraph and $a_n = -1$ for nodes on the other.

In our setting, clusters will not be completely separated, so a_n will assume values between -1 and $+1$. To perform the cluster analysis, we map the continuous value a_i to discrete value \tilde{a}_i using a threshold l :

$$\tilde{a}_n = \begin{cases} -1 & \text{if } a_n \leq l \\ +1 & \text{if } a_n > l \end{cases}. \quad (14)$$

The threshold is a free parameter, and we can determine it by minimizing the residual similarity over all possible values l . In this way, the minimization problem is reduced from $N!$ to just N options (the N values a_n), and the characteristic valuation serves as a heuristic to determine the options that we're considering.

The measure of residual similarity generally favors splitting off a single point from the cluster because Equation 11 contains only $N - 1$ terms in this case, as opposed to $N^2/4$ when splitting symmetrically. This automatically introduces a quality control in the splits because central splits

occur only when the cluster separation is rather favorable, but it might also hinder the analysis of noisy data. However, we only chose the special form in Equation 11 to turn the problem into an eigenvalue problem. Because the whole procedure is heuristic, we could use a different similarity measure when determining the splitting threshold—a measure that includes a combinatorial factor

$$R(C) = \frac{1}{|C|(N-|C|)} \sum_{n \in C, m \notin C} A_{nm}. \quad (15)$$

The correct measure depends mainly on the application. The original measure is stricter in what it returns as a cluster, whereas the latter measure favors balanced splittings. In some problems, such as partitioning matrices for processing in a parallel computer, we could even demand each split to be symmetrical.

After partitioning the data set into two subsets, we recursively reapply the algorithm to these subsets and obtain a splitting tree that terminates only when the cluster size is two. To identify good clusters in the splitting tree, we found it useful to consider the cluster's average width (in the feature vector space) relative to that of its parent cluster. This quantifies how many points in the subcluster are, on average, closer to each other than in the original cluster and thus how much the split improves the cluster criterion.

Consider, for example, the situation in which there are three clusters. The first split will produce one correctly identified cluster and a second pseudocluster that encompasses the other two, but the true cluster's relative width is much smaller than the pseudocluster's. Only after the next split will we see that the latter consists of two clusters. Typical values for this quantity range between 0.5 and 0.8.

Applying the cluster algorithm to our trajectory's similarity matrix, we see that the first few splits remove 22 isolated points before the small cluster in Figure 3a appears with 40 points and a relative width of 0.46 (both compared to its immediate predecessor and to the initial point set). The remaining points split some steps further into a cluster with 698 points in Figure 3b and another cluster with 230 points in Figure 3c, with relative widths of about 0.53. After some more steps, the larger subcluster breaks into three subclusters with 388, 52, and 21 points, respectively, and relative widths of 0.91, 0.73, and 0.61, as Figure 3d, 3e, and 3f show. Similarly, the smaller subcluster also separates into three weak subclusters.

