JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 6, Number 2, 1999
Mary Ann Liebert, Inc.
Pp. 143–162

Lattice Simulations of Aggregation Funnels for Protein Folding

S. ISTRAIL,[1] R. SCHWARTZ,[2] and J. KING[3]

ABSTRACT

A computer model of protein aggregation competing with productive folding is proposed. Our model adapts techniques from lattice Monte Carlo studies of protein folding to the problem of aggregation. However, rather than starting with a single string of residues, we allow independently folding strings to undergo collisions and consider their interactions in different orientations. We first present some background into the nature and significance of protein aggregation and the use of lattice Monte Carlo simulations in understanding other aspects of protein folding. The results of a series of simulation experiments involving simple versions of the model illustrate the importance of considering aggregation in simulations of protein folding and provide some preliminary understanding of the characteristics of the model. Finally, we discuss the value of the model in general and of our particular design decisions and experiments. We conclude that computer simulation techniques developed to study protein folding can provide insights into protein aggregation, and that a better understanding of aggregation may in turn provide new insights into and constraints on the more general protein folding problem.

Key words: aggregation, inclusion body, lattice, protein folding, simulation

1. INTRODUCTION

1.1. Protein aggregation

PROTEIN AGGREGATION refers to the self-association of protein chains into insoluble, biologically inactive agglomerations during a folding reaction. These masses of protein, referred to as "inclusion bodies" if they form within cells, form very high molecular weight particles with diameters up to one micron (Krueger *et al.*, 1990). This aggregated state is distinct from the precipitated state. Aggregated proteins are not in equilibrium with correctly folding protein, and in general proteins cannot be separated from inclusion bodies by dilution into native buffer conditions (Marston, 1986); strong denaturants are required to break apart aggregated proteins (Mitraki *et al.*, 1991).

Protein aggregation has become a significant problem for work involving the expression of cloned genes. Prior to the widespread use of cloning, aggregation was primarily a concern when refolding proteins in laboratory experiments, and could usually be controlled in these cases by denaturing and refolding at low concentrations (Anfinsen and Haber, 1961). However, as it became possible to clone genes into heterologous

[1]Department of Algorithms and Discrete Mathematics, Sandia National Laboratories, Albuquerque, New Mexico.
[2]Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts.
[3]Department of Biology, Massachusetts Institute of Technology, Cambridge, Massachusetts.

hosts, aggregated states were found to be a common outcome of cloned gene expression (Marston, 1986; De Bernardez-Clark and Georgiou, 1991). Developing an understanding of aggregation, and thus potentially finding ways to control it, would have implications both for biological research and for the large-scale production of proteins by the biotechnology industry.

Aggregation of protein folding intermediates or protein unfolding intermediates has been implicated in a variety of human diseases. One class of these is referred to as amyloid diseases because of the accumulation of inclusion body-like protein deposits with certain common features. These include light chain amyloidosis, familial amyloid polyneuropathy, and Alzheimer's disease (Wetzel, 1997). Attempts to prevent the self-association of the amyloid precursor in Alzheimer's disease is a major area of therapeutic approach. In other diseases, including cystic fibrosis and certain defects of alpha-antitrypsin, mutations cause folding defects and defective chains aggregate, though not into amyloid deposits. It is important to distinguish the association of transient partially folded or unfolded species from the incorrect association of properly folded proteins (Betts *et al.*, 1997). The polymerization of sickle hemoglobin into filaments is the best studied example of the latter.

Prior theoretical work has studied the high-level kinetics and thermodynamics of inclusion body formation. Ziff (1984) explored a quantitative model of aggregation developed through Smoluchowski's coagulation equation. Kiefhaber *et al.* (1991) modeled the problem as a simple kinetic competition between the first-order correct folding reaction and higher-order aggregate formation. Speed *et al.* (1996) compared three quantitative models, matching each to experimental data, and concluded that a multimeric polymerization model best described the observed data. In addition, Patro and Przybycien (1994, 1996) have conducted lattice simulations to explore the structure of inclusion bodies and the kinetics of their formation. While these approaches have provided insight into the kinetics of the aggregation reaction, they have not explored the role of the folding process in facilitating aggregation or the properties of sequences that may promote or inhibit aggregation during folding.

Experimental evidence indicates that protein aggregation results from interactions of partially folded intermediates. Temperature sensitive mutations of the P22 tailspike have been shown to promote inclusion body formation at temperatures at which the native protein rarely aggregates (Haase-Pettingell and King, 1988). However, when these temperature-sensitive mutants are folded at permissive temperature, the resulting folded protein is equivalent to the folded native, and maintains the correct fold at temperatures far above those that produce misfolding (Thomas *et al.*, 1990). Aggregation rates have also been found to be reduced by folding in the presence of high concentrations of chaperonins (Gordon *et al.*, 1994).

Questions about folding kinetics have often proven difficult to answer through laboratory work. Existing experimental techniques have had only limited success in determining the folds of the transient, partially folded intermediates leading to native states since even early intermediates are large multimers of partially folded chains. Aggregation events have been even more difficult to analyze, as aggregated proteins appear during the folding process since their formation is kineticly controlled (Zettlmeissl *et al.*, 1979; Speed *et al.*, 1996); they therefore cannot in general be studied through equilibrium thermodynamics. Computational methods, including Monte Carlo simulations, have proven crucial in providing insight into aspects of the kinetics of protein folding that current laboratory techniques cannot provide, and may therefore have similar value for studies of aggregation.

1.2. Adapting lattice Monte Carlo methods to the study of aggregation

Lattice Monte Carlo models have long been a popular approach to studying the protein folding problem. Early lattice models attempted to model biologically relevant proteins, and were biased to fold to known native states (Taketomi *et al.*, 1975; Gō and Taketomi, 1978). Later protein folding models tended to be simpler, often using the HP model (Dill, 1985) and the two-dimensional square or three-dimensional cubic lattice. Since then, more complicated lattice models have been developed, which attempt to provide more flexibility and better capture realistic folds and side-chain packing.

These methods have been successful in exploring aspects of folding kinetics that have not been amenable to other techniques. The earliest models explored the relative importance of short- and long-range interactions and the nature of transitions between the native and denatured states (Taketomi *et al.*, 1975). Later simplified models provided information on coil-globule transitions (Šali *et al.*, 1994a,b) and the origin of secondary structure (Chan and Dill, 1989, Lau and Dill, 1990). More complicated models have provided more detailed theories for the formation of secondary structure and the forces driving it (Skolnick and Kolinski, 1991). For simple lattices, lattice methods can allow exhaustive enumeration of all small peptides and of all possible folds of those peptides (Lau and Dill, 1989).

Since protein aggregation during refolding results from interactions of kinetics intermediates (Zettlmeissl *et al.*, 1979; Speed *et al.*, 1996), studying aggregation should require a reasonable model of folding and the intermediates produced. Lattice Monte Carlo models have provided the only computationally tractable simulation method for studying this class of problem for large numbers of sequences, and furthermore, seem to be a reasonable simplification for studying aggregation events. We have therefore chosen to simulate aggregation through these methods.

2. METHODS

Our overall approach is to fold chains separately using conventional Monte Carlo techniques while testing for packings of the pairs of conformations in which aggregation is "favorable." However, just as many variants exist for such methods for studying protein folding, many design decisions are required to apply them to aggregation. At this stage, it remains unclear how effective different reasonable variations of these models will be at answering questions that come up in studies of aggregation. However, we can speculate on the benefits and drawbacks of different decisions by examining how analogous methods have performed for general protein folding problems. In Results, we describe those variants of the model we have applied to the present work; however, we include a discussion here of how the model might be applied differently and what the trade-offs in doing so might be. We divide the discussion of the simulation model into two broad areas, the folding model and the aggregation model, each of which must be specified in order to define a usable simulation model. In the appendix, we discuss some algorithmic issues in implementing such a simulation model.

2.1. The folding model

A necessary prerequisite to studying aggregation during folding is developing a model of the folding process itself. The folding model can influence aggregation directly, by limiting when and how aggregation can occur, or indirectly, by affecting the kinetics of the folding process. We now consider four basic areas in which the folding model can be adapted: the lattice, the amino acid alphabet, the energy matrix, and the model dynamics.

2.1.1. Lattice. One of the most important considerations in defining a lattice Monte Carlo model is the choice of lattice. Our proposed model is generalized with respect to the lattice, and should be adaptable to all major crystal lattice models. The advantages and disadvantages of each for our purposes are similar to those for the general protein folding problem. Simple lattices, such as the two-dimensional square lattice, may distort or fail to capture crucial aspects of the folding process. For example, they may not provide a sufficiently realistic model of secondary structure, which could be a serious problem if the packing of secondary structure elements on different proteins is an important factor in aggregation, as suggested by Brems *et al.* (1986). However, simple models greatly reduce computational time. In addition, two-dimensional models may be more accurate than three-dimensional models at producing realistic surface area to volume ratios for the small chain lengths that can be modeled (Chan and Dill, 1991). More complicated lattices, such as the 210 lattice (Skolnick and Kolinski, 1991), can more accurately model real proteins, allowing realistic depictions of secondary structure and more accurate packing of amino acids. However, if we rely on techniques involving exhaustive enumeration then more complicated lattices will require more computer time, although it is possible that with other methods more realistic lattices could prove more computationally tractable than simpler ones (Hart and Istrail, 1997a,b). As with standard protein folding simulations, important results about aggregation might be derived from even the simplest models, but more detailed work will likely require more realistic lattices.

