PROTEINS: Structure, Function, and Bioinformatics 67:897–907 (2007)

Fast and Reliable Analysis of Molecular Motion Using Proximity Relations and Dimensionality Reduction

Erion Plaku,[1] Hernán Stamati,[1] Cecilia Clementi,[2,3]* and Lydia E. Kavraki[1,3,4]*
[1]*Department of Computer Science, Rice University, Houston, Texas 77005*
[2]*Department of Chemistry, Rice University, Houston, Texas 77005*
[3]*Structural and Computational Biology and Molecular Biophysics, Baylor College of Medicine, Houston, Texas 77030*
[4]*Department of Bioengineering, Rice University, Houston, Texas 77005*

ABSTRACT **The analysis of molecular motion starting from extensive sampling of molecular configurations remains an important and challenging task in computational biology. Existing methods require a significant amount of time to extract the most relevant motion information from such data sets. In this work, we provide a practical tool for molecular motion analysis. The proposed method builds upon the recent ScIMAP (Scalable Isomap) method, which, by using proximity relations and dimensionality reduction, has been shown to reliably extract from simulation data a few parameters that capture the main, linear and/or nonlinear, modes of motion of a molecular system. The results we present in the context of protein folding reveal that the proposed method characterizes the folding process essentially as well as ScIMAP. At the same time, by projecting the simulation data and computing proximity relations in a low-dimensional Euclidean space, it renders such analysis computationally practical. In many instances, the proposed method reduces the computational cost from several CPU months to just a few CPU hours, making it possible to analyze extensive simulation data in a matter of a few hours using only a single processor. These results establish the proposed method as a reliable and practical tool for analyzing motions of considerably large molecular systems and proteins with complex folding mechanisms. Proteins 2007;67:897–907.** © 2007 Wiley-Liss, Inc.

Key words: reaction coordinates; free energy landscapes; molecular dynamics simulation; distance measures; ScIMAP; DPES

INTRODUCTION

Molecular motion plays an important role in our understanding of biological processes at the molecular level. The computational study of molecular motion is typically performed by running simulations of molecular models on a computer system. The data sets of molecular configurations produced by computer simulations need to be analyzed and interpreted in order to extract the important features of the motion of the system under consideration. Different simulation techniques[1–9] and molecular models[10–22] have been proposed over the years to gather such data sets.

The extraction of important motion information from extensive sampling of the relevant configurations populated under specified conditions remains challenging. Researchers have suggested several approaches to address this problem, including clustering[23–29] and dimensionality reduction.[30–44] The recently proposed ScIMAP[45] method reliably extracts from extensive simulation data the main modes of motion of a molecular system, summarizing the molecular motion with only a few parameters. The ScIMAP method arranges the molecular configurations along a set of orthogonal axes that best characterize the molecular motion. Unlike principal components analysis and other linear dimensionality reduction techniques,[46–49] ScIMAP reliably captures even nonlinear motions.

The ScIMAP coordinates place the simulation configurations as points in a low-dimensional map that describes the geometric progression of the system as it evolves through its different states. The work in Das et al.[45] uses the ScIMAP-extracted coordinates as reaction coordinates to characterize a protein folding reaction, computing a free energy surface as a function of these coordinates. This free energy surface has been shown to correctly capture the main features in the folding landscape of a simulated protein folding reaction, such as the main folding route and the transition-state ensemble. The reaction coordinates computed by ScIMAP can be generally applicable to any molecular system, potentially eliminating the need to devise system-specific reaction coordinates.[45]

Despite the computational advantages ScIMAP offers over existing methods, such as fast local computations of

Grant sponsor: NSF; Grant numbers: CHE-0349303, CCF-0523908, CNS 0454333; Grant sponsor: NSF in partnership with Rice University, AMD and Cray; Grant number: CNS 0421109; Grant sponsor: NIH; Grant numbers: GM078988, 5 T90 DK070109-02 (Training fellowship from the Keck Center Pharmacoinformatics Training Program of the Gulf Coast Consortia awarded to HS); Grant sponsor: Robert A. Welch Foundation; Grant number: C1570; Grant sponsor: Sloan Foundation.

*Correspondence to: Lydia E. Kavraki, Department of Computer Science, Rice University, 6100 Main St., MS-132, Houston, Texas 77005, USA. E-mail: kavraki@rice.edu or Cecilia Clementi, Department of Chemistry, Rice University, 6100 Main St., MS-60, Houston, Texas 77005, USA. E-mail: cecilia@rice.edu.

Received 25 August 2006; Revised 2 November 2006; Accepted 13 November 2006

Published online 22 March 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/prot.21337

coordinates of oversampled regions and incorporation of parallel and iterative methods to perform efficient top-eigenvector computations,[50,51] its application to analyze motions of large molecules remains computationally expensive. As summarized in Materials and Methods, ScIMAP relies on proximity relations and the inherent connectivity of the input data set to reliably characterize molecular motion. The computation of proximity relations, which assigns to each molecular configuration in the data set a list of nearest neighbors as determined by some distance measure, e.g., lRMSD[52,53] (least root-mean-squared-deviation), constitutes the major bottleneck of ScIMAP, taking over 95% of the total computational time. The analysis of motions of large molecules requires the computation of nearest neighbors for millions of molecular configurations.

Even the most efficient algorithms require time proportional to the size of the data set in order to compute the nearest neighbors of a molecular configuration.[54–59] The quadratic computational cost associated with computing the nearest neighbors for millions of molecular configurations renders the application of ScIMAP to analyze motions of large molecules computationally challenging.

The main contribution of this work is to provide a practical tool for molecular motion analysis. The proposed DPES-ScIMAP (Distance-based Projection onto Euclidean Space ScIMAP) method, motivated by DPES,[60] is based on the idea of projecting the molecular configurations onto a low-dimensional Euclidean space[61] and computing proximity relations in the Euclidean space. The projection renders DPES-ScIMAP computationally practical, since generally fewer distance evaluations are required to compute proximity relations in a low-dimensional Euclidean space. Furthermore, proximity relations in the Euclidean space are based on the Euclidean distance, which can be evaluated at a fraction of the time required to evaluate distance measures for molecular configurations, such as lRMSD. As a result, DPES-ScIMAP computes proximity relations significantly faster than ScIMAP and thus effectively reduces the major computational bottleneck of ScIMAP.

The results presented in this work on the characterization of protein folding reactions reveal that the folding landscapes emerging from the application of DPES-ScIMAP and ScIMAP are practically indistinguishable. The advantage is that, in many instances, by using DPES-ScIMAP instead of ScIMAP, the computational time required to analyze the simulation data is reduced from several CPU months to just a few CPU hours. To put these results in a different perspective, the most relevant motion information can now be extracted from considerably large data sets in a matter of a few hours by running DPES-ScIMAP on a single processor, as opposed to hundreds of processors required by ScIMAP. These results establish DPES-ScIMAP as a practical tool for analyzing motions of large molecular systems starting from extensive simulation data.

MATERIALS AND METHODS

In this section, we first summarize the main ideas of ScIMAP.[45] We then describe in detail the proposed DPES-ScIMAP method and present a simple procedure for selecting good values for the parameters used by DPES-ScIMAP. We conclude the section by describing how the data sets of molecular configurations used in this work are generated.

