Vijay S. Pande[1–4]
Ian Baker[1]
Jarrod Chapman[4,*]
Sidney P. Elmer[1]
Siraj Khaliq[5]
Stefan M. Larson[2]
Young Min Rhee[1]
Michael R. Shirts[1]
Christopher D. Snow[2]
Eric J. Sorin[1]
Bojan Zagrovic[2]

Atomistic Protein Folding Simulations on the Submillisecond Time Scale Using Worldwide Distributed Computing

[1] *Department of Chemistry, Stanford University, Stanford, CA 94305-5080*

[2] *Biophysics Program, Stanford University, Stanford, CA 94305-5080*

[3] *Department of Structural Biology, Stanford University, Stanford, CA 94305-5080*

[4] *Stanford Synchrotron Radiation Laboratory, Stanford University, Stanford, CA 94305-5080*

[5] *Department of Computer Science, Stanford University, Stanford, CA 94305-5080*

Received 27 March 2002; accepted 22 May 2002

Abstract: *Atomistic simulations of protein folding have the potential to be a great complement to experimental studies, but have been severely limited by the time scales accessible with current computer hardware and algorithms. By employing a worldwide distributed computing network of tens of thousands of PCs and algorithms designed to efficiently utilize this new many-processor, highly heterogeneous, loosely coupled distributed computing paradigm, we have been able to simulate hundreds of microseconds of atomistic molecular dynamics. This has allowed us to directly*

Correspondence to: Vijay S. Pande; email: pande@stanford.edu

*Present address: Physics Department, University of California, Berkeley, CA

Contact grant sponsor: ACS PRF, NSF MRSEC CPIMA, NIH BISTI, ARO, and Stanford University

Contact grant numbers: 36028-AC4 (ACS PRF), DMR-9808677 (NSF MRSEC CPIMA), IP20 6M64782-01 (NIH BISTI), 41778-LS-RIP (ARO)

Biopolymers, Vol. 68, 91–109 (2003)

simulate the folding mechanism and to accurately predict the folding rate of several fast-folding proteins and polymers, including a nonbiological helix, polypeptide α-helices, a β-hairpin, and a three-helix bundle protein from the villin headpiece. Our results demonstrate that one can reach the time scales needed to simulate fast folding using distributed computing, and that potential sets used to describe interatomic interactions are sufficiently accurate to reach the folded state with experimentally validated rates, at least for small proteins. Biopolymers 68: 91–109, 2003

Keywords: *atomistic protein folding; microsecond time scale; computer hardware; computer algorithms; molecular dynamics; distributed computing; villin; beta hairpin*

INTRODUCTION

Understanding the sequence–structure relationship of proteins will play a pivotal role in the postgenomic era, and will have great impact on genetics, biochemistry, and pharmaceutical chemistry.[1–3] A detailed picture of the folding process itself will be important in understanding diseases, such as Alzheimer's and variant Creutzfeldt–Jacob disease, believed to be related to protein misfolding.[4] Finally, an understanding of protein folding mechanisms will be important in protein design and nanotechnology, in which self-assembling nanomachines may be designed using synthetic polymers with protein-like folding properties.[5]

Unfortunately, current computational techniques to tackle protein folding simulations are fundamentally limited by the long time scales (from a simulation point of view) needed to study the dynamics of interest. For example, while the fastest proteins fold on the order of tens of microseconds, current single computer processors can only simulate on the order of a nanosecond of real-time folding in full atomic detail per CPU day—a 10,000-fold-computational gap. Great strides in traditional parallel molecular dynamics (MD), utilizing many processors to speed a single dynamics simulation, have been made and have partially overcome this divide. A tour-de-force parallelization of simulation code for supercomputers by Duan and Kollman has previously led to the simulation of 1 μs of dynamics for the villin headpiece three-helix bundle,[6] demonstrating that parallelization schemes using hundreds of processors can be used to make significant progress at closing this computational gap. However, such methods have fundamental drawbacks: in particular, these methods require complex, expensive supercomputers due to the need for fast communication between processors. Moreover, due to the stochastic nature of folding, in order to study the folding of a 10-μs folder, one must simulate hundreds of microseconds, requiring computing power equal to thousands or tens of thousands of today's processors.

Developing such large-scale parallelization methods is very difficult, and current parallelization schemes cannot scale to the level of even thousands of processors (i.e., cannot use so many processors efficiently). To understand why scalability to thousands of processors is so difficult, consider an analogy to a graduate student thesis. A typical thesis takes 1500 graduate student days. If one employed 1500 graduate students to accomplish this goal, would it be possible to complete a thesis in a single day? Clearly not—the overhead of communication between students, as well as the inability to devise an "algorithm" to divide the labor evenly, would make 100% efficiency impossible in this case. At this level of scaling, it is likely that the work would actually take *longer.* These issues, in particular balancing communication time against time spent actually doing work, are mirrored in the division of labor between computer processors. Clearly, the only way to efficiently utilize such a large number of processors is to divide work in such a way that requires minimal communication.

Even with an algorithm with *perfect* scalability (e.g., with a 10,000-fold increase in speed using 10,000 processors), we are still left with the problem of obtaining a 10,000-processor supercomputer. For comparison, the largest unclassified supercomputer in the world (the SP at NERSC) has 2500 processors, and of course this resource must be shared between many (hundreds) different research groups. Recently, another approach has been developed to bridge this enormous computational gap: worldwide distributed computing.[7] There are hundreds of millions of idle PCs potentially available for use at any given time, the majority of which are vastly underused. These computers could be used to form the most powerful supercomputer on the planet by several orders of magnitude.[7] However, to tap into this resource efficiently and productively, we must employ nontraditional parallelization techniques.[8,9] Indeed, we wish to accomplish a seemingly impossible goal: to push the scalability of MD simulations to previously unattainable levels (i.e., the efficient use of tens of thousands

of processors) using an extremely heterogeneous network of processors that are loosely coupled by relatively low-tech networking (primarily modems).