2.1.2. Amino acid alphabet. Another important property of a lattice Monte Carlo model is the alphabet used for encoding amino acids. The simplest is the HP alphabet (Dill, 1985). This alphabet does not accurately represent the full range of differences between amino acids and tends to result in comparatively few sequences with unique native states (Chan and Dill, 1996), but is easily handled computationally and is simple enough to allow exhaustive enumeration of short peptides (Lau and Dill, 1989). Other models effectively support an unbounded alphabet by allowing each residue a unique set of interaction energies, determined by knowledge of the native conformation (Taketomi *et al.*, 1975). While such unbounded alphabets can produce rapid folding, they may be difficult to justify biologically. Another extension of the alphabet is the addition of solvent/amino

acid interaction energies (Sun *et al.*, 1995), which may better capture the effects of hydrophobic forces. Again, simple models may provide useful results which can then be probed more carefully by more realistic models.

2.1.3. Energy matrix. Another key consideration is the energy function for interactions between neighboring amino acids and between amino acids and the solvent. In a contact energy model, this energy function takes the form of a matrix of interaction values. The choice of energy matrix can affect aggregability directly, by influencing the energetics of forming aggregation complexes, or indirectly, by changing the kinetics (Miller *et al.*, 1992) or thermodynamics (Chan and Dill, 1996) of the folding process. We provide simulation evidence for the effect of the energy matrix on kinetics in section 3.3. Considerations in selecting the interaction energies include: how physically reasonable the energy matrix is; whether it tends to produce stable, unique natives; and how it affects the folding rate, which can be slowed by making folding less directed or by stabilizing local minima (Miller *et al.*, 1992). Alphabets can be selected empirically, based on amino acid contact frequencies observed in solved protein structures (Tanaka and Scheraga, 1975), although simulation evidence indicates that values derived in this way may not accurately reflect actual energy functions (Thomas and Dill, 1996). An alternative technique deduces an energy matrix, given some known native and nonnative folds of one or more chains, through a series of linear constraints imposed by the assumption of a gap between the native and nonnative free energies (Crippen, 1996). For simple alphabets, an energy matrix may be derived by trial and error, in order to give reasonable folding behavior for typical sequences.

2.1.4. Dynamics. One key issue in the model dynamics is the choice of move set. The move set can affect folding kinetics (Chan and Dill, 1994), and thus, potentially, aggregability. Some issues that must be considered when evaluating potential move sets are whether or not they are ergodic, how physically reasonable they are, and how quickly they allow movement between different conformations. Considering aggregability also suggests that we examine whether a move set biases chains to fold in ways that expose them to or protect them from aggregation. It is beyond the scope of this paper to consider the many move sets that have been proposed, their physical justifications, or their effects in practice. However, it is likely that aggregability will depend on choice of move set, as foldicity does (Chan and Dill, 1994).

Once a move set has been determined, it is necessary to decide the transition probabilities between neighboring folds of the chain. The most widely used method is the asymmetric Metropolis criterion (Metropolis *et al.*, 1953), which we can generalize for multiple chains either by randomly choosing one move among all moves possible for all chains, or by accepting one move for each chain per Metropolis step. We could also apply other variations on the more general idea of Kawasaki dynamics (Kawasaki, 1966a–c). For example, we can determine the energy ΔG_i of each possible move i, then choose one move at random among all moves with probability distribution:

$$p_i = \frac{e^{-\Delta G_i/kT}}{\sum_j e^{-\Delta G_j/kT}}$$

This method yields identical thermodynamics to the asymmetric Metropolis method, but potentially very different kinetics. It is not clear what effects these or other reasonable models of dynamics should have on aggregation rates, suggesting a need for further study.

2.2. *The aggregation model*

In general, we test aggregability of two chains by folding them independently and performing a test for the aggregability of the two current conformations after each folding event. Knowing the number of ways they can aggregate and the energies of those potential aggregation events as a function of time can allow us to estimate the probability that those two chains, folding together in a solution of known volume, will aggregate to each other. To achieve a realistic estimate, it would be necessary to incorporate statistics derived from the aggregation simulation model on the number of possible dimer interactions and their energetics into a more complicated formula accounting for other factors affecting the favorability of aggregation, such as the nature of the solvent, the solution concentration, and the temperature. This in turn would allow us to estimate the probability that a solution of a known concentration of simultaneously folding chains will produce a particular yield of non-aggregating, high-throughput proteins. For more accurate results, it may be necessary to model large numbers of chains explicitly; however, due to the computational difficulties this would involve, we currently restrict ourselves to two chains at a time.

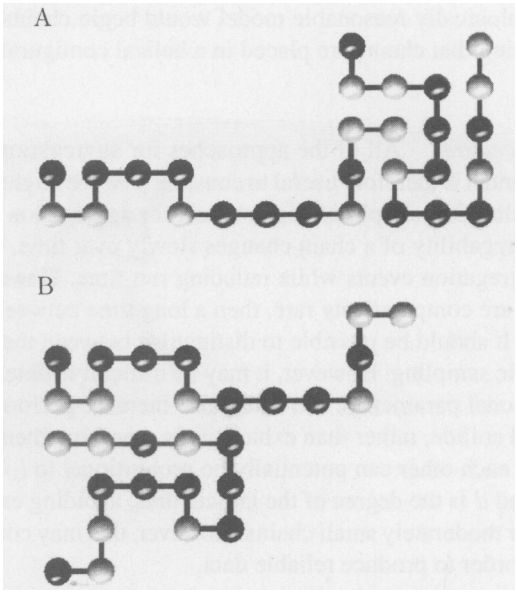

FIG. 1. Aggregation packings for a pair of chains. Two conformations of a short peptide are packed into two distinct packings, each of which has three H-H contacts. Each sphere in these pictures represents a single amino acid, with the lighter color representing hydrophobic and the darker hydrophilic amino acids.

2.2.1. Conditions for aggregation. Perhaps the most crucial issue in incorporating aggregation into a folding model is deciding under what conditions aggregation can occur. One simple answer is to use an energy cutoff. Specifically, we compare possible "aggregation packings" of two partially folded chains and allow aggregation to occur if the energy change of some packing is below a specified threshold. We define an aggregation packing of two chain conformations to be a way of placing those two conformations against each other on the lattice used in the simulation such that the chain folds are unchanged and the two chains do not overlap. Aggregation packings are illustrated in Fig. 1. This model allows for a relatively simple aggregation test. In addition, given an accurate estimate of the entropy loss of aggregation, we should be able to provide a realistic threshold at which aggregation becomes energetically favorable. A similar approach would allow aggregation for any packing energy, but with the probability of aggregation determined by the energy change of the aggregation packing; this would be a more complicated model and most likely would be less consistent between runs, but may be more realistic. Another very different approach involves adding geometric constraints, allowing aggregation only when a sizable portion of one chain can be packed into a corresponding pocket of the other. However, it is not currently known whether geometric constraints will more accurately model actual aggregation, and including them could substantially complicate the computational aspects of aggregation testing. Further investigation will likely be needed to determine the robustness of aggregation measures to such issues.

2.2.2. Initial conditions. Another issue is the initial condition of chains in an aggregability test. We must specify some starting state, and an unrealistic choice might significantly disrupt measured probabilities over the course of a folding reaction. The simplest model would begin all chains in a linear state. However, for many reasonable folding models, two identical chains will be highly aggregable when both are in linear states, since all hydrophobic residues will be exposed. This high aggregability early on could skew the total aggregation probabilities over the course of the folding process. Another possibility is to begin chains in a random state; this would largely eliminate the problem described above, but may be difficult to justify biologically. Although this is physically reasonable for fully denatured chains, it is a poor representation of the newly synthesized chain within cells. We might also allow chains to fold without the possibility of aggregation for some amount of time before we begin testing for aggregation. However, this requires finding some reasonable value for the time before aggregation can occur. A fourth possibility is to start chains in some "partially protected" state, which would reduce the probability of aggregation early in the simulation. One such state might be a maximally compact globule; we could justify this from results indicating that chains fold quickly to a compact state, then slowly to the native (Šali *et al.*, 1994a; Tiktopulo *et al.*, 1994), suggesting that the aggregation test should sample predominantly from intermediates that occur after the chain reaches a compact state. A

different and possibly more biologically reasonable model would begin chains in a helix or helix-like state, defensible from results suggesting that chains are placed in a helical configuration as they exit the ribosome (Lim and Spirin, 1985).

2.2.3. Simplified testing procedures. All of the approaches for aggregation testing described above can require considerable run-time, and it is therefore useful to consider how we might simplify the problem without substantially changing our results. One simplification is to test for aggregation only periodically, rather than after each folding event. If aggregability of a chain changes slowly over time, then this may still provide an accurate count of potential aggregation events while reducing run-time. However, if aggregability changes rapidly and aggregation events are comparatively rare, then a long time between samples could substantially alter measured aggregabilities. It should be possible to distinguish between these possibilities by comparing results with and without periodic sampling. However, it may be difficult to determine how robust the result is to changes in sequences and model parameters. We could also increase performance by randomly sampling possible ways two chains could collide, rather than exhaustively searching them. The number of aggregation packings of two chains against each other can potentially be proportional to $l_1 l_2 (d-2)^3$, where l_1 and l_2 are the lengths of the two chains and d is the degree of the lattice; thus, avoiding exhaustive checking could lead to a significant speedup even for moderately small chains. However, this may come at the expense of requiring many more simulation runs in order to produce reliable data.

2.2.4. Reversibility. A final issue is whether we allow reversible aggregation events. The simplest model would assume that aggregated chains cannot subsequently detach and fold to the native, in accordance with experimental observations of the properties of inclusion bodies, which are irreversibly associated under the conditions of their formation (Marston, 1986; Chrunyk *et al.*, 1993). However, we could permit some probability of chains breaking apart after aggregation, allowing the folding process to resume. This is closely related to considerations of aggregation conditions. If we allow aggregation to occur with some probability even when it is energetically unfavorable or only slightly favorable, then gathering realistic aggregation rates might require allowing reversible aggregation events. On the other hand, if we use a threshold to restrict aggregation events to very stable complexes, then allowing the reaction to reverse itself might have very little impact on the results and therefore be an unnecessary complication. Resolving the question of how to account for reversibility in the simulation model also may require further experimental investigation.