ScIMAP: Scalable Isomap Method

The ScIMAP[45] method processes a given data set of molecular configurations to extract the most relevant coordinates that effectively characterize the process being studied (e.g. protein folding). The extracted coordinates constitute the vector basis of a low-dimensional embedding of the data set. The idea of ScIMAP is to find an embedding that preserves as much as possible the underlying connectivity of the data set. To this effect, proximity relations are defined for each configuration. The proximity relations of a configuration s are defined in ScIMAP as the k closest configurations according to a distance measure, such as lRMSD, and are referred to as the k exact nearest neighbors of s. Each configuration s is connected to k of its exact nearest neighbors and the emerging network or graph captures the connectivity of the data set, as Figure 1(a–c) illustrates. Each edge of the graph is associated with the lRMSD distance between the configurations that it connects. The distance between any pair of configurations s' and s'' is estimated as the length of the shortest path from s' to s'' in the graph, where the path length is obtained by adding up the lRMSD distances associated with the edges of the path. The reaction coordinates are then computed as a function of the distance matrix whose entries represent shortest-path distances between a significant portion of the configurations in the data set. A detailed description of ScIMAP is presented in Das et al.[45]

DPES-ScIMAP: A Practical Tool for Molecular Motion Analysis

The application of ScIMAP to analyze motions of large molecular systems remains computationally expensive due to the quadratic cost associated with the computation of proximity relations.[54–59] The proposed DPES-ScIMAP method however renders such analysis computationally practical by projecting the simulation data and computing proximity relations in a low-dimensional Euclidean space. The proximity relations of a configuration s, as computed by DPES-ScIMAP, are referred to as the k *approximate* nearest neighbors of s. Figure 1 provides an illustration.

The projection offers certain advantages. First, the projection enables DPES-ScIMAP to prune certain computations and reduce the overall number of distance evaluations required to determine the proximity relations of all data points. Second, DPES-ScIMAP gains additional computational efficiency by using the Euclidean distance to define proximity relations in the projected space, which can be evaluated much faster than the lRMSD distance used by ScIMAP. Although the projection could alter the proximity relations and thus affect the coordinates that are extracted, as our results indicate, when extensive simulation data has been gathered, differences between approximate and exact

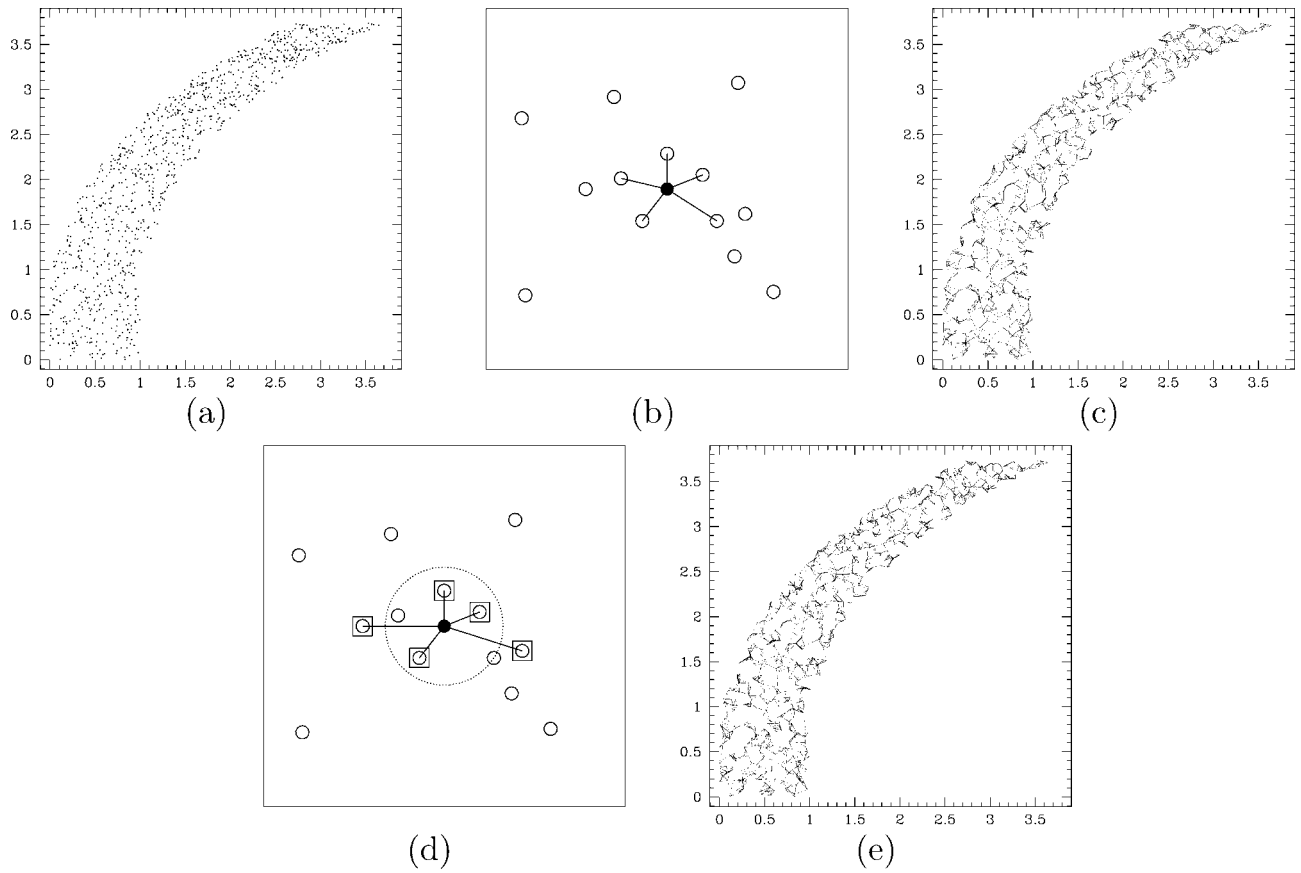

Fig. 1. (a-c) Illustration of proximity relations computed by ScIMAP. (**a**) A data set of points in $\Re^2$. (**b**) Proximity relations of a point are defined as its k nearest neighbors according to a distance measure. Each circle represents a point. Connections are shown from the point indicated by the filled circle to its $k = 5$ nearest neighbors. (**c**) Proximity relations of all the points capture the connectivity of the data set in (a). (d,e) Illustration of proximity relations computed by DPES-ScIMAP. (**d**) Proximity relations of a point s are indicated by connections between the filled circle and squares. A comparison with (b) shows that three out of five neighbors computed by ScIMAP and DPES-ScIMAP are the same. The other two neighbors computed by DPES-ScIMAP are outside the dashed circle centered at s with radius equal to the distance from s to its $k = 5$-th nearest neighbor. (**e**) Proximity relations of all the points in the data set of (a), as computed by DPES-ScIMAP. A comparison with (c) reveals that DPES-ScIMAP captures the connectivity of the data set in (a) essentially as well as ScIMAP.

nearest neighbors are negligible. As such, the coordinates extracted by DPES-ScIMAP and ScIMAP are practically indistinguishable. The overall effect is that the computation of proximity relations in the projected Euclidean space enables DPES-ScIMAP to remove the major computational bottleneck from ScIMAP while maintaining its reliability in characterizing molecular motions. As our results indicate, in many instances, DPES-ScIMAP reduces the required CPU time from months to hours.

Projection of the data set of molecular configurations onto a Euclidean space

The projection computed by DPES-ScIMAP is not a standard projection, but is instead based on distances between each molecular configuration in the data set to a set of carefully selected pivots. The data set S of molecular configurations is projected onto $\Re^m$, where $m > 0$ is the dimension of the Euclidean space. The projection of S onto $\Re^m$ is obtained by first selecting a set $P = \{p_1, p_2, \ldots, p_m\} \subset S$ of m pivots. Then each $s \in S$ is projected onto some $v(s) \in \mathcal{R}^m$ by setting the j-th coordinate of $v(s)$ to the lRMSD distance from s to p_j, i.e., $v(s)[j] = \text{lRMSD}(s, p_j)$, for $j = 1, \ldots, m$. The collection of all the projected configurations forms the set $V(S) = \{v(s) : s \in S\}$.