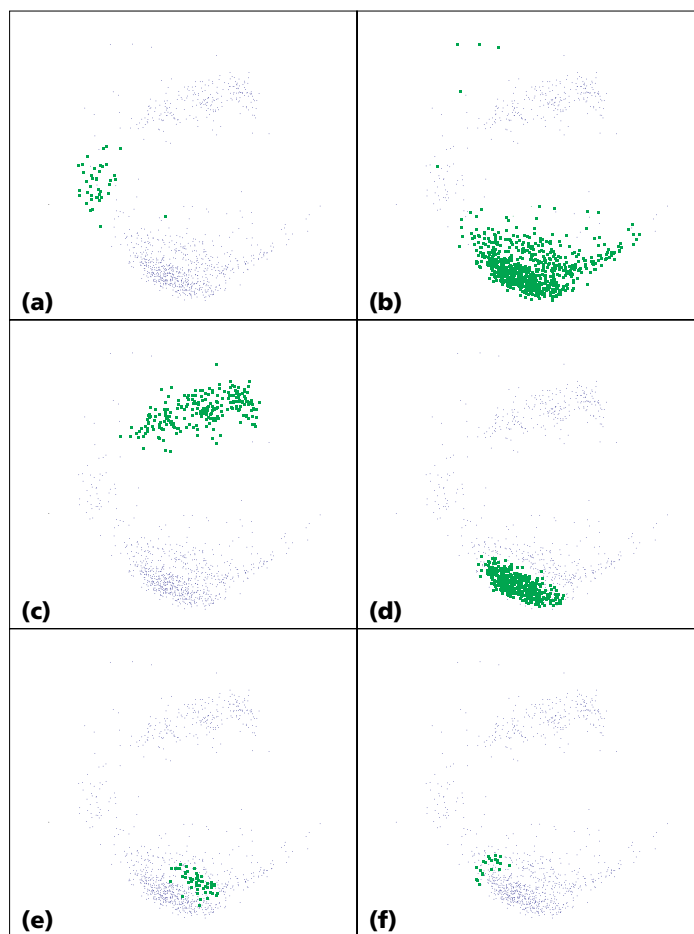
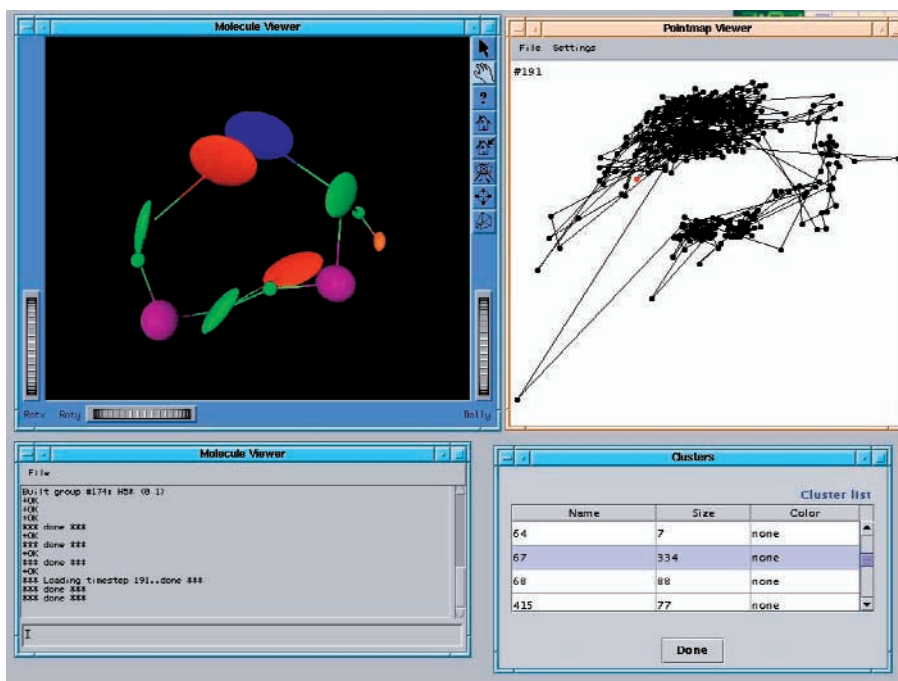


Figure 3. Different clusters identified in the trajectory by the clustering algorithm. (a), (b), and (c) show the trajectory's decomposition into three conformational clusters, whereas (d), (e), and (f) show the substructure of one such cluster.

The splitting line of the large cluster at the bottom is also visible in Figure 1. Such a pattern indicates another smaller conformational change, possibly in one of the glucose rings, independent of the large conformational change. Because it only affects a small part of the molecule, the conformational distance is smaller and imprinted like a fine structure on the clusters. That such small changes are visible in the plot is again made possible by taking several intramolecular distances as the feature vector.

To simplify a trajectory's analysis, we created a Java application that reads the output files of the combined plane mapping–cluster analysis program and displays a 2D map (see Figure 4). This program interacts directly with a molecular visualization application written in C++ using SGI's OpenInventor library by means of a Unix pipe. Whenever the user selects a point in the plane, the corresponding configuration appears

Figure 4. An integrated environment for cluster analysis, with a molecule viewer on the left and a trajectory map viewer on the right.



in the visualization program. The user can also choose to display identified clusters using different colors in the map.

Identifying which parts of the molecule are responsible for different conformational ensembles is much more difficult. We use a visualization application that lets the user form groups of atoms that are visualized by ellipsoids. In this way, we can easily reduce a molecule to its functional groups where it is much easier to spot conformational changes. However, small conformational changes as those that show up as a fine structure on the plane map are easily lost in this representation.

As a first attempt at aiding the eye in discovering unusual molecular motions, we implemented a simple OpenGL effect in the visualization application that lets us blend several animation frames in the hope that large changes will stand out more clearly in this representation (see Figure 5).

Information visualization and cluster analysis are powerful tools in analyzing molecular dynamics trajectories. Based on the notion of conformational distance, which we can derive from a feature vector built from intramolecular distances, we constructed a 2D map of the trajectory that reveals conformational ensembles and applied a cluster analysis procedure that allows for the automatic identification of these clusters. The whole procedure is applicable to any molecule without having any information about its structure. Certainly more research will help identify and visualize essential degrees of freedoms.

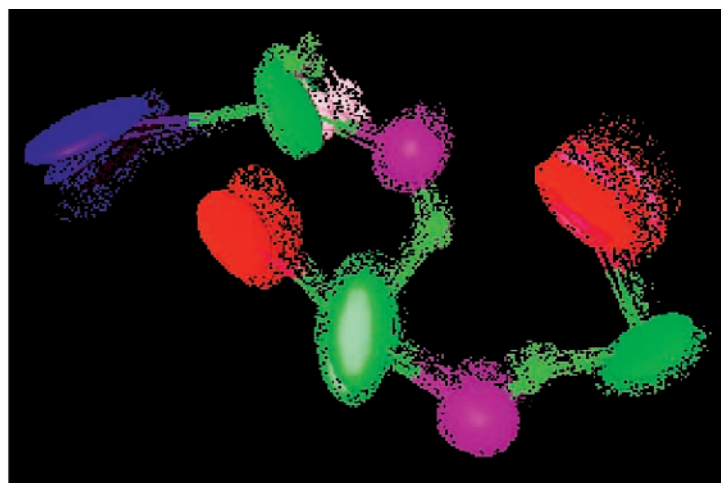


Figure 5. Visualization of the collective motion by an OpenGL fading effect.