3. RESULTS

We have applied a basic variant of this model to several experiments on the nature and source of aggregability and the influence of the model on experimental results. For these preliminary experiments, we used simple variations on the model, both because those variations are computationally easier to implement and run and because we believe it is useful to learn what we can from the simplest models before attempting to extend them. All of the experiments described below therefore use a two-dimensional lattice and either an HP or HP/solvent alphabet. In addition, all apply an energy threshold aggregation test, with the threshold energy set at -15 kcal/mole, and treat aggregation events as nonreversible. We use a helix-like initial state for all chains. For these tests, we have worked only with very short sequences, because at longer lengths the computational time becomes prohibitive.

3.1. Predicting aggregability

Our first experiments investigated the robustness of aggregability as a sequence property and the identification of characteristics of a sequence which influence its propensity for aggregation. For this experiment, we have tested all sequences of length 16 that have unique native states for the HP model using only H-H contact energies on the two-dimensional square lattice. For each such sequence, we ran Metropolis Monte Carlo simulations of two instances of that sequence, using the MS2 move set of Chan and Dill (1993), an energy of -5 kcal/mole, and a temperature of 293 K. We sampled the possible ways the two could aggregate every 100 steps. We will henceforth refer to the model created by this choice of parameters as our standard model. Because the gaps between the energies of the natives and the energies of other nearby conformations are too small to stabilize chains of this length, we artificially froze each chain's state when it reached native. We terminated each test when both chains reached the native state. We have defined the aggregability of a

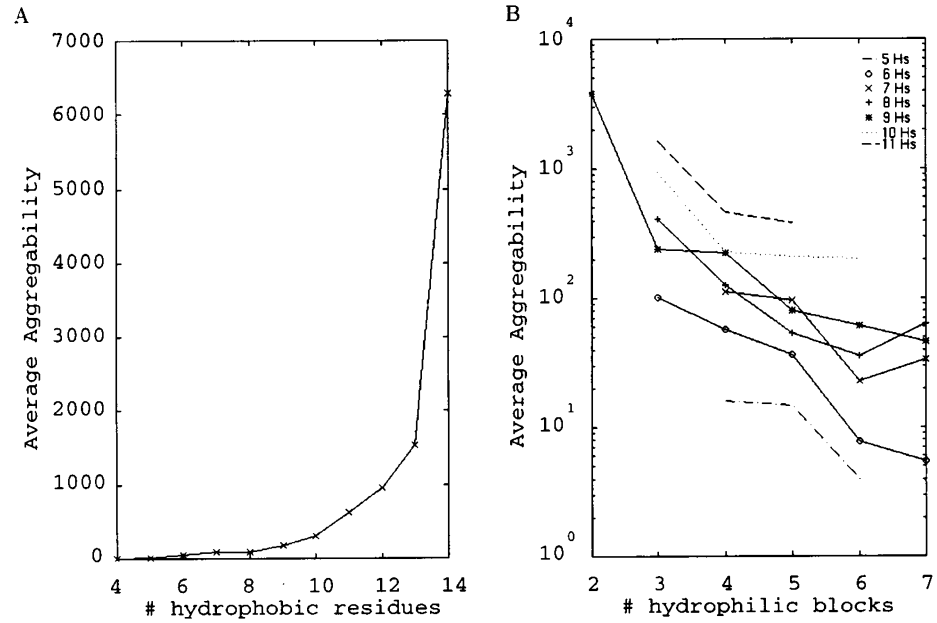

FIG. 2. Average aggregability as a function of HP sequence. Average aggregability data is shown for native-unique sequences of length 16. Aggregability is defined as the number of ways of packing two folding chains together with energy at most −15 kcal/mole, sampling every 100 moves until both chains have folded to the native state. (**A**) Aggregability data separated according to number of hydrophobic residues. (**B**) Semi-log plot of data further subdivided by the number of disjoint blocks of hydrophilic residues (i.e., groups of consecutive hydrophilic residues containing no hydrophobic residues). Data in B is shown only for numbers of hydrophobic residues that would produce at least two data points.

chain to be the number of possible aggregation events we counted, sampling every 100 moves, for the chain folding alongside a copy of itself.

3.1.1. Aggregability and HP sequence. The first concern was in the direct relationship between aggregability and HP sequence. As we would expect, as the number of hydrophobic residues increases, average aggregability increases. This is illustrated in Fig. 2A. This in itself suggests some competition between aggregability and foldicity, at least for the HP model and the sequence length we considered, as sequences with unique native states have slightly more hydrophobic residues than would be expected by chance, with 8.67 hydrophobic residues per sequence among native-unique sequences of length sixteen, versus 8.00 among all sequences of length 16. However, the numbers of hydrophobic and hydrophilic residues alone do not entirely determine the aggregability. Even when the number of hydrophobic residues is held constant, aggregability is related to how concentrated hydrophobic and hydrophilic residues are in contiguous blocks of the sequence. Figure 2B shows average aggregabilities for fixed numbers of hydrophobic residues as a function of the number of disjoint blocks of hydrophilic residues. This data illustrates a general, although not entirely consistent, trend in which fewer blocks of either type leads to increased aggregability; a similar trend holds, although with more inconsistencies, if we look instead at the number of hydrophobic blocks. These trends implies that sequences which alternate often between hydrophobic and hydrophilic residues are better protected from aggregation than those that concentrate all hydrophobics and all hydrophilics into a few large groups.

3.1.2. Aggregability and folding time. We have also analyzed aggregability as a function of folding time. This data is illustrated in Fig. 3A. The data is clustered near the origin, meaning that most native unique sequences have relatively low folding times and aggregabilities. Those that have very high aggregability tend to have intermediate folding times. Those with the longest folding times tend to have comparatively low aggregabilities. The low aggregabilities of the fastest folding sequences can be largely explained by the fact that they have little time in which to aggregate before they fold to native. We can confirm this by looking at aggregability divided by folding time, as a function of folding time, illustrated in Fig. 3B. This illustrates that sequences with rapid folding times tend to have comparatively high propensities for aggregation before they finish folding, but that this is more than offset by the short time they spend before folding to native. The low aggregabilities of slow-folding chains appears to be due to their unusually large

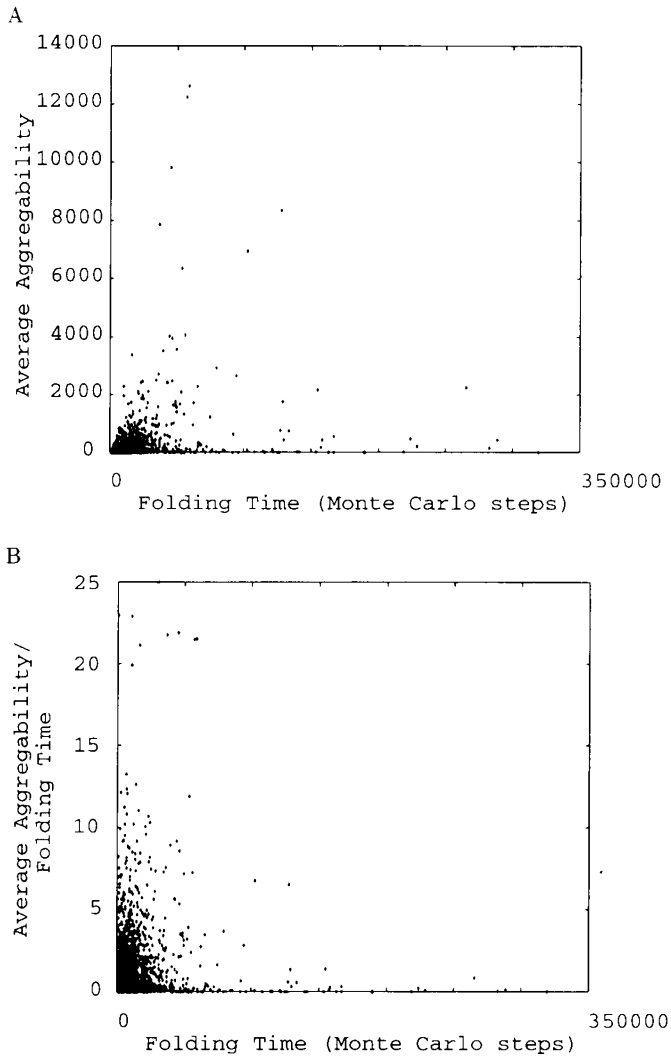

FIG. 3. Aggregability versus folding time. Aggregability values are plotted against the time required for both chains of a pair to fold to native for all native-unique sequences of length 16 using our standard model. (**A**) Aggregability versus folding time. (**B**) Aggregability divided by folding time versus folding time.

numbers of hydrophilic residues, making the folding process less directed while simultaneously reducing the possibilities for aggregation. On the whole, this data suggests that while rapid foldicity often accompanies low aggregability, they are distinct properties with neither guaranteeing the other and with occasional competition between the two.

3.1.3. Aggregability and the native energy. Finally, we looked at how aggregability relates to the native state. Figure 4A shows that lower native energies tend to produce greater aggregability. However, this may be a side effect of the correlation we would expect between low native energies and large numbers of hydrophobic residues. We can confirm this by breaking data down according to numbers of hydrophobic residues per sequence, shown in Fig. 4B. We then see that if the number of hydrophobic residues is held constant, then sequences tend to become less aggregable with decreasing native energies. This suggests that properties of the native state, and in particular, how well the native state buries its hydrophobic residues, can help to predict aggregability during the folding process. We chose to look at the native energy because it is an easily calculated sequence property on the lattice which exhibits a reasonable amount of variation among the 16-mers we consider in these experiments; other sequence properties may prove stronger predictors of aggregability when calculated using more realistic models and more biologically relevant sequence lengths.