The objective is to select pivots that preserve relative distances between configurations in S when projected onto $\Re^m$, for example, when projections are close according to the Euclidean distance in $\Re^m$, then the corresponding configurations in S are close according to the lRMSD distance. A strategy that works well in practice is to select pivots that are as far away from each-other as possible.[60,62] The first pivot, p_1, is selected uniformly at random from all the points in S. The second pivot, p_2, is selected as the point in $S - \{p_1\}$ that is the farthest away from p_1 according to the lRMSD distance. In general, the j-th pivot, p_j, is selected as the point in $S - \{p_1, \ldots, p_{j-1}\}$ that is the farthest away from the already selected pivots, i.e., p_j maximizes $\min_{i=1..j-1} \text{lRMSD}(p_i, p_j)$.

Computation of proximity relations

DPES-ScIMAP computes proximity relations of configurations in S by computing proximity relations of the pro-

jections in $V(S)$. In particular, exact nearest neighbors of $s \in S$ according to the lRMSD distance are approximated by computing exact nearest neighbors of the corresponding projection $v(s) \in \Re^m$ according to the Euclidean distance. The first step of DPES-ScIMAP uses the projected points $V(S) \subset \Re^m$ to select $\ell > k$ configurations from S that are close to the exact nearest neighbors of s. For this reason, these ℓ configurations are selected as those ℓ configurations in S whose projections onto $V(S)$ correspond to the ℓ exact nearest neighbors of $v(s)$ according to the Euclidean distance. The computation of $\ell > k$ neighbors in the projected Euclidean space greatly increases the accuracy of DPES-ScIMAP, as indicated by the results. The second step of DPES-ScIMAP uses the lRMSD distance to select only the k closest out of the ℓ neighbors in the projected space as the approximate nearest neighbors of s.

Measuring the quality of the proximity relations

In this work, we also present results that indicate how similar the approximate nearest neighbors computed by DPES-ScIMAP are to the exact nearest neighbors computed by ScIMAP. We now discuss how to quantitatively measure these similarities. A strong indicator of the quality of the approximation is the ratio of false dismissals, RFD_ϵ.[63] The RFD_ϵ error indicates the fraction of the approximate nearest neighbors of a configuration s that are outside a small ball centered at s. The radius of the ball is set to $(1 + \epsilon)\alpha$, where $\epsilon \geq 0$ is some small constant and α is the distance from s to the k-th exact nearest neighbor of s. Small values of RFD_ϵ, for a small value of ϵ, indicate that most of the approximate nearest neighbors of s are not much farther away than the exact nearest neighbors of s. Figure 1(d) provides an illustration. Note, however, that two approximate nearest neighbors s' and s'' that are outside the small ball centered at s contribute the same value to RFD_ϵ even when $\text{lRMSD}(s, s') > \text{lRMSD}(s, s'')$. It is thus possible that two different sets of approximate nearest neighbors have the same RFD_ϵ error even when configurations in the first set are farther away from s than configurations in the second set. Intuitively, the second set provides a better approximation. This intuition is expressed by the ratio of distance errors, RDE,[63] which is small only when the sum of distances from s to its approximate nearest neighbors, denoted by β, is close to the sum of distances from s to its exact nearest neighbors, denoted by γ. The RDE error is then defined as $1 - \gamma/\beta$. The RFD_ϵ and RDE errors range in [0,1] Small values in this interval indicate that differences between approximate and exact nearest neighbors are negligible, which is the case for proximity relations computed by DPES-ScIMAP and ScIMAP, as indicated in Results and Discussion.

Parameter Selection

The application of DPES-ScIMAP to process a given data set S requires the selection of parameter values. The dimension (m) of the projection and the number (ℓ) of neighbors in the Euclidean space depend on several factors, such as the number of dimensions (d) required to represent each configuration, the number (n) of configurations in S, and the distribution of configurations in S.

Even though the parameters m and ℓ can be varied independently by savvy users, a simple way to select parameter values that works well in practice is to restrict the search of m and ℓ inside some reasonable intervals and use an error function $\text{err}(m, \ell)$, such as RFD_ϵ or RDE, to estimate the impact of the current selection for m and ℓ on DPES-ScIMAP. We compute $\text{err}(m, \ell)$ using only a small fraction of the points in S, e.g., $\min\{|S|/10, 1000\}$, selected uniformly at random. Since the dimension of the projection is more important than the number of neighbors in the projected Euclidean space, we start by setting $\ell = 25k$ and performing a binary search to find a value for $m \in [\alpha_0, \beta_0] = [1, d]$. During the i-th iteration, we search for values of $m \in [\alpha_i, \beta_i]$. The value of m during the i-th iteration is denoted by m_i and is equal to $m_i = (\alpha_i + \beta_i)/2$. The search stops during the i-th iteration with $m = m_{i-1}$ when $\text{err}(m_{i-1}, \ell)$ and $\text{err}(m_i, \ell)$ are similar. Otherwise, if $\text{err}(m_i, \ell)$ is large, we increase the lower bound on m by setting $\alpha_{i+1} = m_i$, and, if $\text{err}(m_i, \ell)$ is small, we decrease the upper bound on m by setting $\beta_{i+1} = m_i$. Once we have selected a value for m, we proceed with a similar binary search to find a value for $\ell \in [k, n]$.

We present results of the DPES-ScIMAP method for different values of a normalized parameter $D \in [0,1]$, which is introduced to express to the best of our intuition and experience the relation between m and ℓ and their dependence on d and n. The purpose of the normalized parameter D is to be able to give an indication on the accuracy and computational efficiency when DPES-ScIMAP is used with different values of m and ℓ to analyze the input data. More specifically, even though values of m, ℓ, d, and n could change depending on the data set that is being analyzed, it is desirable that similar values of the parameter D for different values of m, ℓ, d, and n to indicate similar results obtained by using DPES-ScIMAP to analyze the input data. For example, the same level of accuracy could be achieved by DPES-ScIMAP by either doubling the dimension of the projection (m) or doubling the number of the candidate neighbors (ℓ). For this reason, based on extensive testing, we set $D = \log_2(m)\log_2(\ell)/(\log_2(d)\log_2(n))$. The minimum and maximum values of D are achieved by setting $m = 1$ and $m = d$, $\ell = n$, respectively. This definition for D worked well in our experiments, since D is directly proportional to the logarithmic value of m and ℓ and indirectly proportional to the logarithmic value of d and n. The logarithmic function is used as a nonuniform scaling factor to smooth the dependence of D on m, ℓ, d, and n. In general, the analysis of large and high-dimensional data sets is challenging, which is reflected by a decrease in the value of the parameter D when the data dimension (d) and the number of data points (n) are increased. The quality of the proximity relations computed by DPES-ScIMAP can be improved by increasing the dimension of the projection (m) and/or the number of candidate neighbors (ℓ), which results in larger values of D. It is important to remember, however, that the parameter

TABLE I. Parameter Values Used by DPES-ScIMAP

SH3	$D = 0.52, m = 50, \ell = 8000$	$D = 0.36, m = 50, \ell = 500$	$D = 0.30, m = 15, \ell = 500$
CV-N	$D = 0.34, m = 50, \ell = 780$	$D = 0.28, m = 25, \ell = 780$	$D = 0.24, m = 15, \ell = 780$

Parameter values used by DPES-ScIMAP for the analysis of SH-3 and CV-N data sets. The value of the parameter D is computed as a function of the values selected for the dimension of the projection (m) and the number of candidate neighbors (ℓ) (see Materials and Methods for more details).

selection strategy described in this section computes D based on the values selected for m and ℓ. In Results and Discussion, we present results obtained by DPES-ScIMAP on different data sets for different values of the parameter D. In each case, we also indicate the values of m and ℓ that were used to obtain the value of D.

Protein Models

The extensive data sets of molecular configurations used in this work were obtained by molecular dynamics with coarse-grained models. Two different models were used to run the folding/unfolding simulations of SH3 (src-homology 3) and CV-N (cyanovirin-N) proteins.