In this review, we first present the details of our method to simulate protein folding using distributed computing, and then summarize our folding simulation results for several small, fast folding proteins and polymers. Specifically, we demonstrate the applicability of our method by simulating the folding of protein helices[10,11] in atomic detail. Next, as an additional quantitative test of our methodology, we examine the folding rate and folding time distribution of a nonbiological helix,[12] for which results of traditional MD simulations[13] and experiments[14] are known. We finally apply these methods to larger and more complex proteins, including a β-hairpin[15,16] and a three-helix bundle.[6] We conclude with an assessment of the validity of our methods including a quantitative comparison of these results with experimental measurements of folding rates and equilibrium constants, and a discussion of what we have learned about folding mechanisms.

METHODS

Why Is the Dynamics of Complex Systems so Slow and How Can This Be Circumvented?

The dynamics of complex systems typically involves the crossing of free energy barriers.[17] It has been demonstrated[18,19] that free energy barrier crossing dynamics, such as protein folding, does not make steady, gradual progress from one state to another (such as dynamics from the unfolded to folded states during a folding transition), but rather spends most of the trajectory time dwelling in a free energy minimum, "waiting" for thermal fluctuations to push the system over a free energy barrier. Indeed, this process is *dominated* by the waiting time, and the time to cross the free energy barrier is in fact much shorter than the overall folding time, typically by several orders of magnitude.[20] This opens the door to the possibility that one may simulate complex processes, such as folding, using trajectories much shorter than the folding time[20] (i.e., using nanosecond simulations to reproduce kinetics with microsecond rate constants).

Methods to exploit this observation have been previously developed (the most notable being path sampling, in which one simulates the paths over the barrier, rather than the time spent waiting in the original free energy minimum), with promising results.[20–22] However, several technical complications have limited the use of these methods in simulating protein folding and in making quantitative comparisons to experiment. First, some of these methods require that the path in question not dwell in metastable states and thus may get stuck in local meta-stable free energy minima along the pathway (which have been found in many folding and unfolding simulations[19]). Second, these methods require knowledge of the native state as an end goal and in a sense apply a field to the trajectories to reach this native state. Finally, in the end, the heart of the protein folding problem lies in *sampling,* and even with the great benefits of path sampling methodology, a tremendous degree of sampling (in this case, in path space) must still be performed.

Thus, the main goal of the method we have developed is to simulate folding dynamics starting purely from the protein sequence and an atomistic force field, without using any knowledge of the native state in the folding simulation. It is important that our method can successfully tackle the issue of lingering in metastable free energy minima. To achieve this, we use the following algorithm ("ensemble dynamics"). Consider running M independent simulations started from a given initial condition (each run starts from the same coordinates, but with different velocities). We next wait for the *first* simulation to cross the free energy barrier (see Figure 1). Since the average time for the first of M simulations to cross a single barrier is M times less than the average time for all the simulations (assuming an exponential distribution of barrier crossing times[8,9]; see below for details), we can use M processors to effectively achieve an M times speedup of a dynamical simulation, thus avoiding the waiting in free energy minima. In a sense, we *distribute* the waiting to each processor in parallel, rather than in series, as in traditional parallel MD. Given the ability to identify individual barrier crossings, one can then speed the *entire* (multiple barrier) problem by turning it into a series of single barrier problems, restarting the processors from the new free energy minima after each barrier crossing (see below and Refs. 8 and 9). Also, it can be shown that one need not use identical computers for these calculations, an important fact in employing heterogeneous public clusters.[8,9]

To more quantitatively see how one can use these simulations to examine events that occur on considerably longer time scales, consider a protein with single exponential kinetics, where the fraction that fold in time t is given by

$$f(t) = 1 - \exp(-k\,t)$$

where k is the folding rate. On average, a folding event will occur on the $1/k$ time scale. However, we expect to see some folding events even at short times compared to the folding rate, i.e., when kt is small. In this case, we have $f(t) \approx kt$. How many folding events would we expect to see? Consider studying a protein that folds with $k = 1/10{,}000$ ns, given $M = 10{,}000$ simulations each of length $t = 30$ ns we would expect to see $M\,f(t) \approx M\,k\,t = 30$ folding events.

The above discussion shows how one can speed dynamics by a factor of M for a single barrier system, but what about multiple barrier problems? To handle multiple barriers, we suggest a scheme similar to Voter's parallel replica method.[9] This method is summarized in Figure 1. We start M simulations from a single initial condition, and then wait

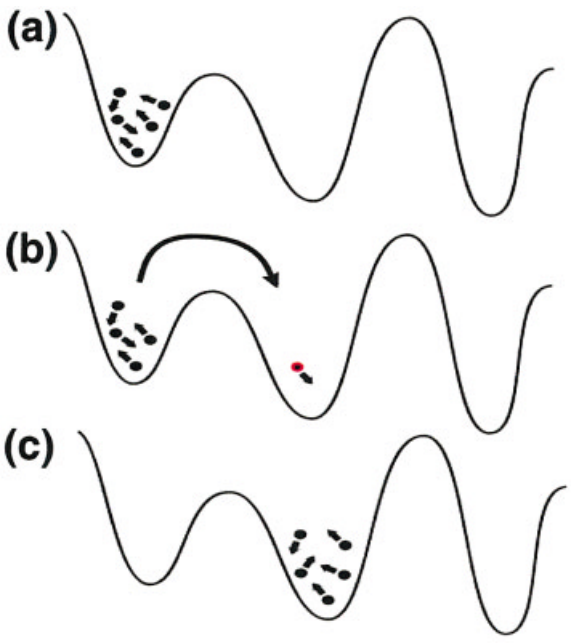


FIGURE 1 *Simulating dynamical events using worldwide distributed computing.* Traditional methods utilize multiple processors to speed a single dynamics calculation. We suggest that multiple processors can be used to generate sets of calculations, and that the desired thermodynamic or kinetic observables can be calculated from such an ensemble. This method does not require supercomputers and can run well on massively parallel clusters.[7–9] Consider a multiple barrier-crossing problem (a model of many complex phenomena). Since the bulk of simulation time is spent waiting for thermal fluctuations to bring the system over the barriers, one can speed the calculation by starting many simulations in the first free energy minimum (a), and waiting until just *one* of them has crossed. At this point, we couple the simulations by placing them all in the same place in the configuration space as the simulation that has crossed that barrier (b). This process is then repeated as many times as needed to cross additional free energy barriers (c). One can show (see text) that this algorithm, with M processors, is equivalent to a single processor system running M times faster.[8,9] Thus, with hundreds to thousands of processors and assuming that one can identify transitions, we would be able to bridge the computational barriers currently limiting protein folding and reach well into the microsecond time-scale.