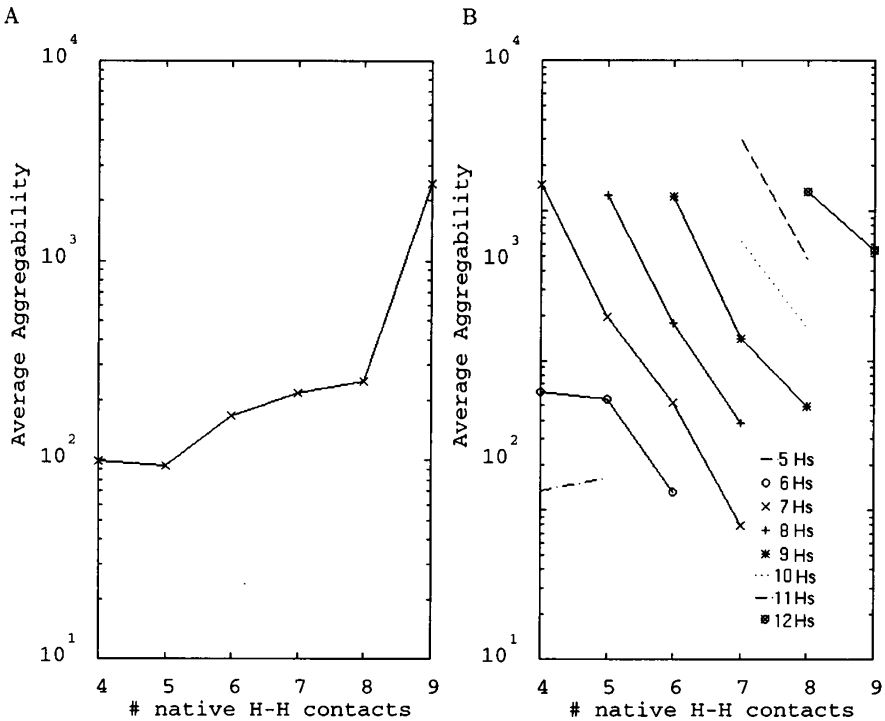

FIG. 4. Aggregability versus native energy. (**A**) Average aggregability as a function of the number of H-H contacts in the minimum energy state for all native-unique 16-mers. (**B**) Average aggregability as a function of the number of native H-H contacts for native-unique 16-mers with fixed numbers of hydrophobic residues.

3.2. Properties of aggregability

In order to better understand the nature of aggregability, we have attempted more thorough testing of a small number of slightly longer sequences. We chose 10 sequences randomly from among the native-unique sequences of length eighteen and ran each through 10 repetitions of an aggregability test using our standard model, as described for the previous experiment. We recorded both the number of ways of aggregating and the energy of each sequence as a function of number of Monte Carlo moves completed, sampled every 100 moves. This data is illustrated in Fig. 5.

3.2.1. Aggregability as a sequence property. One feature of this data is that the general level of aggregability per unit time of a particular chain pair during the folding process is approximately consistent between separate runs of a single chain and often noticeably different between runs of separate chains. For example, the difference between Fig. 5B and Fig. 5F is noticeably greater than that between distinct runs of either particular chain. This confirms the notion that aggregability as we measure it is a meaningful and intrinsic property of sequences. However, we can also note that the large differences in folding time between runs can significantly affect overall aggregability.

3.2.2. Aggregability of the native state. A second important feature is that in all cases there is a noticeable drop in aggregability once one chain reaches the native state and is frozen there by the simulator. In some cases, the aggregability of the pair drops to zero as soon as either one reaches native, while in others it remains nonzero but always at significantly lower levels than when both were folding. This may be explained by the fact that the native would be expected to bury its hydrophobic residues more effectively than the folding intermediates, thereby insuring that there are few exposed hydrophobics with which an aggregation event can occur. This observation creates the notion that some conformations of a sequence, including the native, are better "protected" than others and are therefore less susceptible to aggregation.

3.3. Dependence on the energy matrix

In order to begin exploring the robustness of the model, we have examined how the results depend on the energy matrix. For this experiment, we have examined three different energy matrices:

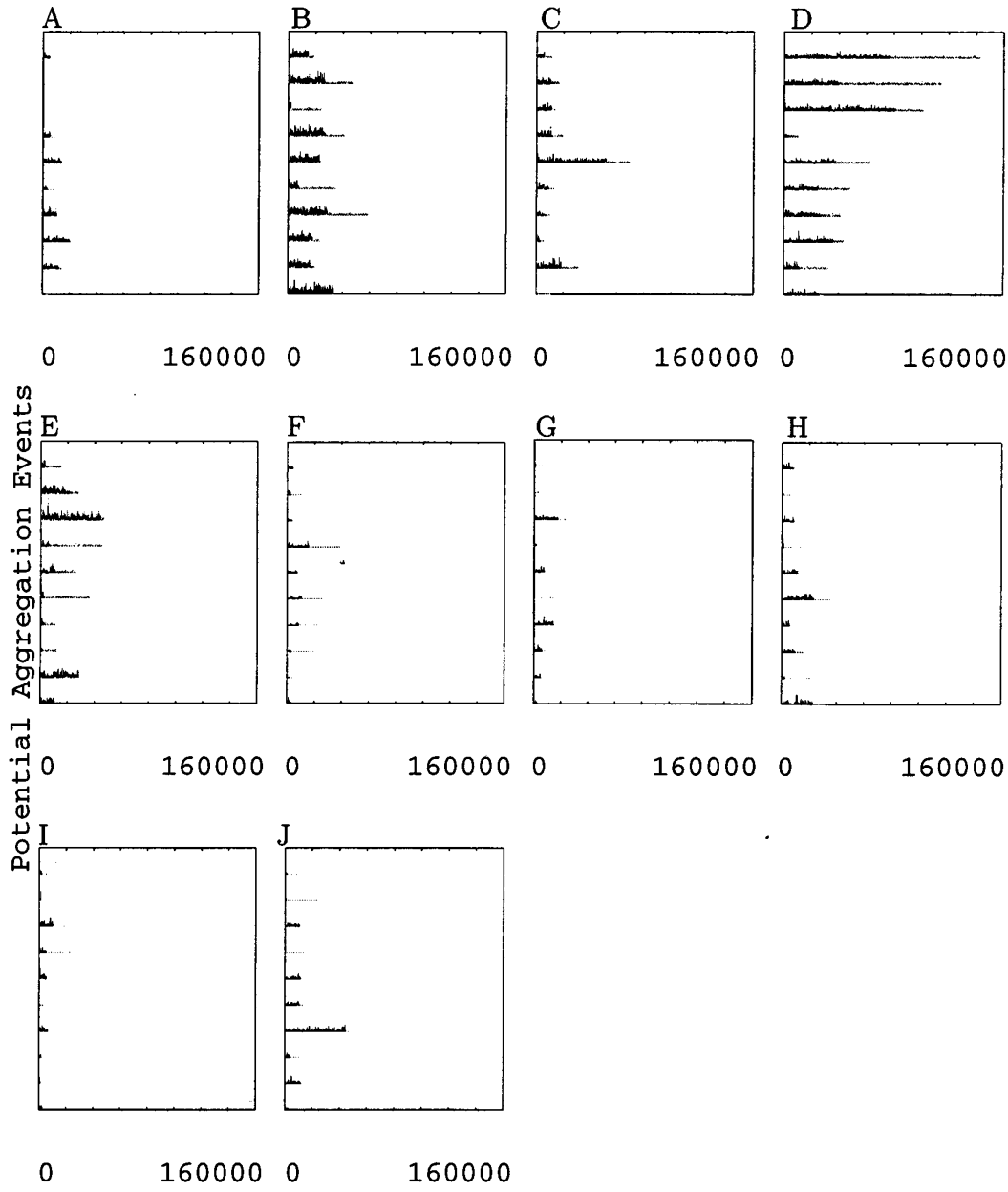

FIG. 5. Aggregability as a function of time for 10 peptides (A, B, C, D, E, F, G, H, I, and J) in 10 experiments each. Each picture plots the number of potential aggregation events per unit time measured for a pair of identical chains during the folding process, for each of 10 separate runs. Data was sampled every 100 Monte Carlo steps. The change from black to grey in each graph shows where one of the chains first reached the minimum energy state, at which point it was frozen in that state. A run was ended once both chains had reached their minimum energy state.

- M_1: H-H contact energy is -5; all others are 0.
- M_2: H-H contact energy is -3; H-solvent is 1; all others are 0.
- M_3: H-H contact energy is -1; H-solvent is 2; all others are 0.

M_1 corresponds to the standard HP energy of Dill (1985), while M_2 and M_3 are variants on the sHP model of Sun *et al.* (1995). These matrices were chosen so that in each case the net energy contribution caused by an H-H contact (which is equal to the H-H contact energy minus twice the H-solvent contact energy) will be -5.