For SH3, we used the coarse-grained model developed in Das et al.,[10] which was tested on SH3 in that same work, and was used in Das et al.[45] for the first application of ScIMAP. The model uses sequence information to produce a "minimally frustrated" folding landscape that drives the protein into its native state, but considering nonnative interactions as well.

For CV-N, following the work of Cho et al.,[64] we used a Gō-like model as defined in Clementi et al.[20] The Gō-like model considers native interactions only (with a short-range repulsion term for nonnative contacts) excluding disulfide bonds, which allows the appearance of an intermediate state.[64] As stated earlier, this intermediate state proves useful in testing the power of ScIMAP and DPES-ScIMAP for capturing nonlinear motions.

RESULTS AND DISCUSSION

The applications presented in this work show that DPES-ScIMAP is a reliable and practical tool for molecular motion analysis. As in Das et al.,[45] the focus is on the definition of reliable reaction coordinates that effectively characterize protein folding, starting from extensive simulation data. We show that DPES-ScIMAP characterizes folding practically as reliably as ScIMAP, even though DPES-ScIMAP alters the proximity relations used in the definition of the coordinates. The reason is that, for a wide range of parameter values, differences between proximity relations computed by DPES-ScIMAP and ScIMAP are negligible, and thus have minimal impact on the reaction coordinates that are extracted from simulation data.

We validate the accuracy and demonstrate the efficiency of the proposed DPES-ScIMAP method by characterizing the folding free energy landscapes associated with minimalist folding models of SH3 and CV-N, as described in Materials and Methods. The results we obtain reveal that, for good selection of parameter values, the folding landscapes emerging from the application of DPES-ScIMAP and ScIMAP are practically indistinguishable. The extracted reaction coordinates identify important features of the folding landscape including the folded and unfolded states, transition-state ensemble, and in the case of CV-N, on-route intermediate ensembles. The main advantage of DPES-ScIMAP is that while it characterizes the folding process essentially as well as ScIMAP, it does so at a small fraction of the computational cost required by ScIMAP. In many instances, the computational time is reduced from over two CPU months to less than seven CPU hours. We first studied the performance of DPES-ScIMAP using SH3, since it is the model used in Das et al.[45] to validate ScIMAP. We also conducted tests on a larger protein with a more complex folding mechanism, CV-N. Experimental data[65] and recent computational studies[64] have shown that CV-N has an intermediate state and that its folding landscape requires more than one reaction coordinate to be characterized. We present results when DPES-ScIMAP computes proximity relations by projecting each data set of protein configurations onto Euclidean spaces of varying dimensionality.

We present the results as a function of the parameter D, which, as detailed in Materials and Methods, defines how DPES-ScIMAP computes proximity relations. Table I contains a summary of the parameter values used in the experiments in this work. We conclude the section with a quantitative analysis that focuses on the differences between proximity relations computed by DPES-ScIMAP and ScIMAP and indicates that such differences are negligible.

Characterizing the Folding Free Energy Landscape of SH3

The SH3 data set consists of 473,300 protein configurations obtained by multiple folding/unfolding molecular dynamics simulations of a coarse-grained model close to the folding temperature.[10] Each protein configuration is represented by $d = 3 \times 57 = 171$ dimensions, corresponding to the (x, y, z) coordinates of the C_α atoms of the 57 residues of SH3. The SH3 data set is processed both by ScIMAP and DPES-ScIMAP to extract coordinates that characterize the folding reaction. Free energy surfaces can be defined as a function of the extracted reaction coordinates.[66–68] Figure 2(a) shows the folding landscape of SH3 at the folding temperature as a function of the first and second reaction coordinates, as computed in Das et al.[45] by ScIMAP. Figure 2(b–d) shows the folding landscape of SH3 as computed by DPES-ScIMAP for different values of the parameter D. Table II(a) indicates that DPES-ScIMAP

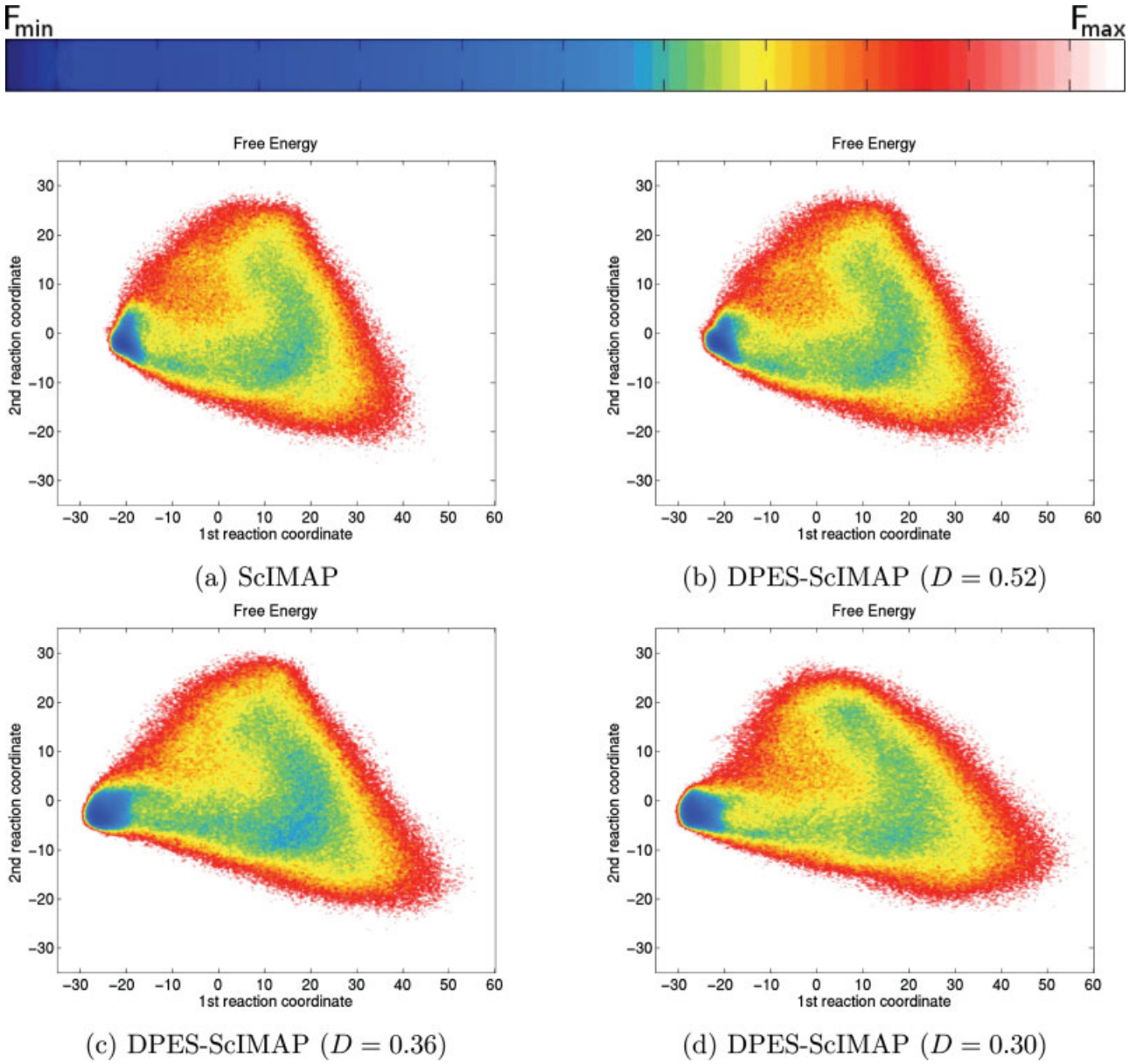

Fig. 2. Comparison of the folding free energy landscapes associated with SH3 as computed by ScIMAP and the proposed DPES-ScIMAP method. (**a**) Two-dimensional free energy profile as a function of the first and second reaction coordinates as extracted by ScIMAP. The free energy is shown color-coded, as indicated by the color bar at the top, with blue being the lowest and red the highest. (**b**–**d**) The free energy landscapes emerging from the application of DPES-ScIMAP for different values of the parameter *D* (see Materials and Methods for details on *D*).

significantly reduces the computational time required to effectively characterize the folding process of SH3.