for the first simulation to cross a free energy barrier. Once this simulation has crossed over to the next free energy minimum, we restart all other simulations from that new location in configuration space, restart the dynamics, and wait for another replica to cross another free energy barrier. Since we employ stochastic dynamics, even though all simulations are restarted from the same configuration, they quickly decorrelate from each other and explore different regions of the phase space.

Quantitative Rate Prediction

Below, we show how this scheme can be used to predict folding rates. This result stems from the fact that the distribution of barrier crossing times for the first-to-cross is directly linked to the distribution of usual barrier crossing times. To demonstrate this, we again assume single exponential kinetics (deviations from single exponential kinetics have also been examined elsewhere[8]). For a single processor, we expect that a particular simulation would have crossed the barrier by time t with a probability

$$P_1(t) = k \exp(-k\,t)$$

For the M simulation case, the probability that the *first* simulation has crossed in time t is

$$P_{\mathrm{M}}(t) = [\,k \exp(-k\,t)]\,M\,[1 - k \int_0^t \exp(-k\,t)dt]^{M-1}$$

(i.e., the probability that one simulation has crossed, times a degeneracy factor of M, times the probability that the remaining $M - 1$ simulations have not folded). Evaluation of the integral above yields

$$P_M(t) = M\,k \exp[-M\,k\,t]$$

which is exactly the *same distribution* as the single processor case, except with an effective rate which is M times faster. Since this method simply speeds the effective rate of crossing each barrier, one can use the number of processors M and the rate for first crossing to predict the experimental rate.

The error inherent in the above procedure for calculating the rate and the time constant can be estimated in the following way. As shown above, for a given N_{total} and a given t, the rate k is simply proportional to the number of molecules that have folded by time t, $N_{\mathrm{folded}}(t)$. Since each folding process behaves probabilistically (according to an exponential distribution) and given fixed t and N_{total}, the number of processes that will fold by time t, $N_{\mathrm{folded}}(t)$, will be a random variable. In other words, different realizations of the "large experiment" containing N_{total} individual processes will, by their very nature, yield different values of $N_{\mathrm{folded}}(t)$ for a fixed time t. From this it follows that our rate estimate will also be associated with a certain inherent uncertainity. From elementary probability theory, we know that the number of folding events by time t, $N_{\mathrm{folded}}(t)$, given a constant rate, will be distributed according to the Poisson distribution. This in turn means that the rate estimate, which is proportional to $N_{\mathrm{folded}}(t)$, will also be distributed according to the Poisson distribution. The standard deviation of a Poisson distribution with rate λ is equal to $\lambda^{1/2}$, meaning that our rate estimate $\pm$ standard deviation will simply be

$$k = N_{\mathrm{folded}}/(N_{\mathrm{total}}t) \pm {N_{\mathrm{folded}}}^{1/2}/(N_{\mathrm{total}}t)$$

Standard propagation of error results in time constant ± standard deviation of

$$\tau = 1/k = (N_{\text{total}}t)/N_{\text{folded}} \pm (N_{\text{total}}t)/N_{\text{folded}}^{3/2}$$

For example, for the β-hairpin folding data (see below), we have $N_{\text{total}} = 2700$, $N_{\text{folded}} = 8$, and $t = 14$ ns, which results in $k = 2.1 \times 10^5 \pm 0.74 \times 10^4\ \text{s}^{-1}$, and $\tau = 4.7 \pm 1.7\ \mu\text{s}$.

We stress that as long as one can identify transitions, thus allowing an M times speed-up for all barrier crossings, the dynamics we simulate will faithfully follow the dynamics one would obtain from traditional MD, but simply M times faster. If there are off-pathway traps, our method will go to them; indeed, we will reach them M times faster. However, we will escape these traps M times faster as well. This method is not intended as a structure prediction algorithm, but rather a means to speed dynamics and study the mechanism of folding, which may include on-pathway intermediates or diversions to traps.

How Can One Identify Free Energy Barrier Crossings ("Transitions")?

Of course, the utility of this method rests on our ability to identify transitions, i.e., to calculate whether a simulation has crossed a free energy barrier. Voter's parallel replica method was intended to accelerate the dynamics of solid-state systems that have *energy* barriers, in which one can identify new states by performing energy minimization to see whether one has crossed an energy barrier.[9] However, in protein folding (as well as many other complex systems), the relevant barriers are *free energy* barriers, and thus an energy minimization technique is not applicable. In order for this method to be applied to a broad range of barrier crossing problems, one needs to use a more general way to identify free energy barrier crossings.

We suggest that, in analogy to first-order phase transitions, one could look for a large variance in energy, which can loosely be related to a momentary surge in the heat capacity (a common sign of a first-order phase transition). Such energy variance peaks have been seen to coincide with free energy barrier crossings in simple models[23] and all-atom (S. Perkins and V. Pande, unpublished results) models of protein folding. This technique has the significant advantage that it does not require any knowledge of the structure of the protein at the barrier.[24] Moreover, to the extent that energy variance peaks correctly identify transitions, these peaks would aid in the interpretation of the simulation results, since they would demarcate transitions to new free energy minima.

Of course, in the case of single-exponential kinetics, as is experimentally observed for almost all small proteins, there exists only one rate-determining free energy barrier, and thus recognizing the barrier is not essential to the technique. In fact, although we do not discuss it in this paper, in some cases ignoring the transitions can result in reaching the folded state faster than by recognizing all the barriers.[8] Finally, simulating completely independent trajectories is another appealing possibility for systems with single exponential kinetics; we discuss this possibility in the Discussion section below.