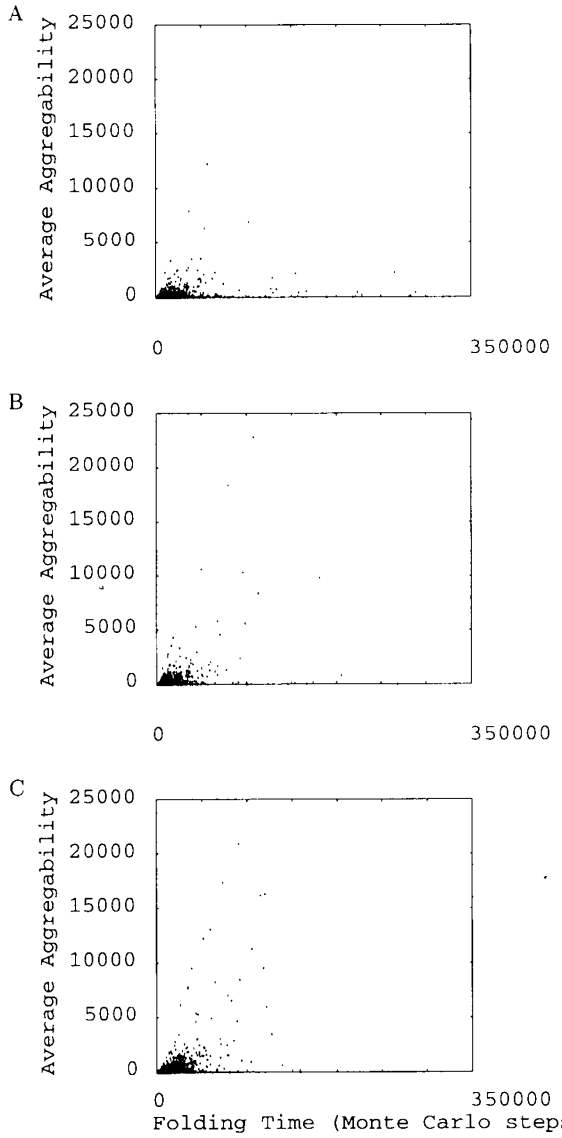

FIG. 6. Aggregability versus folding time for three energy matrices. Aggregability tests were performed for the three energy matrices, using all sequences of length 16 that have unique native states on the two-dimensional square lattice for all three matrices. Testing conditions were otherwise identical to those in section 3.1. (**A**) Data from M_1. (**B**) Data from M_2. (**C**) Data from M_3.

As in section 3.1, we used sequences of length 16. In order to insure that results are comparable between the three matrices, we have examined only those sequences that have unique native states for all three matrices. Otherwise, experimental conditions are identical to those of section 3.1.

3.3.1. Features preserved between energy functions. One notable observation is that all three experiments share the qualitative features observed in section 3.1, although there are significant quantitative differences between them. Plots of aggregability as a function of folding time are shown for the three matrices in Fig. 6. For all three matrices, rapidly folding and slowly folding sequences have comparatively low aggregation rates, with high aggregability sequences concentrated around intermediate folding times. Figure 7 shows three representative tables of aggregability as a function of numbers hydrophilic blocks, displaying the same general trend observed in section 3.1, although less strongly as H-solvent energy increases. Despite these similarities, average aggregabilities increase as we increase the H-solvent energy, with values of 246, 264, and 416 for M_1, M_2, and M_3, respectively. The increased aggregabilities are not surprising, since any aggregation packing for M_2 must have at least as much energy as the same packing for M_1, and any aggregation packing for M_3

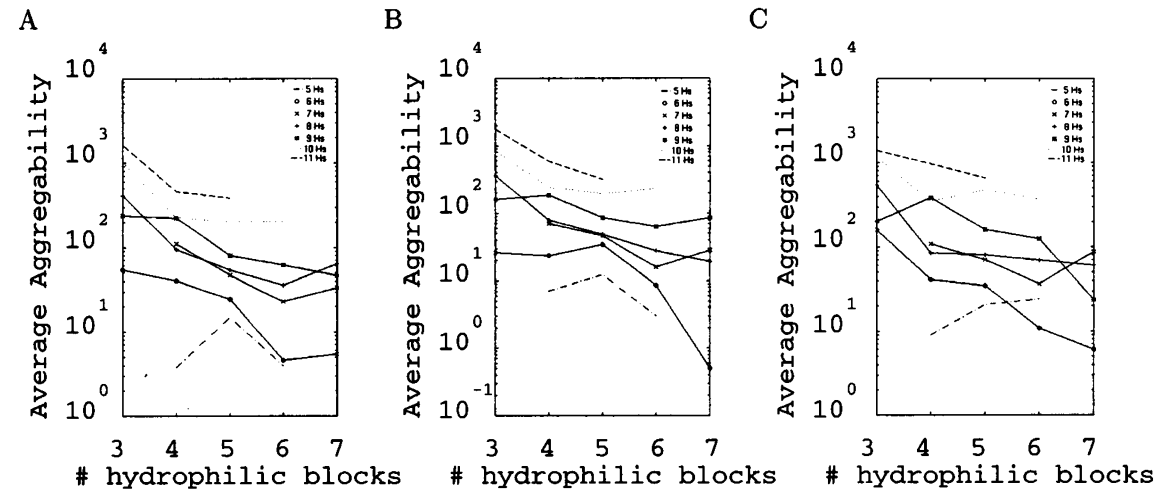

FIG. 7. Aggregability as a function of the number of hydrophilic blocks for distinct numbers of hydrophobic residues. Data is separated for each energy matrix according to the number of hydrophobic residues per sequence then average aggregability values are plotted for each number of hydrophilic blocks. (**A**) Data from M_1. (**B**) Data from M_2. (**C**) Data from M_3.

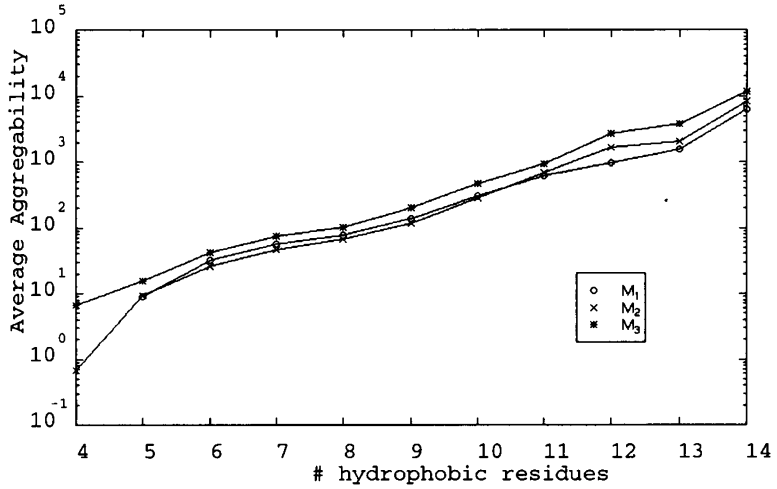

FIG. 8. Semi-log plot of aggregability as a function of number of hydrophobic residues for three energy matrices. Average aggregability data is plotted, separated by the number of hydrophobic residues in each sequence for M_1, M_2, and M_3. The value of zero for M_1 at four hydrophobics is not shown, since it cannot be displayed on a semi-log plot.

must have at least as much energy as the same packing for M_2. Therefore, these results would be expected if the folding pathways were similar for the three matrices.

3.3.2. Distinctions between energy functions. Another important observation is that the difference in aggregability between runs with M_1 and M_2 is dependent on the number of hydrophobic residues in the sequence. Figure 8 shows a semi-log plot of average aggregability as a function of number of hydrophobic residues for the three matrices. We discarded data from three chains that had four hydrophobic residues each, because they did not aggregate at all with M_1 and therefore could not be plotted on a semi-log plot. We can note that chains with six to 10 hydrophobics were on average more aggregable with M_1 than with M_2, while for any other number, chains were more aggregable with M_2 than with M_1. Because, as stated above, any particular pair of intermediates must be at least as aggregable with M_2 as they are with M_1, this result implies that some chains folded with M_2 favor less aggregable intermediates than they do when folded with M_1. This effect may occur because M_2 will tend to favor conformations that bury hydrophobic residues, even if it is through H-P contacts instead of H-H contacts, thereby lowering aggregability of chains with sufficiently many hydrophilic residues. It is unclear whether the small increase in aggregability for chains with four or five hydrophobic residues folded with M_1 relative to M_2 reflects an actual trend or noise in the data due to the

small number of sequences in those two categories (29 out of 1,424). However, the overall result is significant in that it demonstrates that some folding funnel regions leading to the native may be better protected from aggregation than others, and an accurate model of folding kinetics may therefore be necessary to reliably estimate aggregability.

3.4. Energy landscape

One aspect of folding that is not strictly related to aggregation but can help in interpreting other results is the nature of the energy landscape of folding. Of specific interest is the nature of conformations along the folding pathway that are reachable in a single Monte Carlo step from those on the pathway. The simulation tools were modified to gather statistics on those conformations over the course of the folding process of a protein. The results for five randomly selected native unique 16-mers are shown in Fig. 9. Figure 9A shows the average number of conformations reachable from the current conformation as a function of the number of

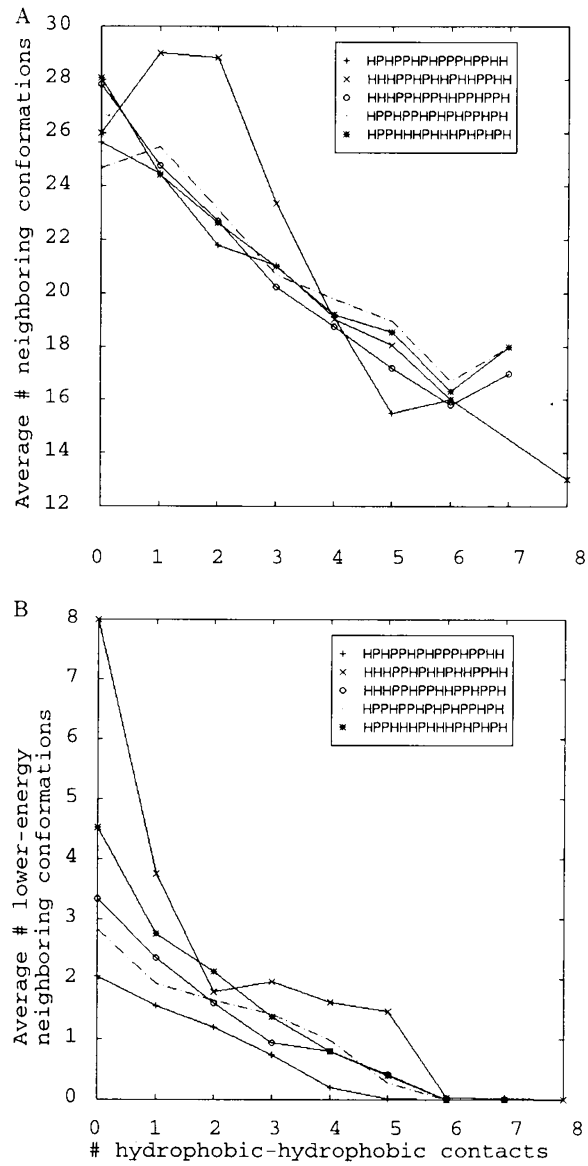

FIG. 9. The nature of the energy landscape in the immediate vicinity of the folding pathway for five randomly chosen native-unique 16-mers. (**A**) The number of valid conformations reachable in one Monte Carlo step from the current conformation, over the course of a folding reaction, as a function of the number of hydrophobic-hydrophobic contacts. (**B**) The number of valid conformations reachable in one Monte Carlo step from the current conformation that have lower energy than the current conformation, over the course of a folding reaction, as a function of the number of hydrophobic-hydrophobic contacts.