Figure 2(a) indicates that SH3 is basically a two-state system, with the folded and unfolded states identified by the free-energy minima on the left and on the right, respectively. The free energy barrier separating these states can be identified as the transition-state ensemble as discussed in Das et al.,[45] where a thorough P_{fold} analysis[69] was used to validate the location of the transition state as computed by ScIMAP.

Figure 2(b) shows the folding landscape of SH3 as computed by DPES-ScIMAP when $D = 0.52$, which corresponds to $m = 50$ and $\ell = 8000$, as shown in Table I. A comparison between Figure 2(a,b) indicates that the folding landscapes emerging from the application of ScIMAP and DPES-ScIMAP are almost identical. To quantify the similarities, we compute the correlation between the reaction coordinates obtained by ScIMAP and DPES-ScIMAP. Figure 3 plots the correlation error $1 - (R_1 + R_2)/2$ as a function of the parameter D, where R_1 and R_2 indicate the Pearson correlation coefficient between the first and second reaction coordinates, respectively, as extracted by ScIMAP and DPES-ScIMAP. As Figure 3 reveals, the correlation error is practically zero for $D = 0.52$, indicating that

TABLE II. Computational Efficiency of DPES-ScIMAP

Method	CPU Time	Speedup
(a) SH3		
ScIMAP	15.60 days	1.00
DPES-ScIMAP ($D = 0.52$)	46.42 h	8.07
DPES-ScIMAP ($D = 0.36$)	1.84 h	203.46
DPES-ScIMAP ($D = 0.30$)	1.40 h	266.72
(b) CV-N		
ScIMAP	82.63 days	1.00
DPES-ScIMAP ($D = 0.34$)	6.67 h	297.33
DPES-ScIMAP ($D = 0.28$)	5.00 h	396.64
DPES-ScIMAP ($D = 0.24$)	4.15 h	477.88

Comparison of the computational efficiency of the proposed DPES-ScIMAP method over ScIMAP. In the case of DPES-ScIMAP, the value of the parameter D is indicated inside parentheses. The second column of each table indicates the CPU time required by ScIMAP and DPES-ScIMAP. The third column indicates the resulting computational speedup of DPES-ScIMAP over ScIMAP. The results indicate that DPES-ScIMAP significantly reduces the computational time—in many cases, from months to just a few hours.

the folding landscapes computed by ScIMAP and DPES-ScIMAP are essentially the same. However, DPES-ScIMAP speeds up the computation by a factor of eight times, as Table II(a) shows.

Figure 2(c) shows the results for a smaller value of D, namely $D = 0.36$. A comparison of Figure 2(a–c) reveals that, for this value of D, certain areas of the folding landscape appear slightly different. For instance, in the folding landscape obtained by DPES-ScIMAP, the basin around the unfolded state is slightly bigger and the folded state is less concentrated as well. However, the folding landscape computed by DPES-ScIMAP still remains remarkably similar to that of Figure 2(a) computed by ScIMAP, as indicated by the very small correlation error in Figure 3. Qualitatively, it can be seen that the locations, relative sizes and free energy values of the main features are still in good agreement with Figure 2(a). Since the free energies are well preserved, thermodynamic computations on this landscape remain reliable. Table II(a) indicates that DPES-ScIMAP requires less than 2 CPU hours of computation on a modern single processor machine, a speedup of more than 200 times with respect to ScIMAP.

Figure 2(d) shows the results when DPES-ScIMAP uses an even smaller value of the parameter D, namely $D = 0.30$. As expected, the differences in the folding landscape are more noticeable for such small value of D. The basin corresponding to the unfolded state occupies a larger portion of the plot, and the folded state region is also becoming larger. However, the relevant features are still clearly distinguishable, and a landscape computed in this way still shows the nature of the process as being primarily two-state, with a main route connecting the folded and unfolded states. The efficiency of DPES-ScIMAP is even higher in this case. As indicated in Table II(a), DPES-ScIMAP requires only 1.40 CPU hours, a speedup of 266 times over ScIMAP.

Overall, Figure 2 reveals that DPES-ScIMAP provides a robust and reliable method for analyzing the folding landscape of a two-state protein such as SH3. For a wide range of parameter values, the differences between the folding landscapes as computed by DPES-ScIMAP and ScIMAP are negligible. Figure 3 indicates that the correlation error quickly approaches zero for $D \gtrsim 0.31$. Additionally, important landscape features, such as the folded and unfolded states, main folding route, and transition-state ensemble, are highly preserved. Furthermore, DPES-ScIMAP extracts the most relevant reaction coordinates in a matter of a few CPU hours as opposed to over 15 CPU days required by ScIMAP.

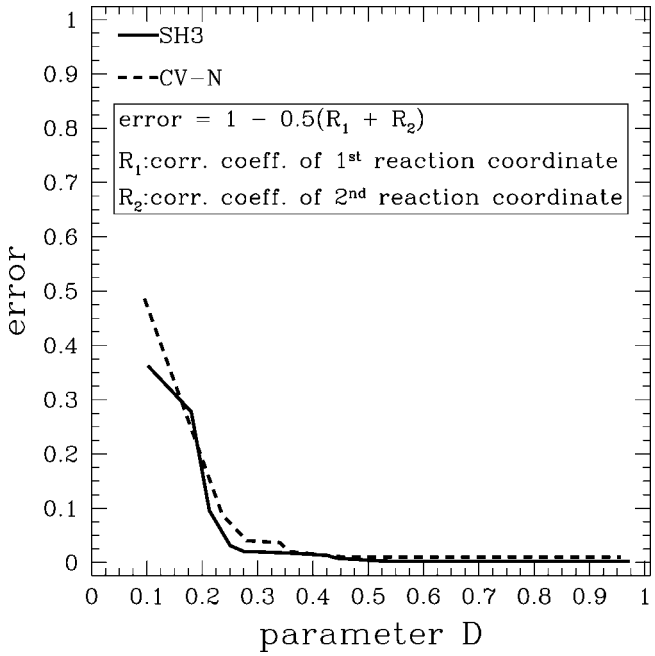

Fig. 3. Correlation error between the reaction coordinates extracted by ScIMAP and DPES-ScIMAP to characterize folding as a function of the parameter D. The error is expressed as $1 - (R_1 + R_2)/2$, where R_1 and R_2 denote the Pearson correlation between the first and second reaction coordinates, respectively.

Characterizing the Folding Landscape of CV-N

The CV-N protein data was gathered by running molecular dynamics simulations using a Gō-like coarse-grained[20] model, also close to the folding temperature. The CV-N data set consists of 640,000 protein configurations and each protein configuration is represented by $d = 3 \times 101 = 303$ dimensions, corresponding to the (x, y, z) coordinates of the C_α atoms of the 101 residues of CV-N. Figure 4(a) shows the folding landscape associated with CV-N, as computed by ScIMAP. Figure 4(b–d) shows the folding landscape of CV-N, as computed by DPES-ScIMAP for different values of the parameter D. Table II(b) indicates that the application of DPES-ScIMAP reduces the required CPU computational time from several months to just a few hours.