References

1. S. Hayward and H.J.C. Berendsen, "Systematic Analysis of Domain Motions in Proteins from Conformational Change: New Results on Citrate Synthase and T4 Lysozyme," *Proteins*, vol. 30 no. 2, 1 Feb. 1998, pp. 144–154.
2. A. Amadei, A.B.M. Linssen, and H.J.C. Berendsen, "Essential Dynamics of Proteins," *Proteins*, vol. 17, no. 4, Dec. 1993, pp. 412–425.
3. X. Daura, W.F. van Gunsteren, and A.E. Mark, "Folding-Unfolding Thermodynamics of a Beta-Heptapeptide from Equilibrium Simulations," *Proteins*, vol. 34, no. 3, 15 Feb. 1999, pp. 269–280.
4. W. Huisinga et al., "From Simulation Data to Conformational En-

sembles: Structure and Dynamics-Based Methods," *J. Computational Chemistry*, vol. 20, no. 16, Dec. 1999, pp. 1760–1774; www.zib.de/PaperWeb/abstracts/SC-98-36.

5. A. Fischer, F. Cordes, and C. Schütte, "Hybrid Monte Carlo with Adaptive Temperature in Mixed-Canonical Ensemble: Efficient Conformational Analysis of RNA," *J. Computational Chemistry*, vol. 19, no. 15, 30 Nov. 1998, pp. 1689–1697.
6. F. Cordes, E.B. Starikov, and W. Saenger, "Initial State of an Enzymatic Reaction: Theoretical Prediction of Complex Formation in the Active Site of RNASE T1," *J. American Chemical Society*, vol. 117, no. 41, 18 Oct. 1995, pp. 10365–10372.
7. W. Donath and A. Hoffman, "Lower Bounds for Partitioning of Graphs," *IBM J. Research and Development*, vol. 17, no. 5, 1973, pp. 420–425.
8. M. Fiedler, "Property of Eigenvectors of Nonnegative Symmetric Matrices and Its Application to Graph Theory," *Czechoslovakian Mathematical J.*, vol. 25, no. 4, 1975, pp. 619–633.
9. B. Hendrickson and R. Leland, "An Improved Spectral Graph Partitioning Algorithm for Mapping Parallel Computations," *SIAM J. Scientific Computation*, vol. 16, no. 2, Mar. 1995, pp. 452–469.
10. H.L. Gordon and R.J. Somorjai, "Fuzzy Cluster Analysis of Molecular Dynamics Trajectories," *Proteins*, vol. 14, no. 2, Oct. 1992, pp. 249–264.
11. 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, 19 Jan. 1993, pp. 412–420.
12. P. Drineas et al., "Clustering in Large Graphs and Matrices," *Proc. Symp. Discrete Algorithms*, SIAM, Philadelphia, 1999, pp. 291–299; www.cs.yale.edu/users/kannan/Papers/cluster.ps.

Christoph Best is a Max Kade postdoctoral fellow at the Center for Theoretical Physics at the Massachusetts Institute of Technology. (He worked on the research

for this article while at the Konrad Zuse Institute.) His research interests include multiscale methods and their application in computational field theory and high-performance distributed computing. He has a PhD in theoretical physics from Johann Wolfgang Goethe University in Frankfurt, Germany. Contact him at the Center for Theoretical Physics, 6-308, MIT, Cambridge, MA 02139; c.best@computer.org; www.mit.edu/~cbest; <http://tigertiger.de/cb>.

Hans-Christian Hege is head of the Scientific Visualization Department at Zuse Institute, Berlin. His current research interests are in computer graphics, data visualization, image analysis, and biomedical computing. He is coeditor of Springer-Verlag's series *Mathematics + Visualization*, and acts as the managing director of Indeed Visual Concepts, a company specializing in data visualization that he cofounded in 1999. He studied physics at the Free University Berlin. Contact him at Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Division Scientific Computing, Dept. Visualization, Takustr. 7, D-14195 Berlin-Dahlem, Germany; hege@zib.de; www.zib.de/visual.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

THE CISE JOB BOARD

is an online, searchable jobs database that is accessed up to 175,000 times monthly.

Through the job board we place hundreds of people a year, but the most impressive fact is that we update the site daily. Your job will be seen by thousands of people worldwide.

FIND THE RIGHT FIT

To post or search jobs, visit

<http://computer.org/ciseportal>

today!

computing
in SCIENCE & ENGINEERING