hydrophobic-hydrophobic contacts of the current conformation over the course of the folding reaction. The graph shows a notable downward trend, with fewer conformations reachable the lower the energy becomes. One interesting feature is that for four of the five, the number of neighbors noticeably increases between the second lowest and lowest energy states. Figure 9B shows the average number of conformations reachable from the current conformation that contain more hydrophobic-hydrophobic contacts than the current conformations, again as a function of the number of hydrophobic-hydrophobic contacts in the current conformation. This graph shows an even more pronounced trend than Fig. 9A, with numbers of neighbors of lower energy dropping sharply as energy drops.

4. DISCUSSION

A primary conclusion of our experiments is that aggregability is a meaningful and consistent property of protein sequences. We can define a quantifiable measure of aggregability that is reasonably consistent in multiple tests of a single chain and distinguishes between different chains. Furthermore, this quantity is related to measurable sequence properties, suggesting the possibility of developing predictive routines for estimating aggregability where simulation work would require an unreasonable amount of time. However, because some significant variability is introduced by differences in folding time from one trial to another, it may be that aggregability and folding time must be considered together in order to give an accurate picture of aggregation probabilities. We should also note that the property of aggregability appears to occur along a continuum, rather than to be either present or absent. Furthermore, we would expect from laboratory results that whether a chain actually aggregates during synthesis will depend not only on the innate aggregability of the sequence, but also on concentration during the folding process (Anfinsen and Haber, 1961; Kiefhaber *et al.*, 1991), the temperature (King *et al.*, 1996), and the presence of chaperonins (Gordon *et al.*, 1994). It will most likely require a much more sophisticated understanding of the interactions of these and other factors influencing aggregability to reliably predict whether particular sequences will aggregate in significant numbers under a given set of folding conditions.

A related conclusion is that aggregability is primarily a property of kinetic intermediates, rather than native conformations, consistent with the results of experimental work on aggregation (Zettlmeissl *et al.*, 1979). As section 3.2 demonstrates, aggregability of a pair of chains drops considerably when one is frozen in the native state, suggesting that some conformations, including the native, tend to be well-protected from aggregation. Section 3.3 confirms this by showing that changing the energy function in a way that would tend to affect folding pathways can alter aggregability of many sequences in a consistent manner. The evidence that some conformations of a chain are better protected from aggregation than others implies that chains that tend to fold predominantly or exclusively through such "protected" conformations can be expected to have relatively low aggregability. Results from section 3.3 suggest that a sufficiently large ratio of hydrophilic to hydrophobic residues can promote the existence of protected conformations. We can conjecture that other sequence properties that diminish aggregability, such as increasing numbers of hydrophilic blocks, may also favor the emergence of protected conformations.

On the basis of these conclusions, we can propose a model for "protected folding" of globular proteins, in which hydrophilic residues quickly form a barrier around an initially disordered hydrophobic core, allowing the hydrophobic core to fold to its native state while preventing hydrophobic interactions with other chains. This model is illustrated in Fig. 10. Understanding better the nature of protected conformations or how their use as folding intermediates can be encouraged could be an important part of adding knowledge of aggregability to sequence design work.

We can also note that, in our model, aggregability is strongly influenced by folding rate, and that rapid folding can protect against aggregation. This suggests that we cannot understand or predict aggregation without also considering sequence properties that control the rate of folding. It may be possible to incorporate knowledge of aggregability into a more general understanding of protein folding pathways. Efficient folding would require finding energetically favorable pathways that avoid not only standard potential wells in all simultaneously folding proteins, but also the kinetic traps created by aggregability constraints. Results on the nature of the energy landscape suggest that the local regions searched will tend to shrink as a chain gets closer to its native state; this is most likely due to a combination of the collapse of the chain leading to fewer self-avoiding neighbors and the difficulty of improving the energy as the chain approaches its native state. These results would be expected to lead to a more directed search when the conformation is close to the native; this may suggest that proteins need to have relatively large areas of their conformational space protected from

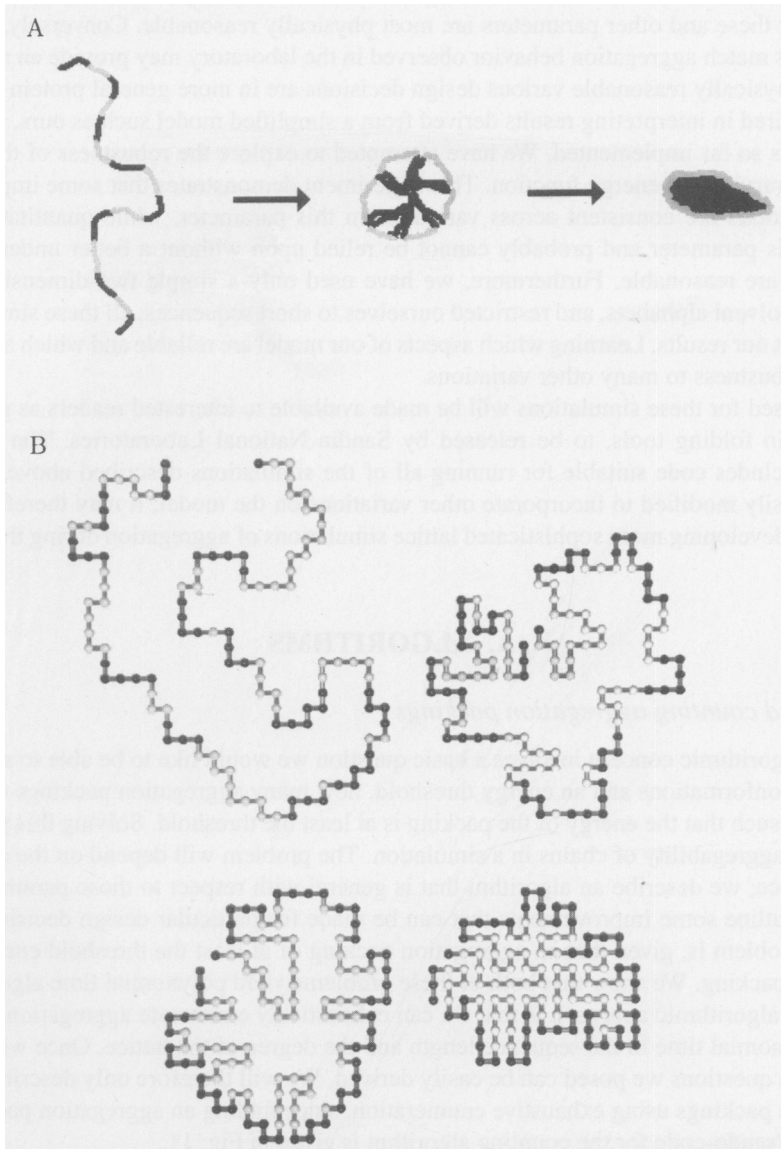

FIG. 10. Proposed model of protected folding. This diagram illustrates a model of protein folding which favors intermediates that are protected from aggregation. (**A**) Abstract model of protected folding. Black regions are hydrophobic, while grey regions are hydrophilic. Folding proceeds from an initially unfolded protein through an early folding intermediate in which a hydrophobic core is shielded by hydrophilic regions, providing protection against aggregation during the folding process, to the final folded form, which is protected from aggregation. (**B**) An illustration of how such a model might appear for the folding process on a lattice.

aggregation early in the folding reaction, but could have only narrow protected regions as they get close to the native. The additional constraints these factors place on the nature of folding could be expected to impose greater restrictions on valid folding pathways than are accounted for by prior lattice folding studies, which did not consider the effects of aggregation.

Finally, we note that considerations of aggregation may place additional constraints on what models can be considered reasonable for studying aggregability and for more general protein folding problems. Our results indicate that aggregation in our model results primarily from the interactions of folding intermediates. It has already been demonstrated that folding intermediates are affected by the choice of move set (Chan and Dill, 1994). It might be possible to develop aggregation estimates independent of the move set by enumerating all possible pairs of conformations that are capable of aggregating to each other; however, this may prove computationally difficult and also will lose information on folding kinetics. Furthermore, as we demonstrate in section 3.3, the choice of energy function can also influence folding intermediates and thereby affect aggregation rates. Developing more realistic simulations of aggregation will therefore require determining

which choices for these and other parameters are most physically reasonable. Conversely, testing how well simulation models match aggregation behavior observed in the laboratory may provide an additional method for testing how physically reasonable various design decisions are in more general protein folding models.