Simulation Details

For all of the molecules presented here, each simulation is started from a completely extended state. This is done to avoid any possibility of biasing the initial state toward the native state of the molecule. Clearly the extended state does not represent the structure of the unfolded state. Indeed, we find that rapidly—i.e., within 1–3 ns of MD simulation—this extended state relaxes to the unfolded state of the protein. While this practice utilizes more computational time than, for example, starting from some predicted unfolded state, it has the virtue of not making any assumptions of the unfolded ensemble and removes any possibility of biasing the system to the native state.

For each run, we have used M "clone" processors, each simulating folding in atomic detail with molecular or stochastic dynamics simulations (Figure 1). Once one of these clones makes a transition (identified by a spike in the energy variance: see below), we declare that the simulation has gone through a transition, copy the resulting configuration to *all* of the other processors, and recommence simulations from the new configuration. After restarting all simulations from the coordinates of the barrier-crossing simulation, one must ensure decorrelation of the next ensemble of trajectories in order to achieve an increase in computational speed. This process is performed many times, over several "runs."

In our simulations, the spatial coordinates of the barrier-crossing simulation were copied and unique random number seeds (for Langevin dynamics random forces) were used to immediately differentiate the simulations. In a purely deterministic simulation, one would need to differentiate each simulation by restarting them with differing velocities, which may lead to potentially nonphysical discontinuities in the path; however, if the velocity decorrelation time is much shorter than the conformational decorrelation time (certainly true for dynamics in any water-like solvent), then the effects are likely to be minimal. In either case, the path obtained would correspond to a fast traversal of the potential landscape, but the total simulation time among all processors would be equivalent to the additional time waiting in minima that a representative "serial" simulation would take.

The Folding@Home distributed computing system (http://folding.stanford.edu) was used for the two most demanding calculations (the β-hairpin and villin simulations) presented here. The Folding@Home client software (which performs the scientific calculations) is based upon the Tinker molecular dynamics code,[25] with numerous modifications performed by Michael Shirts, other members of the Pande group, Jed Pitera, and Bill Swope. We simulated folding and unfolding at 300 K and at pH $\sim$ 7 (unless noted otherwise), using the OPLS[26] parameter set and the GB/SA[27] implicit solvent model. Stochastic dynamics were used to simulate the viscous drag of water ($\gamma = 91$/ps), and a 2 fs integration time step was used with the RATTLE

algorithm[28] to maintain bond lengths. Long-range interactions were truncated using 16 Å cutoffs and 12 Å tapers.

We identified transitions by a heat capacity spike associated with crossing a free energy barrier. It has been previously shown that this is a means to identify transitions in all-atom[8] and simplified protein model[29] simulations. To monitor the heat capacity during the simulation, we calculate the energy variance, and use the thermodynamic relationship $C_v = (E^2 - E^2)/T$, where E is the energy of the system; note that since we are using an implicit solvent, our "energy" is often called an "internal free energy" (the total free energy except for protein conformational entropy). Each PC runs a 100 ps MD simulation ("1 generation"), calculates the energy variance within this time period, and then returns this data to the Folding@Home server. If the energy variance exceeds a preset threshold value, the server identifies this trajectory as having gone through a transition, and then resets all other processors to the newly reported coordinates. Since the heat capacity is extensive, we used a fixed value of this threshold per atom (0.8 $kcal^2/mol^2$/atom) for all molecules (which for example leads to a threshold of ~300 $kcal^2/mol^2$ for villin).

Since transitions occur relatively infrequently (see below), one need not run these simulations on massively parallel supercomputers (with high speed communication); instead, these simulations are well suited for large, decentralized distributed computing clusters, such as the Folding@Home project. Not only is this a demonstration that such distributed computing clusters can be used to study long time-scale kinetics with molecular dynamics—we stress that this is likely the *only* way such calculations could have been practically performed, considering the great computational demands of these calculations.

RESULTS

Protein α-Helices

To test the methodology presented above, we have simulated the folding of two different α-helical peptides. One sequence we examined, the "Fs peptide" Ac–$A_5(A_3RA)_3A$–NH_2, has been shown experimentally to have biexponential kinetics, with characteristic times of 10 ns and 160 ± 60 ns.[10] Helices are believed to form via nucleation,[30,31] which is influenced by the disorder in a system (either as a nucleation accelerator or blocker), analogous to a liquid with impurities. In our system, the arginine residues could be considered to be an analog of these impurities that blocks propagation, and it is interesting to consider the role of this disorder in the sequence above, i.e., whether the arginine residues affect the nucleation processes. To address this, we have also folded a pure poly-A chain, Ac–A_{20}–NH_2.

We have been able to fold both of these protein sequences and find rates comparable to experiment at a temperature of 10°C. The initial configurations were completely elongated chains ($\phi = -135°$, $\psi = 135°$). Qualitatively, both the poly-A helix and the Fs peptide folded by first undergoing nucleation followed by propagation toward the termini. We found propagation in both directions (N to C and C to N), although we do not have sufficient statistics to determine a bias in propagation direction.[30] Quantitatively, the Fs peptide folded (i.e., reached 15 helical residues, the value expected from experiment[10]) in 82 ± 60 ns in our simulations. Note that while 7 runs were used for this average, 2 of the 7 Fs peptide runs did not fold after 160 ns. Since these runs were included in the average as folding in 160 ns, the average is somewhat lower than it should be. However, since the experimental rate is 1/160 ns, one would expect that on average 4.4/7 [$63\% = 1 - \exp(-kt)$] of the runs would fold after 160 ns, whereas we observed 5/7, which is well within the experimental bounds.[11]

Moreover, our simulations capture some finer detail about the nature of folding. We see fast early events, as found experimentally. The alanine-rich N-terminal part of the Fs peptide folded very quickly, in 15 ± 10 *ns,* consistent with the observation of N-terminal fluorophore quenching in 10 ns by Eaton and co-workers[11] and the faster rate observed by Dyer and co-workers.[10] The poly-A helix folded considerably faster (18 ± 8 ns, out of 8 runs) and typically had more helical content (17.8 residues vs 15.1 for the Fs peptide).