Figure 4(a) reveals that CV-N folds by going through an intermediate state when not constrained by disulfide bonds, in agreement with the results in Cho et al.[64]

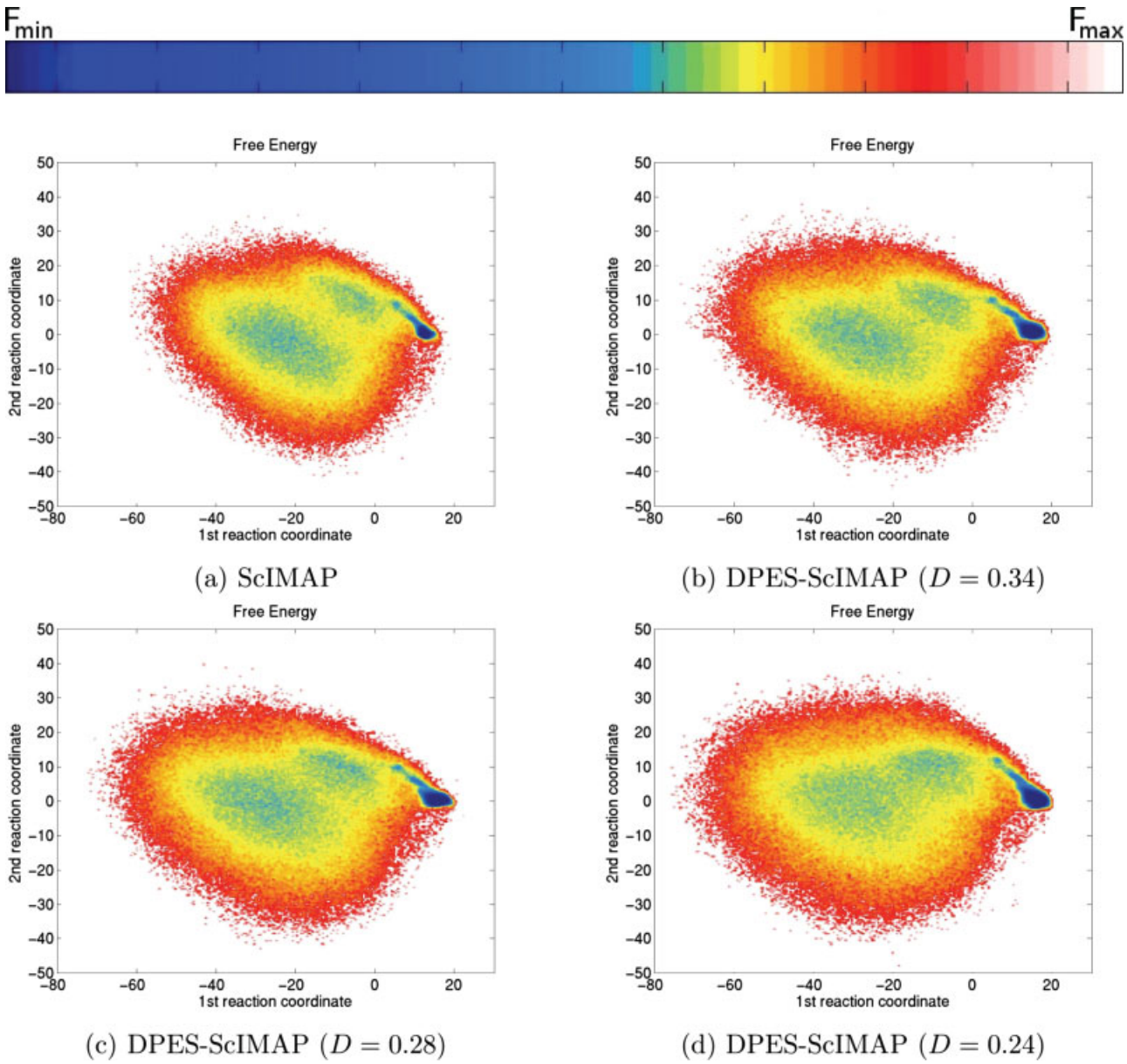

Fig. 4. Comparison of the folding free energy landscapes associated with CV-N as computed by ScIMAP and the proposed DPES-ScIMAP method. (**a**) Two-dimensional free energy profile as a function of the first and second reaction coordinates as extracted by ScIMAP. The free energy is shown color-coded, as indicated by the color bar at the top, with blue being the lowest and red the highest. (**b**–**d**) The free energy landscapes emerging from the application of DPES-ScIMAP for different values of the parameter D (see Materials and Methods for details on D).

The compact folded state, on the right, has the lowest free energy. A clearly defined route connects it to the intermediate state. Being a three-state system, this model of CV-N presents two transition states, and the P_{fold} analysis carried out in Cho et al.[64] shows that the ensemble of conformations with $P_{\text{fold}} = 0.5$ corresponds mainly to the intermediate state. Therefore, a P_{fold} test would not identify a transition state for this system.

Figure 4(b) shows the folding landscape of CV-N as computed by DPES-ScIMAP when $D = 0.34$, which corresponds to $m = 50$ and $\ell = 780$, as shown in Table I. Figure 3 indicates that quantitatively the folding landscapes emerging from the application of ScIMAP and DPES-ScIMAP are practically the same. Qualitatively, we observe that all the features have the same relative placement. The three minima and two main transition states also have almost the same free energy values. The only differences are the marginally bigger folded state and a negligible dispersion in the periphery, as in the case of SH3. The advantage is that DPES-ScIMAP reduces the CPU computational time from 82 days required by ScIMAP to 6.72 CPU hours, a speedup of around 300 times, as Table II(b) indicates.

Figure 4(c) shows the results when $D = 0.28$. Using a smaller value of the parameter D introduces some changes

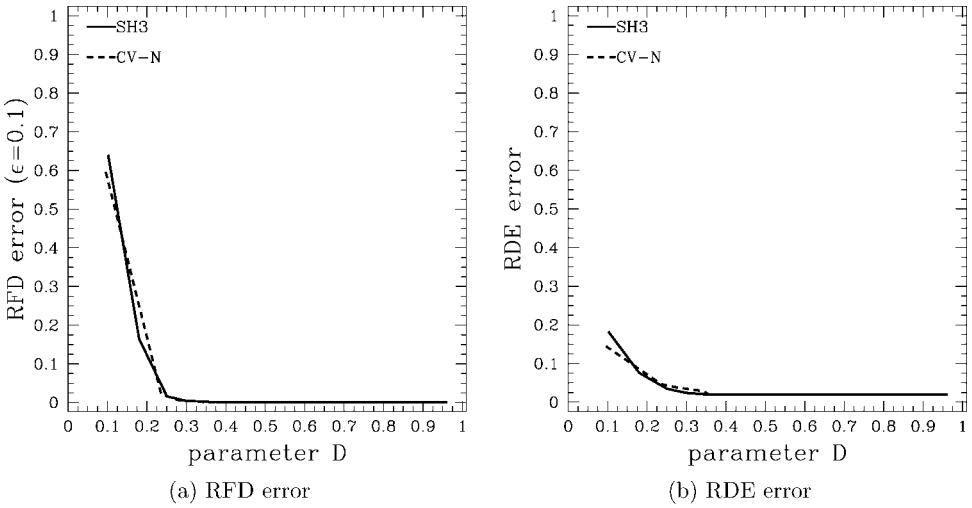

Fig. 5. A closer look at the accuracy of DPES-ScIMAP. Error measurements of the proximity relations as computed by ScIMAP and DPES-ScIMAP as a function of the parameter *D*.

in the folding landscape. Namely, the intermediate state is slowly starting to merge with the unfolded state and the folded state becomes slightly stretched horizontally. Table II(b) indicates that DPES-ScIMAP improves the computational efficiency by around 400 times.

Figure 4(d) shows the results when $D = 0.24$. This is a smaller value for the parameter D than in the previous cases and the resulting differences in the folding landscape are more noticeable. The intermediate and unfolded states have become less distinguishable. Certain features are however still clearly visible. The folded state, folding route and transition states remain similar to the original landscape. The overall shape is still preserved even for this very small value of D, so a landscape computed in this way can still prove useful at least for a preliminary analysis. Furthermore, DPES-ScIMAP finishes the computation in 4.15 CPU hours as opposed to almost three CPU months required by ScIMAP, a speedup of about 480 times.