Caution is required in interpreting results derived from a simplified model such as ours, particularly in the very basic variants so far implemented. We have attempted to explore the robustness of the model within a limited scope by varying the energy function. This experiment demonstrates that some important qualitative aspects of simulations are consistent across variations in this parameter, while quantitative data tends to be sensitive to this parameter and probably cannot be relied upon without a better understanding of what parameter values are reasonable. Furthermore, we have used only a simple two-dimensional model, used only HP and HP-solvent alphabets, and restricted ourselves to short sequences; all these simplifications could significantly affect our results. Learning which aspects of our model are reliable and which are not will require exploring their robustness to many other variations.

The software used for these simulations will be made available to interested readers as part of the Tortilla package of protein folding tools, to be released by Sandia National Laboratories. The specific software described here includes code suitable for running all of the simulations described above. In addition, it is designed to be easily modified to incorporate other variations on the model; it may therefore be useful as a starting point for developing more sophisticated lattice simulations of aggregation during the folding process.

A. ALGORITHMS

A.1. Finding and counting aggregation packings

Our primary algorithmic concern involves a basic question we would like to be able to answer efficiently: given two chain conformations and an energy threshold, how many aggregation packings of the two conformations are there such that the energy of the packing is at least the threshold. Solving this problem is critical to measuring the aggregability of chains in a simulation. The problem will depend on the energy matrix and the choice of lattice; we describe an algorithm that is general with respect to those parameters, although in section A.3 we outline some improvements that can be made for particular design decisions. A related but less significant problem is, given that an aggregation packing of at least the threshold energy exists, that of finding one such packing. We show that both of these problems yield polynomial time algorithms.

The key to our algorithmic approach is that we can exhaustively enumerate aggregation packings without repetition in polynomial time in the sequence length and the degree of the lattice. Once we accomplish this, the answers to the questions we posed can be easily derived. We will therefore only describe in detail how to count aggregation packings using exhaustive enumeration, since finding an aggregation packing will then be straightforward. Pseudo-code for the counting algorithm is given in Fig. 11.

The intuition behind the algorithm is to attempt all ways of packing the two chains with at least one connection such that repeats are eliminated. We can assume that chain a is frozen in a particular position and orientation, and we therefore only need to move chain b into different positions and orientations relative to a. The outermost *for* loop defines the orientation of chain b, by iterating over all ways of rotating or flipping the lattice. The second loop picks one residue from the first chain, against which the second will be packed. The third loop iterates over the vectors defining the lattice, i.e., the set of directions from each lattice point to all of its neighboring lattice points; this corresponds to choosing one face of the residue of chain a chosen in the previous *for* loop. The next *for* loop iterates over the residues of chain b, choosing one to pack against the chosen residue of chain a. This specifies both the position and orientation of chain b, at which point the individual residues of b can be positioned one at a time on a's lattice. Since these loops will cover all residues of each chain, and pack each pair of faces of each residue in all possible ways, they will cover every conformation. However, an additional complication is needed to insure conformations are not overcounted. This is the purpose of the matrix M, which records the position of the first residue of chain b for each orientation of b. Since chain a does not move, specifying the orientation of chain b and the position of a single residue completely specifies the position of the entire chain. Therefore, checking for repeated values of the position of a particular residue, for a fixed orientation, eliminates all duplicated packings.

The above algorithm yields a polynomial run-time in all inputs. The outermost loop iterates over rotations of the lattice, which can be bounded by $|L|^2$, since a rotation of the lattice can be specified by anchoring any two lattice edges that are not parallel. The next loop requires $|a|$ iterations, the third $|L|$, and the fourth $|b|$. The code inside these four loops will therefore be run at most $|L|^3|a||b|$ times. Placing chain b in

Given chains a and b, energy threshold T, and a set L of vectors defining the lattice, define a_i to be residue i of a, $\vec{u}_i = (u_{x,i}, u_{y,i}, u_{z,i})$ to be the position of residue i of a, and A_{xyz} to be i if $\vec{u}_i = (x, y, z)$, 0 otherwise. Similarly, define b_i to be residue i of b, $\vec{v}_i = (v_{x,i}, v_{y,i}, v_{z,i})$ to be the position of residue i of b:

define an array of boolean values, M, with the same dimension as the lattice, with size $2\max(|a|, |b|)\max_{\vec{l}\in L}(\|\vec{l}\|_\infty)$ in each dimension; an integer *count*, initially 0; a boolean variable *collision*; and a new coordinate array w of length $|b|$

for each rotation σ of the lattice vectors which preserves the lattice:
  $M_{xyz} \leftarrow false, \forall x, y, z$
  for $i = 1$ to $|a|$:
    for each $\vec{l} \in L$:
      for $j = 1$ to $|b|$:
        collision $\leftarrow$ *false*
        $\vec{w}_j \leftarrow \vec{u}_j + \vec{l}$
        if $A_{w_{x,j}, w_{y,j}, w_{z,j}} \neq 0$ *then collision* $\leftarrow$ *true*
        $k \leftarrow j - 1$
        while $(k \geq 1)$ and $\neg$*collision*:
          $\vec{w}_k \leftarrow \vec{w}_{k+1} + \sigma(\vec{v}_k - \vec{v}_{k+1})$
          if $A_{w_{x,k}, w_{y,k}, w_{z,k}} \neq 0$ *then collision* $\leftarrow$ *true*
          $k \leftarrow k - 1$
        end while
        $k \leftarrow j + 1$
        while $(k \leq |b|)$ and $\neg$*collision*:
          $\vec{w}_k \leftarrow \vec{w}_{k-1} + \sigma(\vec{v}_k - \vec{v}_{k-1})$
          if $A_{w_{x,k}, w_{y,k}, w_{z,k}} \neq 0$ *then collision* $\leftarrow$ *true*
          $k \leftarrow k + 1$
        end while
        if $\neg M_{v_{x,1}, v_{y,1}, v_{z,1}}$
        then $M_{v_{x,1}, v_{y,1}, v_{z,1}} \leftarrow$ *true*
        else if $\neg$*collision* and $(energy(\vec{u} \cup \vec{w}) - energy(\vec{u}) - energy(\vec{v})) \geq T$ *then count* $\leftarrow$ *count* $+ 1$
        end if
      end for
    end for
  end for
end for

FIG. 11. Pseudo-code for counting aggregable packings exceeding a threshold energy.

its new position and orientation requires $O(|b|)$ time, while computing the energy change of the packing requires $O(|L|\min(|a|, |b|))$ time. This accounts for everything except clearing matrix A, which requires $O(\max(|a|, |b|)^2)$ time for a two-dimensional lattice and $O(\max(|a|, |b|)^3)$ time for a three-dimensional lattice, for each of $O(|L|^2)$ iterations. However, we can easily maintain a separate list of elements of A that have been set on each pass through the outer loop, and use that list to clear the array, reducing the total run-time for all calls to this operation to $O(|L|^2|b|)$. Combining all of these factors gives a run-time complexity of $O(|L|^2|b| + |L|^3|a||b|^2 + |L|^4|a||b|\min(|a|, |b|))$. Because the problem is symmetrical with respect to the two chains, we can assume without loss of generality that $|a| \geq |b|$ to give a bound of $O(|L|^4|a||b|^2)$.

The space complexity of the algorithm is also polynomially bounded. The only data structures maintained that are not linear in all parameters are the three arrays A, B, and M. For a two-dimensional lattice, each of these has size $O((\max(|a|, |b|)\max_{\vec{l}\in L}(\|\vec{l}\|_\infty))^2)$, while for three-dimensional lattice, it is $O((\max(|a|, |b|)\max_{\vec{l}\in L}(\|\vec{l}\|_\infty))^3)$. Since there are only a constant number of data structures in use at any time, those values provide a bound on the space complexity of the entire algorithm. Applying the assumption that $|a| \geq |b|$ gives a space bound of $O(|a|^2)$ for two dimensions and $O(|a|^3)$ for three dimensions.

A.2. Reducing memory overhead

Two modifications have been incorporated to reduce the memory usage of the algorithm. Both rely on replacing multi-dimensional arrays with hash tables.

The first modification replaces the array M with a hash table. We can note that while M contains $O(|a|^2)$ elements for a two-dimensional lattice and $O(|a|^3)$ elements for a three-dimensional lattice, we will never use more than $|b|$ elements of either at any one time. Therefore, we can replace M with a hash table of size $O(|b|)$ while still retaining an expected $O(1)$ time to modify or examine elements of the table. The second modification replaces array A with a hash table. As with the previous modification, the hash table need only be proportional in size to $|a|$, rather than to $|a|^2$ for a two-dimensional lattice or $|a|^3$ for a three-dimensional

lattice. This modification will also preserve the expected $O(1)$ time to search or modify a single element of either data structure.

Combining these modifications results in a reduction in space complexity in both the two- and three-dimensional cases. For both cases, we can observe that there are still only a constant number of data structures in use, and none, other than the two arrays we are replacing with hash tables, requires more than $O(|a|)$ space. Therefore, in the two-dimensional case, replacing those arrays with hash tables will reduce the overall space complexity from $O(|a|^2)$ to $O(|a|)$. In the three dimensional case, the reduction will be from $O(|a|^3)$ to $O(|a|)$.

A.3. Heuristics

The run-time of the algorithm might be improved given certain assumptions about the input data or the variant of the model used. Neither of these modifications will give an asymptotic improvement for general data, although they might give noticeable improvements for typical data.

The first heuristic is designed to improve performance in cases where we use an energy threshold test and our energy matrix is negative for hydrophobic-hydrophobic contacts and zero for all others. This condition is valid for most of the experiments described in section 3. The heuristic is based on the observation that aggregation in this model can only occur if at least one hydrophobic-hydrophobic contact is made. Therefore, the *for* loops that iterate over residues of the two chains need only iterate over hydrophobic residues. If we precompute the indices of the hydrophobic residues, then iterate only over those precomputed indices, then we can conceivably speed computations by a factor of nearly four for typical chains (i.e., those with approximately as many hydrophobic residues as hydrophilic) and potentially by much more for chains with few hydrophobic residues, without appreciably increasing run time for any input.