Apparently, the arginine residues are responsible for the differences in these folding rates by acting as blockers of helix nucleation and propagation. Looking at the formation of secondary structure vs. time (see Figure 2, right), we see that helical propagation halts at the arginine residues (R) and often the completion of helix formation requires additional nucleation events. While our eight poly-A helix runs did show stalling of propagation (Figure 2), these events were not localized to any particular point in the chain. Why does arginine limit propagation? We suggest that the long Arg side chain significantly limits its mobility, and moving into a helical ϕ/ψ orientation thus occurs much more slowly.

Nonbiological Helices

How generally applicable and accurate is the coupled simulation method? To address this question, we have applied this method to study the folding of a nonbiological helix, a 12-mer of polyphenylacetylene (PPA).[12] This polymer can be considered to be a nonbiological analog of polyalanine, since it is a homopolymer with a simple side chain that folds into a

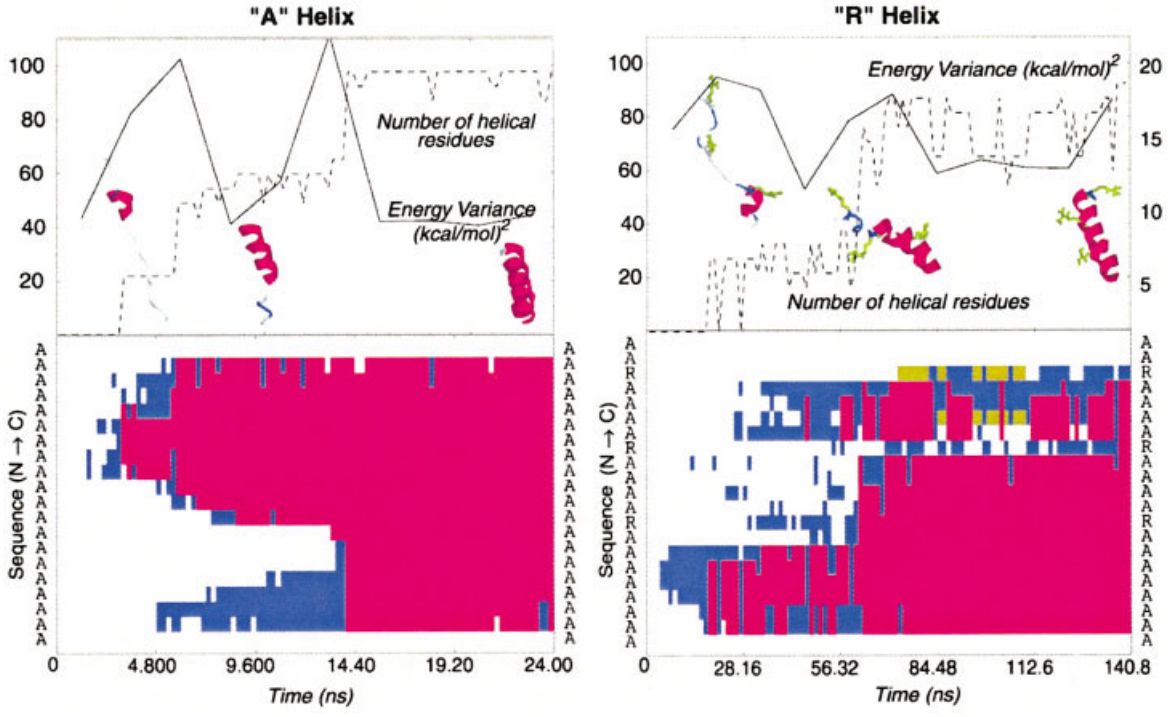


FIGURE 2 *Folding simulation of α-helices.* Shown above are trajectory data for simulations of the poly-A helix (left) and Fs peptide (right). *Top*: number of helical units vs time (dotted line) and energy variance vs time. We see that peaks are associated with nucleation events. *Bottom*: Secondary structure formation vs time: red, yellow, and blue denote helices, β-sheets, and turns respectively. In both cases, we see nucleation events (corresponding to energy variance peaks). However, in the case of the Fs peptide, nucleation events did not occur at the arginine residues (R) and propagation typically was blocked at these residues (also seen in the other seven runs we performed, data not shown). We estimated the time by multiplying the directly simulated time t by the number of processors M ($M = 24$ and $M = 128$ for the left and right trajectories, respectively).

helix.[12] We have previously shown[13] that this polymer folds to a helix on the tens of nanosecond timescale, in accordance with previous experimental observations.[14] We find that our ensemble dynamics method works well for PPA. The mean folding time and folding time distribution are consistent with brute force, traditional simulations of PPA.[13] This is demonstrated by the agreement in mean folding times between the two methods and the similarity of the folding time distribution (see Figure 3 and Table 1).

C-Terminal β-Hairpin of Protein G

α-Helices and β-hairpins together represent the most ubiquitous secondary structural elements in proteins. In a previous section, we discussed our simulation results for helices and now we concentrate on hairpins. We have recently reported a full-atom, implicit-solvent simulation of folding of the hairpin at a biologically relevant temperature,[32] and here we briefly summarize those results. We have obtained a very large ensemble of conformations, which includes mostly partially folded structures, as well as eight complete, fully independent folding trajectories. These data sets allow us to determine the key trends characterizing the folding process and determine several average properties that have been measured or could, in principle, be measured experimentally.