We note that in the case of CV-N, DPES-ScIMAP maintains the accuracy of ScIMAP remarkably well even for smaller values of the parameter D than in the case of SH3. We believe the reason is that the Gō-like model used in the molecular dynamics simulations of CV-N to generate the input data set is simpler than the coarse-grained model used in the case of SH3. As summarized in the description of these protein models in Materials and Methods, the Gō-like model only considers interactions between native contacts. Consequently, the underlying connectivity of the resulting data set is smoother when a Gō-like model is used instead of the coarse-grained model defined by Das et al.,[10] where nonnative interactions are also present. We speculate that DPES-ScIMAP is able to capture the connectivity of the data set for smaller values of D in the case of CV-N, since the projection of a smoother data set of protein configurations onto a Euclidean space better preserves the proximity relations.

Overall, Figure 4 indicates that DPES-ScIMAP effectively characterizes the folding landscape of even large proteins with a complex folding mechanism, such as CV-N. The extracted reaction coordinates clearly identify important features of the folding landscape including the folded and unfolded states, the transition-state ensemble, and the on-route intermediate ensemble. Such results are obtained at a fraction of the computational cost required by ScIMAP. By reducing the CPU computational time from months to 4–7 hours, DPES-ScIMAP provides a reliable and practical tool for analyzing folding landscapes associated with large proteins.

A Closer Look at the Accuracy of DPES-ScIMAP

The results obtained in the case of SH3 and CV-N reveal that DPES-ScIMAP is a fast method and practically as reliable as ScIMAP. As detailed in Materials and Methods, DPES-ScIMAP computes proximity relations by projecting the data set of protein configurations onto a Euclidean space. Changes in the proximity relations due to the projection could impact the underlying connectivity of the data set and consequently alter the reaction coordinates that are extracted.

The claim is that DPES-ScIMAP preserves well the protein folding landscape since for a wide range of projections, differences in the proximity relations as computed by ScIMAP and DPES-ScIMAP are negligible. The analysis presented in this section provides quantitative evidence that confirms the above claim by examining differences in distances between exact and approximate nearest neighbors. The accuracy is high when such differences are negligible.

Figure 5 shows the results on the quality of approximate nearest neighbors computed by DPES-ScIMAP for the SH3 and CV-N data sets. We plot the $RFD_{0.1}$ and RDE errors as functions of the parameter D in Figure 5(a,b), respectively. We observe in Figure 5(a) that for very small values of D, the $RFD_{0.1}$ error is high. This indicates that most of the approximate nearest neighbors of a point s are

more than 1.1 times farther away from the k-th exact nearest neighbor of s. Consequently, the projection when D is very small alters the proximity relations. However, Figure 5(b) indicates that the RDE error is reasonably small. This implies that although the proximity relations are modified, the approximate nearest neighbors are not very far from the exact nearest neighbors. As a result, the proximity relations after the projection capture the connectivity of the data set, although not perfectly, and thus the folding landscapes emerging from the application of DPES-ScIMAP still preserve many of the important features, as the analysis of SH3 and CV-N revealed. More importantly, Figure 5(a,b) shows that even a small increase in the value of D causes the $RFD_{0.1}$ and RDE errors to drop significantly. We observe that when $D > 0.30$, the $RFD_{0.1}$ and RDE errors are close to zero. This indicates that differences between approximate and exact nearest neighbors are negligible. Therefore, the projection preserves remarkably well the proximity relations of each point and consequently the connectivity of the data set. As a result, the folding landscapes computed by ScIMAP and DPES-ScIMAP are practically indistinguishable.

CONCLUSIONS

We have presented a practical tool for reliably analyzing molecular motion and extracting reaction coordinates from simulation data. The application of the proposed DPES-ScIMAP method to the folding of a coarse-grained protein model of SH3 and a Gō-like model of CV-N reveals remarkably good agreement with the results obtained by using the recently proposed ScIMAP[45] method. The main advantage of DPES-ScIMAP is that while it is practically as reliable and robust as ScIMAP, it significantly reduces the computational cost. In many instances, the computational benefits of DPES-ScIMAP were dramatic. While ScIMAP requires months of CPU computation time, DPES-ScIMAP requires only a few CPU hours, making it possible to analyze molecular motions using only a single processor instead of hundreds of processors. The results presented in this work establish DPES-ScIMAP as a practical tool for conducting computational folding studies and analyzing motions of considerably large proteins and other biomolecular systems. Potential directions for future work include the analysis of much larger biomolecular systems and the development of a mathematical framework to better analyze the accuracy and computational efficiency of DPES-ScIMAP.

REFERENCES

1. Singhal N, Snow CD, Pande VS. Using path sampling to build better markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. J Chem Phys 2004; 121:415–425.
2. Tai K. Conformational sampling for the impatient. Biophys Chem 2004;107:213–220.
3. Hansson T, Oostenbrink C, van Gunsteren WF. Molecular dynamics simulations. Curr Opin Struct Biol 2002;12:190–196.
4. Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. Nat Struct Biol 2002;9:646–652.
5. Schlick T. Time-trimming tricks for dynamic simulations: splitting force updates to reduce omputational work. Structure 2001;9:R45–R53.
6. Sugita Y, Okamoto Y. Replica-exchange molecular dynamics method for protein folding. Chem Phys Lett 1999;314:141–151.
7. Marchi M, Procacci P. Coordinates scaling and multiple time step algorithms for simulation of solvated proteins in the NPT ensemble. J Chem Phys 1998;109:5194–5202.
8. Figueirido F, Levy RM, Zhou R, Berne BJ. Large scale simulation of macromolecules in solution: combining the periodic fast multipole method with multiple time step integrators. J Chem Phys 1997;106:9835–9849.
9. Feller SE, Zhang Y, Pastor RW, Brooks BR. Constant pressure molecular dynamics simulation: the Langevin piston method. J Chem Phys 1995;103:4613–4621.
10. Das P, Matysiak S, Clementi C. Balancing energy and entropy: a minimalist model for the characterization of protein folding landscapes. Proc Natl Acad Sci USA 2005;102:10141–10146.
11. Hubner IA, Deeds EJ, Shakhnovich EI. High-resolution protein folding with a transferable potential. Proc Natl Acad Sci USA 2005;102:18914–18919.
12. Shirts MR, Pande VS. Solvation free energies of amino acid side chain analogs for common molecular mechanics water models. J Chem Phys 2005;122:134508–134513.
13. Sorin EJ, Pande VS. Empirical force-field assessment: the interplay between backbone torsions and noncovalent term scaling. J Comput Chem 2005;26:682–690.
14. Fujitsuka Y, Takada S, Luthey-Schulten Z, Wolynes PG. Optimizing physical energy functions for protein folding. Proteins 2004;54:88–103.
15. Gallicchio E, Levy RM. AGBNP: an analytic implicit solvent model suitable for molecular dynamics simulations and high-resolution modeling. J Comput Chem 2004;25:479–499.
16. Wang J, Wolf RM, Caldwell JW, Kollman PA, Case DA. Development and testing of a general amber force field. J Comput Chem 2004;25:1157–1174.
17. Gnanakaran S, García AE. Validation of an all-atom protein force field: from dipeptides to larger peptides. J Phys Chem B 2003; 107:12555–12557.
18. Zhou R. Free energy landscape of protein folding in water: explicit vs. implicit solvent. Proteins 2003;53:148–161.
19. Ferrara P, Apostolakis J, Caisch A. Evaluation of a fast implicit solvent model for molecular dynamics simulations. Proteins 2002;46:24–33.
20. Clementi C, Nymeyer H, Onuchic JN. Topological and energetic factors: what determines the structural details of the transition state ensemble and en-route intermediates for protein folding? An investigation for small globular proteins. J Mol Biol 2000; 298:937–953.
21. Murphy RB, Philipp DM, Friesner RA. A mixed quantum mechanics/molecular mechanics (QM/MM) method for large-scale modeling of chemistry in protein environments. J Comput Chem 2000;21:1442–1457.
22. Nymeyer H, García AE, Onuchic JN. Folding funnels and frustration in off-lattice minimalist protein landscapes. Proc Natl Acad Sci USA 1998;95:5921–5928.
23. Domingues FS, Rahnenfuhrer J, Lengauer T. Automated clustering of ensembles of alternative models in protein structure databases. Protein Eng 2004;17:537–543.
24. Zhang Y, Skolnick J. SPICKER: a clustering approach to identify near-native protein folds. J Comput Chem 2003;25:865–871.
25. Agarwal PK, Procopiuc CM. Exact and approximation algorithms for clustering. Algorithmica 2002;33:201–226.
26. Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An effcient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 2002; 24:881–892.
27. Goldberg K, Roeder T, Gupta D, Perkins C. Eigentaste: a constant time collaborative filtering algorithm. Inf Retr 2001;4:133–151.
28. Troyer JM, Cohen FE. Protein conformational landscapes: energy minimization and clustering of a long molecular dynamics trajectory. Proteins 1995;23:97–110.
29. Shenkin PS, McDonald DQ. Cluster analysis of molecular conformations. J Comput Chem 1994;15:899–916.
30. Petrone P, Pande VS. Can conformational change be described by only a few normal modes? Biophys J 2006;90:1583–1593.