An alternate version of the first modification from section A.2 can improve run-time, but at the cost of greater memory usage. This modification replaces M with a hash table indexed by position and residue number, rather than just position. We then store a value of *true* in element $(\vec{w}_k, k)$ for each $\vec{w}_k$ we assign. This allows us to check for duplicated packings by checking $(\vec{w}_j, j)$, thus avoiding the two innermost *for* loops when there is a duplicate. In order to allow $O(1)$ accesses of the hash table, we must increase its size to $O(|b|^2)$, which could then dominate the overall space complexity. However, the algorithm would discover repeated packings in constant time, rather than the $O(|b|)$ time needed with the algorithm as previously described. This heuristic could therefore reduce run-time considerably when there are many repeated packings. However, it will also add a constant overhead to the innermost *for* loop, and therefore could hurt performance when $|b|$ or the number of repeated packings are small.

ACKNOWLEDGMENTS

This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract no. DE-AC04-94AL85000. We would like to thank Ken Dill for his advice on this work. We also thank the anonymous referees who have reviewed this paper for their detailed comments and suggestions for improvements. Finally, we are grateful to Brian Walenz for his assistance in preparing the graphics.

REFERENCES

Anfinsen, C.B., and Haber, E. 1961. Studies on the reduction and re-formation of protein disulfide bonds. *J. Biol. Chem,* 236, 1361–1363.

Betts, S., Haase-Pettingell, C., and King, J. 1997. Mutational effects on inclusion body formation, 243–264. *In:* Richards, F.M., Eisenberg, D.S., and Kim, P.S., eds., *Advances in Protein Chemistry. Volume 50. Protein Misassembly.* Academic Press, San Diego.

Brems, D.N., Plaisted, S.M., Kauffman, E.W., and Havel, H.A. 1986. Characterization of an associated equillibrium folding intermediate of bovine growth hormone. *Biochemistry* 25, 6539–6543.

Chan, H.S., and Dill, K.A. 1989. Compact polymers. *Macromolecules* 22, 4559–4573.

Chan, H.S., and Dill, K.A. 1991. "Sequence space soup" of proteins and copolymers. *J. Chem. Phys.* 95, 3775–3787.

Chan, H.S., and Dill, K.A. 1993. Energy landscape and the collapse dynamics of homopolymers. *J. Chem. Phys.* 99, 2116–2127.

Chan, H.S., and Dill, K.A. 1994. Transition states and folding dynamics of proteins and heteropolymers. *J. Chem. Phys.* 100, 9238–9257.

Chan, H.S., and Dill, K.A. 1996. Comparing folding codes for proteins and polymers. *Proteins Struct. Funct. Genet.* 24, 335–344.

Chrunyk, B.A., Evans, J., Lillquist, J., Young, P., and Wetzel, R. 1993. Inclusion body formation and protein stability in sequence variants of interleukin-1β. *J. Biol. Chem.* 268, 18053–18061.

Crippen, G.M. 1996. Easily searched protein folding potentials. *J. Mol. Biol.* 260, 467–475.

De Bernardez-Clark, E., and Georgiou, G. 1991. *Protein Refolding*. American Chemical Society, Washington, DC.

Dill, K.A. 1985. Theory for the folding and stability of globular proteins. *Biochemistry* 24, 1501–1509.

Gō, N., and Taketomi, H. 1978. Respective roles of short- and long-range interactions in protein folding. *Proc. Natl. Acad. Sci. U.S.A.* 75, 559–563.

Gordon, C.L., Sather, S.K., Casjens, S., and King, J. 1994. Selective *in vivo* rescue by GroEL/ES of thermolabile folding intermediates to phage P22 structural proteins. *J. Biol. Chem.* 269, 27941–27951.

Haase-Pettingell, C., and King, J. 1988. Formation of aggregates from a thermolabile *in vivo* folding intermediate in P22 tailspike maturation. *J. Biol. Chem.* 263, 4977–4983.

Hart, W., and Istrail, S. 1997a. Lattice and off-latice side chain models of protein folding: linear time structure prediction better than 86, 241–260. *In Proc. of the First Ann. Intl. Conf. on Comp. Mol. Biol., Volume 4(3).* ACM Press, Santa Fe.

Hart, W., and Istrail, S. 1997b. Lattice and off-latice side chain models of protein folding: linear time structure prediction better than 86. *J. Comp. Biol.* 4, 241–260.

Kawasaki, K. 1966a. Diffusion constants near the critical point for time-dependent Ising models (Part i). *Phys. Rev.* 145, 224–230.

Kawasaki, K. 1966b. Diffusion constants near the critical point for time-dependent Ising models (Part ii). *Phys. Rev.* 148, 375–381.

Kawasaki, K. 1966c. Diffusion constants near the critical point for time-dependent Ising models (Part iii). *Phys. Rev.* 150, 285–290.

Kiefhaber, T., Rudolph, R., Kohler, H.-H., and Buchner, J. 1991. Protein aggregation *in vivo* and *in vitro*: a quantitative model of the kinetic competition between folding and aggregation. *Bio/Technology* 9, 825–829.

King, J., Haase-Pettingell, C., Robinson, A.S., Speed, M., and Mitraki, A. 1996. Thermolabile folding intermediates: inclusion body precursors and chaperonin substrates. *F.A.S.E.B. J.* 10, 57–66.

Krueger, J.K., Stock, A.M., Schutt, C.E., and Stock, J.B. 1990. Inclusion bodies from proteins produced at high levels in *Escherichia coli*, 136–142. *In:* Gierasch, L.M., and King, J.A., eds. *Protein Folding*. American Association for the Advancement of Science, Washington, DC.

Lau, K.F., and Dill, K.A. 1989. A lattice statistical mechanics model of the conformational and sequence space of proteins. *Macromolecules* 22, 3986–3997.

Lau, K.F., and Dill, K.A. 1990. Theory for protein mutability and biogenesis. *Proc. Natl. Acad. Sci. U.S.A.* 87, 638–642.

Lim, V.I., and Spirin, A.S. 1985. Stereochemical analysis of ribosomal transpeptidation. *J. Mol. Biol.* 188, 565–577.

Marston, F.A.O. 1986. The purification of eukaryotic polypeptides synthesized in *Escherichia coli. Biochem. J.* 240, 1–12.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087–1092.

Miller, R., Danko, C.A., Fasolka, M.J., and Balazs, A.C. 1992. Folding kinetics of proteins and copolymers. *J. Chem. Phys.* 9, 768–780.

Mitraki, A., Haase-Pettingell, C., and King, J. 1991. Mechanisms of inclusion body formation, 35–49. *In:* Georgiou, G., and De Bernardez-Clark, E., eds. *Protein Refolding*. American Chemical Society, Washington, DC.

Patro, S.Y., and Przybycien, T.M. 1994. Simulations of kinetically irreversible protein aggregate structure. *Biophys. J.* 66, 1274–1289.

Patro, S.Y., and Przybycien, T.M. 1996. Simulations of reversible protein aggregate and crystal structure. *Biophys. J.* 70, 2888–2902.

Šali, A., Shakhnovich, E., and Karplus, M. 1994a. How does a protein fold? *Nature* 369, 248–251.

Šali, A., Shakhnovich, E., and Karplus, M. 1994b. Kinetics of protein folding. *J. Mol. Biol.* 235, 1614–1636.

Skolnick, J., and Kolinski, A. 1991. Dynamic monte carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *J. Mol. Biol.* 221, 499–531.

Speed, M.A., King, J., and Wang, D.I.C. 1996. Polymerization mechanism of polypeptide chain aggregation. *Biotechnol. Bioeng.* 54, 333–343.

Sun, S., Brem, R., Chan, H.S., and Dill, K.A. 1995. Designing amino acid sequences to fold with good hydrophobic cores. *Protein Eng.* 8, 1205–1213.

Taketomi, H., Ueda, Y., and Gō, N. 1975. Studies on protein folding, unfolding and fluctuations by computer simulation. *Int. J. Pept. Protein Res.* 7, 445–459.

Tanaka, S., and Scheraga, H.A. 1975. Model of protein folding: inclusion of short-, medium-, and long-range interactions. *Proc. Natl. Acad. Sci. U.S.A.* 72, 3802–3806.

Thomas, G., Becka, R., Sargent, D., Yu, M.-H., and King, J. 1990. Conformational stability of P22 tailspike proteins carrying temperature-sensitive folding mutations. *Biochemistry* 29, 4181–4187.

Thomas, P.D., and Dill, K.A. 1996. Statistical potentials extracted from protein structures: how accurate are they? *J. Mol. Biol.* 257, 457–469.

Tiktopulo, E.I., Bychkova, V.E., Rička, J., and Ptitsyn, O.B. 1994. Cooperativity of the coil-globule transition in a homopolymer: microcalorimetric study of poly(*N*-isopropylacrylamide). *Macromolecules* 27, 2879–2882.

Wetzel, R., ed. 1997. *Advances in Protein Chemistry. Volume 50. Protein Misassembly.* Academic Press, San Diego.

Zettlmeissl, G., Rudolph, R., and Jaenicke, R. 1979. Reconstitution of lactic dehydrogenase. non-covalent aggregation vs. reactivation. I. physical properties and kinetics of aggregation. *Biochemistry* 18, 5567–5571.

Ziff, R.M. 1984. Aggregation kinetics via Smoluchowski's equation. 191–199. *In:* Family, F., and Landau, D.P., eds. *Proceedings of the International Topical Conference on Kinetics of Aggregation and Gelation.* Elsevier Science Publishers, Athena, GA.

Address reprint requests to:
S. Istrail
Department of Algorithms and Discrete Mathematics
Sandia National Laboratories
Albuquerque, NM 87185

E-mail: scistra@cs.sandia.gov