Based on our results, we can estimate the folding rate of the hairpin in the following way: we have simulated 27 independent runs, each consisting of $M = 100$ clone simulations that, on average, completed approximately 14 ns of simulated time, bringing the total to approximately 38 μs of real time simulation. Out of 2700 simulations, we have detected eight complete folding events, which (if we assume single exponential folding kinetics) results in an estimated folding time of approximately 4.7 μs. This prediction is in excellent agreement with the experimentally measured time of 6 μs.[16,32]

Our results offer the following picture of the folding mechanism (Figure 4). Folding from a fully ex-

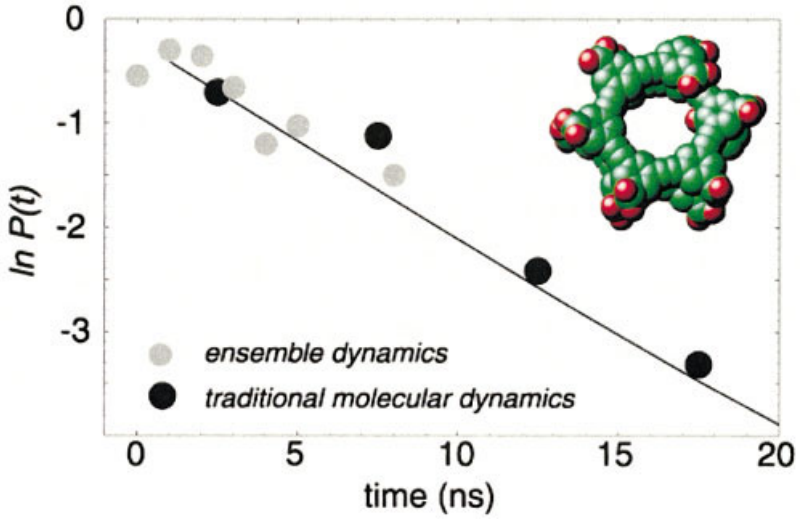


FIGURE 3 *Quantitative validation of our method.* We plot the folding time distribution for a 12-mer PPA helix calculated from the ensemble dynamics (gray) and traditional molecular dynamics (black) methods. We find excellent agreement with both simulation and with experiment (which finds a characteristic time of ≈10 ns). We calculated the folding time as Mt, where t is the simulation time of the individual trajectory and $M = 20$ processors were used. The quantitative agreement here shows that one can indeed achieve a linear speed-up using 20 processors for the 12-mer PPA folding problem. *Inset:* a folded PPA 12-mer.

tended conformation begins with a rapid collapse to a more compact structure. During this time, various temporary hydrogen bonds form, condensing the peptide and decreasing the costly loop entropy that hampers the formation of the hydrophobic core. These temporary hydrogen bonds form and break; their pattern, which varies from run to run, has no resemblance to the final hydrogen-bonding pattern of the hairpin. Next, an interaction between the hydrophobic core residues is established. This is clearly the central event in the folding process and most probably its rate-limiting step. Note that at this point the core is still not fully formed: the initial hydrophobic interaction most often involves just two hydrophobic residues on the opposite sides of the future hairpin. Full formation of the core typically appears simultaneously with the establishment of final hydrogen bonds.

This pathway was also suggested by several other simulation methods. Pande and Rokhsar reported the

Table I Summary of Predicted vs Experimentally Measured Folding Times[a]

Protein/Molecule	Predicted time (ns)	Experimental time (ns)	Experiment reference
Polyphenylacetylene (PPA)	5.3[b]	10	14
Fs peptide [Ac–$A_5(A_3RA)_3A$–NH_2]	127[c]	160 ± 60	10, 11
C-terminal β-hairpin of protein G (Ac–GEWTYDDATKTFTVT-ENH_2)	4700 ± 1700	6000	15, 16
Villin headpiece	20,000[d]	11,111	37, 38

[a] We see a very strong correlation between our prediction and experiment. For a direct correlation, we find $R^2 = 0.993$, p value = 0.008, and for a correlation of the log of these times, to match Figure 8, we find $R^2 = 0.993$, p value = 0.000026.

[b] PPA folds with nonexponential behavior.

These numbers report the fast time in a double exponential fit.

[c] If average the folding times of the runs that folded, we get 82 ± 60 ns. However, if we include the data for the runs that did *not* fold, we see that 5/7 folded in 160 ns; therefore using $5/7 = 1 - \exp(-kt)$ leads to a time of $1/k = 127$ ns.

[d] This number represents an estimate based on one folding event, and therefore has a large error and thus is likely reliable solely as an order of magnitude prediction.

results of high temperature unfolding and refolding of the β-hairpin, in which a discrete unfolding pathway was recognized to include a hydrophobically stabilized intermediate ("H" state) with only the Val54 side chain being released from the core and little hydrogen bonding occurring.[19] Karplus and co-workers used multicanonical Monte Carlo simulations to look at folding of the hairpin with similar results.[33] Garcia and Sanbonmatsu[35] and Berne and co-workers[34] later verified the existence of this intermediate through a temperature-exchange Monte Carlo/molecular dynamics hybrid model of unfolding in which the thermodynamics of the unfolding events are well described.[35] They note that these intermediates appear with, on average, 2 fewer hydrogen bonds than the folded hairpin. This "H" intermediate was then observed in mechanically driven unfolding simulations performed by Bryant et al.,[36] who describe it as including a nearly assembled core and very little backbone hydrogen bonding.

The picture of the folding process that emerges is, in essence, a blend of the hydrogen-bond-centric and the hydrophobic-core-centric views of hairpin folding: nonspecific hydrogen bonds are important in the initial stages of folding, but the key event that stabilizes the U-shaped precursor of the hairpin and guides the downstream folding process is the formation of a hydrophobic interaction between core residues. Final hydrogen bonds appear later, around the same time the full formation of the hydrophobic core occurs, and these continue to fluctuate even after folding is complete.