31. Lange OF, Grubmüller H. Collective Langevin dynamics of conformational motions in proteins. J Chem Phys 2006;124:214903–214920.
32. Kandiraju N, Dua S, Conrad SA. Dihedral angle based dimensionality reduction for protein structural comparison. In IEEE International Conference on Information Technology: Coding and Computing Las Vegas, NV, 2005. pp 14–19.
33. Shah M, Sorensen DC. Principle component analysis and model reduction for dynamical systems with symmetry constraints. In IEEE Conference on Decis and Control. Seville, Spain, 2005. pp 2260–2264.
34. Agrafiotis DK, Xu H. A geodesic framework for analyzing molecular similarities. J Chem Inf Comput Sci 2003;43:475–484.
35. Levy Y, Caisch A. Flexibility of monomeric and dimeric HIV-1 protease. J Phys B 2003;107:3068–3079.
36. Teodoro M, Phillips GN, Jr, Kavraki LE. Understanding protein exibility through dimensionality reduction. J Comp Biol 2003; 10:617–634.
37. Nolde S, Arseniev A, Orekhov V, Billeter M. Essential domain motions in barnase revealed by MD simulations. Proteins 2002; 46:250–258.
38. Hayward S, Kitao A, Berendsen H. Model-free methods of analyzing domain motions in proteins from simulation: a comparison of normal mode analysis and molecular dynamics simulation of lysozyme. Proteins 1997;27:425–437.
39. Balsera MA, Wriggers W, Oono Y, Schulten K. Principal component analysis and long time protein dynamics. J Phys Chem 1996;100:2567–2572.
40. Hayward S, Go N. Collective variable description of native protein dynamics. Ann Rev Phys Chem 1995;46:223–250.
41. Romo TD, Clarage JB, Sorensen DC, Phillips GN, Jr. Automatic identification of discrete substrates in proteins: singular value decomposition analysis of time-averaged crystallographic refinements. Proteins 1995;22:311–321.
42. Hayward S, Kitao A, Go N. Harmonic and anharmonic aspects in the dynamics of BPTI: a normal-mode analysis and principal component analysis. Protein Sci 1994;3:936–943.
43. García A. Large-amplitude nonlinear motions in proteins. Phys Rev Lett 1992;68:2696–2699.
44. Levitt M. Real-time interactive frequency filtering of molecular dynamics trajectories. J Mol Biol 1991;220:1–4.
45. Das P, Moll M, Stamati H, Kavraki LE, Clementi C. Low-dimensional free energy landscapes of protein folding reactions by nonlinear dimensionality reduction. Proc Natl Acad Sci USA 2006;103:9885–9890.
46. Jolliffe IT. Principal Component Analysis, 2nd ed. New York: Springer, 2002. 502 pp.
47. Hyvärinen A, Karhunen J, Oja E. Independent Component Analysis. New York: Wiley, 2001. 504 pp.
48. Cox TF, Cox MAA. Multidimensional Scaling, 2nd ed. London: Chapman & Hall, 2000. 328 pp.
49. Roweis ST, Saul LK. Nonlinear dimensionality reduction by Locally Linear Embedding. Science 2000.;290:2323–2326.
50. Lehoucq RB, Sorensen DC, Yang C. ARPACK Users' Guide: solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. SIAM 1998. 142 pp.
51. Maschhoff KJ, Sorensen DC. P_ARPACK: an efficient portable large scale eigenvalue package for distributed memory parallel architectures. Lect Notes Comp Sci 1996;1184:478–486.
52. Kabsch W. A solution for the best rotation to relate two sets of vectors. Acta Cryst A 1976;32:922–923.
53. Kabsch W. A discussion of the solution for the best rotation to relate two sets of vectors. Acta Cryst A 1978;34:827–828.
54. Indyk P. Nearest neighbors in high-dimensional spaces. In: Goodman JE, O'Rourke J, editors. Handbook of Discrete and Computational Geometry. Boca Raton, Fl: CRC Press, 2004. pp 177–196.
55. Korn F, Pagel BU, Faloutsos C. On the 'dimensionality curse' and the 'self-similarity blessing'. IEEE Trans Knowledge Data Eng 2001;13:96–111.
56. Hinneburg A, Aggarwal CC, Keim DA. What is the nearest neighbor in high dimensional spaces? In: International Conference on VLDB. Cairo, Egypt, 2000. pp 506–515.
57. Beyer K, Goldstein J, Ramakrishnan R, Shaft U. When is "nearest neighbor" meaningful? Lect Notes Comp Sci 1999;1540:217–235.
58. Borodin A, Ostrovsky R, Rabani Y. Lower bounds for high dimensional nearest neighbor search and related problems. In: ACM Symposium on Theory of Computing. Atlanta, Georgia, 1999. pp 312–321.
59. Weber R, Schek HJ, Blott S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In: International Conference on VLDB. New York, NY, 1998. pp 194–205.
60. Plaku E, Kavraki LE. Quantitative analysis of nearest-neighbors search in high-dimensional sampling-based motion planning. In: Workshop Algo Found Robot. New York, NY, in press. Available at http://www.wafr.org/papers/p31.pdf.
61. Weisstein EW. Euclidean space. From MathWorld-A Wolfram Web Resource, 2006.
62. Brin S. Near neighbor search in large metric spaces. In: International Conference on VLDB. San Francisco, California, 1995. pp 574–584.
63. Cui B, Shen HT, Shen J, Tan KL. Exploring bit-difference for approximate knn search in high-dimensional databases. In: Australasian Database Conference. Newcastle, Australia, 2005. pp 165–174.
64. Cho SS, Levy Y, Wolynes PG. P versus Q: structural reaction coordinates capture protein folding on smooth landscapes. Proc Natl Acad Sci USA 2006;103:586–591.
65. Barrientos LG, Lasala F, Delgado R, Sanchez A, Gronenborn AM. Flipping the switch from monomeric to dimeric CV-N has little effect on antiviral activity. Structure 2004;12:1799–1807.
66. Ferrenberg AM, Swendsen RH. Optimized Monte Carlo data analysis. Phys Rev Lett 1989;63:1185–1198.
67. Ferrenberg AM, Swendsen RH. New Monte Carlo technique for studying phase transitions. Phys Rev Lett 1988;61:2635–2638.
68. Roux B. The calculation of the potential of mean force using computer simulations. Comp Phys Comm 1995;91:275–282.
69. Du R, Pande VS, Grosberg AY, Tanaka T, Shakhnovich EI. On the transition coordinate for protein folding. J Chem Phys 1998;108:334–350.