Villin Headpiece

We have also simulated the folding of a thermostable, fast folding,[37] 36-residue α-helical subdomain (pdb-code 1VII) from the villin headpiece[38,6] (the C-terminal domain of the much larger villin actin binding protein). Figure 5 details the nature of this folding trajectory. We start from a completely elongated structure and then see rapid relaxation into a random-walk unfolded state ("U"). Next (Figure 5a), the C-terminal helix forms very quickly (at the tenth generation, "G10" $\approx$250 ns = $M\,t = 250 \times 10$ generations $\times$ 0.1 ns/generation; see Methods for details). This time is consistent with helical folding times found experimentally[39] and in simulation.[8] The protein then collapses, driven by the attraction of its hydrophobic groups. While many residues have native-like secondary structure (see Figure 5b), there is a large degree of non-native side-chain interaction, such as the contact of TRP24 and PHE36 with hydrophobic core residues, although they are solvent exposed in the native structure. This intermediate collapsed thermodynamic state ("I") consists of an ensemble of many conformations with partial native secondary structure, but confounded by a lack of native side-chain packing.

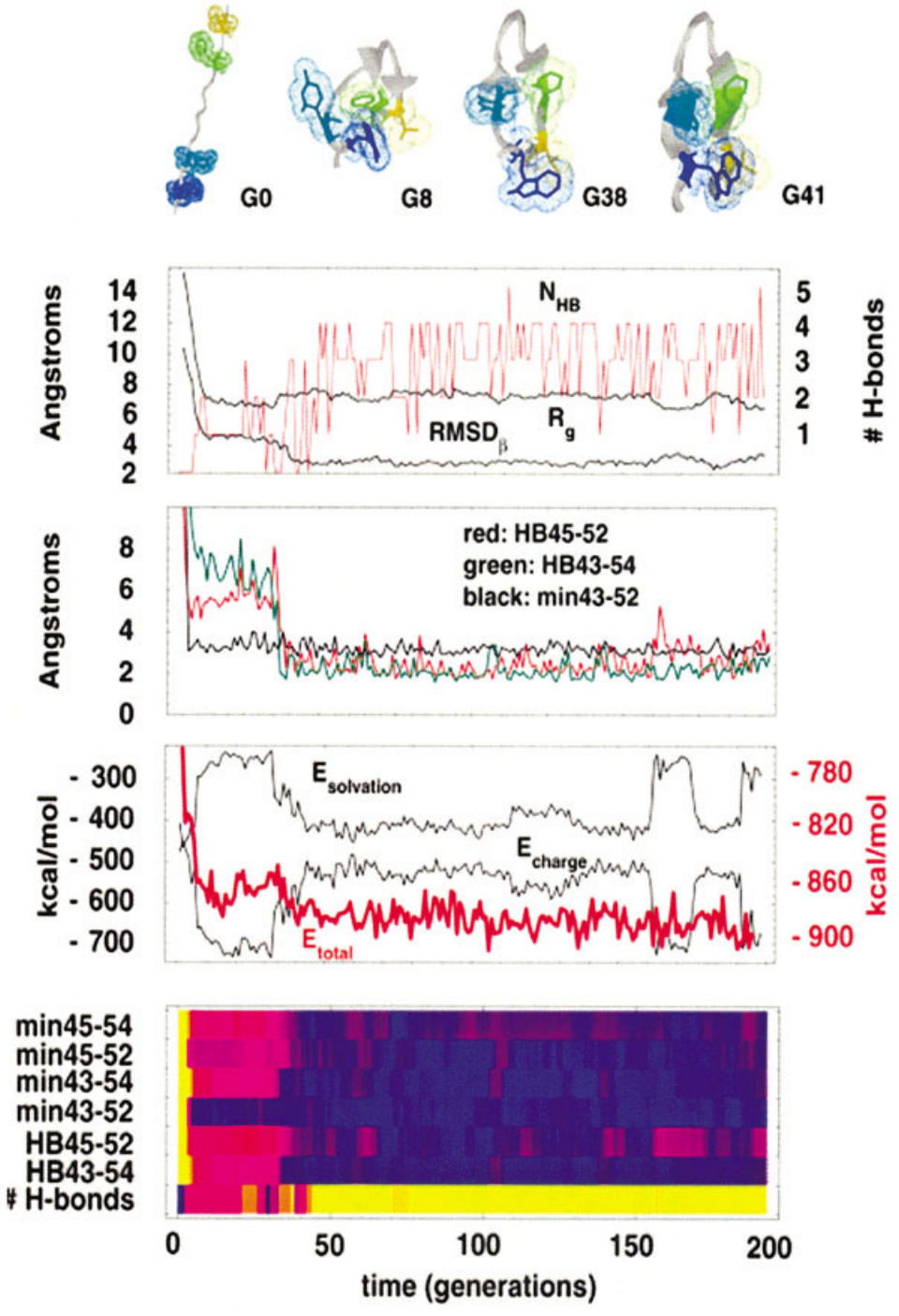
The protein remains in this state for a very long time (the equivalent of $\approx$3.2 μs) until a thermal fluctuation occurs which breaks key non-native interactions that were preventing the formation of the hydrophobic core. Once these non-native contacts are broken, the protein rapidly folds to its native state ("N"). Looking at this transition in more detail (Figure 6), we see that in order to break non-native contacts (such as, but not limited to, the interaction between PHE11 and PHE36 in G200), the protein expands, breaking many contacts (G210), and then collapses into its native fold (G225), as identified by a root mean square deviation (RMSD) similar to that found by exploring the native state in our unfolding simulation (i.e., 3–4 Å; see below). This event occurs after the equivalent of 5.5 μs, which is within the time estimated experimentally (on the order of 10 μs).[37] Since PHE36 forms non-native (misfolded) contacts in this intermediate state (as well as the intermediate found in the Kollman simulation[6]), we predict that removing this bulky hydrophobic side chain would likely increase the folding rate.

We have performed four other coupled simulations that have also each reached the 5 μs time scale (data not shown). All of these trajectories have reached "I" (RMSD between 5 and 7 Å, radius of gyration R_g between 7 and 10 Å), but none have reached the N state. Statistically, this is not surprising and can be used to estimate the folding rate (see Methods, above): if the mean folding time for villin were 20 μs and it follows exponential kinetics, then one would expect that 20% of runs would fold in 5 μs, in agreement with our results.

We have also used Folding@Home to study the native state of villin, i.e., by starting simulations from the NMR structure.[38] One use of such simulations is the determination of the variability of conformations in the native state ensemble. Moreover, since our method allows us to simulate events that would occur on the microsecond time scale, we should also be able to simulate villin unfolding under experimental conditions (e.g., 300 K). We see (Figure 7) that conformations within the native state typically have a 3–4 Å RMSD from the NMR structure. Thus, we identify our "N" state with the native state of this protein since our folding simulation reaches an ensemble of conformations with $\approx$4 Å RMSD (our conformation from the folding run, which was most similar to the NMR structure, had a 3.3 Å RMSD). Moreover, our native state simulation was run long enough to explore un-

folding to the intermediate state: the protein did not completely unfold during the simulation time scale (~1 μs). However, a transition to the partially folded intermediate (I) was detected.

Finally, we compare our results to what one might expect from protein folding theory. One of our primary results is that folding appears to proceed through transitions between free energy minima: starting in an unfolded state (U) to an intermediate (I) and then to the native state (N) (e.g., see reviews Refs. 1–3 and 40, and references therein). As previously discussed,[1–3] the collapse to the I state appears to be driven by hydrophobic interactions. However, considering that villin is one of the fastest folding proteins currently known, it is interesting to consider that many of these interactions were *non-native.*



While there is little structure in U, the intermediate I is collapsed, with some native tertiary structure, much non-native structure, and little side-chain packing. Thus, I is very much like the molten globule intermediates found in other proteins.[41] Also, the intermediate state found by Duan and Kollman[6] fits our I state, since it is collapsed, partly native, but missing certain native contacts, and satisfies our I state definition in terms of the RMSD and radius of gyration. We see a somewhat cooperative U $\rightarrow$ I transition (e.g., reflected in free energy barriers in Figure 7a) and a very cooperative I $\rightarrow$ N transition, as predicted previously.[42,43] Indeed, the cooperativity of the I $\rightarrow$ N transition appears to result from side chain packing in our model, since many non-native contacts must collectively break to allow the formation of native side-chain packing. It seems highly unlikely that the native structure could be reached so rapidly through piecewise movements without this collective event.

It is clear that any potential set employed to model atomic interactions will have its limitations. The relevant question to ask is, How good do they need to be and what would result from errors in these potentials? Since we do see folding to the native state, it appears that the potentials we used were sufficient in this case. However, we cannot rule out the possibility that in our model, the I state is comparably (or more) stable than the N state. This could be the result of slight errors in the potentials.[44] Much like adding denaturant in a physical example, adding errors to a potential reduces the energetic favorability of the N state and thus makes the I state relatively more favorable due to its entropic advantage.[44]

DISCUSSION

Scalability of the Algorithm

As more computers become accessible to distributed computing methods, it is important to understand the limits of the scalability of the method, i.e., the limits to the number of processors one can use to achieve a speed increase. While this method can yield significant speed improvements for simulating complex systems (and scalability considerably beyond traditional parallel MD), there are some important limitations to its scalability we must consider. For example, simulating a process where the mean time to fold $t_{\text{fold}} = 100$ ns using $M = 10^6$ processors will not necessarily mean that one will achieve folding events using only $t_{\text{fold}}/M = 100$ fs trajectories. The scalability will be inherently limited by the barrier crossing time t_{cross} (i.e., the time spent actually crossing the barrier, not including the much longer time spent "waiting" in the free energy minima, which dominates the folding time t_{fold}). Since the speed increase from our method is due to the elimination of the waiting time, we expect that $M > t_{\text{fold}}/t_{\text{cross}}$ additional processors will not give any additional speed increase.[8,9] Thus, the bounds of scalability for this method are also related to an interesting physical question: How much time is required to actually cross the free energy barrier? This time can be quantified by using our method to look for the limits of scalability within our technique.

For the proteins we have examined, it is likely that this time is on the hundreds of picoseconds to nanosecond time scale. It is interesting to consider how

FIGURE 4 *A detailed analysis of a folding trajectory of the β-hairpin from the C-terminal segment of protein G.* (a) Cartoon representation of the folding trajectory; the backbone of the peptide is represented as a gray trace; the core hydrophobic residues (Trp43, Tyr45, Phe52, Val54) are shown in dot representation; (b) RMSD from the 1GB1 structure of the hairpin (residues 43–54), radius of gyration, and the number of backbone–backbone hydrogen bonds; (c) distance between key hydrogen bonding partners (green: Trp43–Val54; red: Tyr45–Phe52), and the minimum distance between Trp43 and Phe52 (black). Note that the minimum distance between Trp43 and Phe52 reaches its final value before the key hydrogen bonds are established; (d) solvation energy ($E_{\text{solvation}}$), charge–charge energy (E_{charge}), and total potential energy vs time. The initial hydrophobic collapse of the unfolded peptide correlates with a sharp decrease in E_{total}, while the attainment of the final structure correlates with E_{total} reaching its final value. A significant deviation around G160 of E_{charge} and $E_{\text{solvation}}$ from their final value is correlated with the temporary breaking of the key Tyr45–Phe52 hydrogen bond; (e) a concise summary of the key events along the folding trajectory (color code: yellow—high; violet—low). HB-ij denotes the distance between the hydrogen bonding partners i and j; min-kl denotes the minimum distance between residues k and l. Note that the establishment of the hydrophobic Trp43–Phe52 interaction is the earliest event of significance along the trajectory. Time is reported in the number of generations: roughly, 1 generation corresponds to 100 processors $\times$ 0.1 ns/generation/processor = 10 ns/generation.

this minimum time varies with the folding time. Since there need not be any correlation between these times, it is possible that slower folding proteins (e.g., those which fold on the millisecond and longer time scale) could be folded using our method with current microprocessors by simply employing more of them. Indeed, computational resources on the million-processor scale have been proposed, such as IBM's Blue Gene, as have other distributed computing projects. With such computational resources, it is possible that we could push our simulations from the hundreds of microsecond timescale to fractions of a second, allowing us to reach timescales relevant for slow folders.

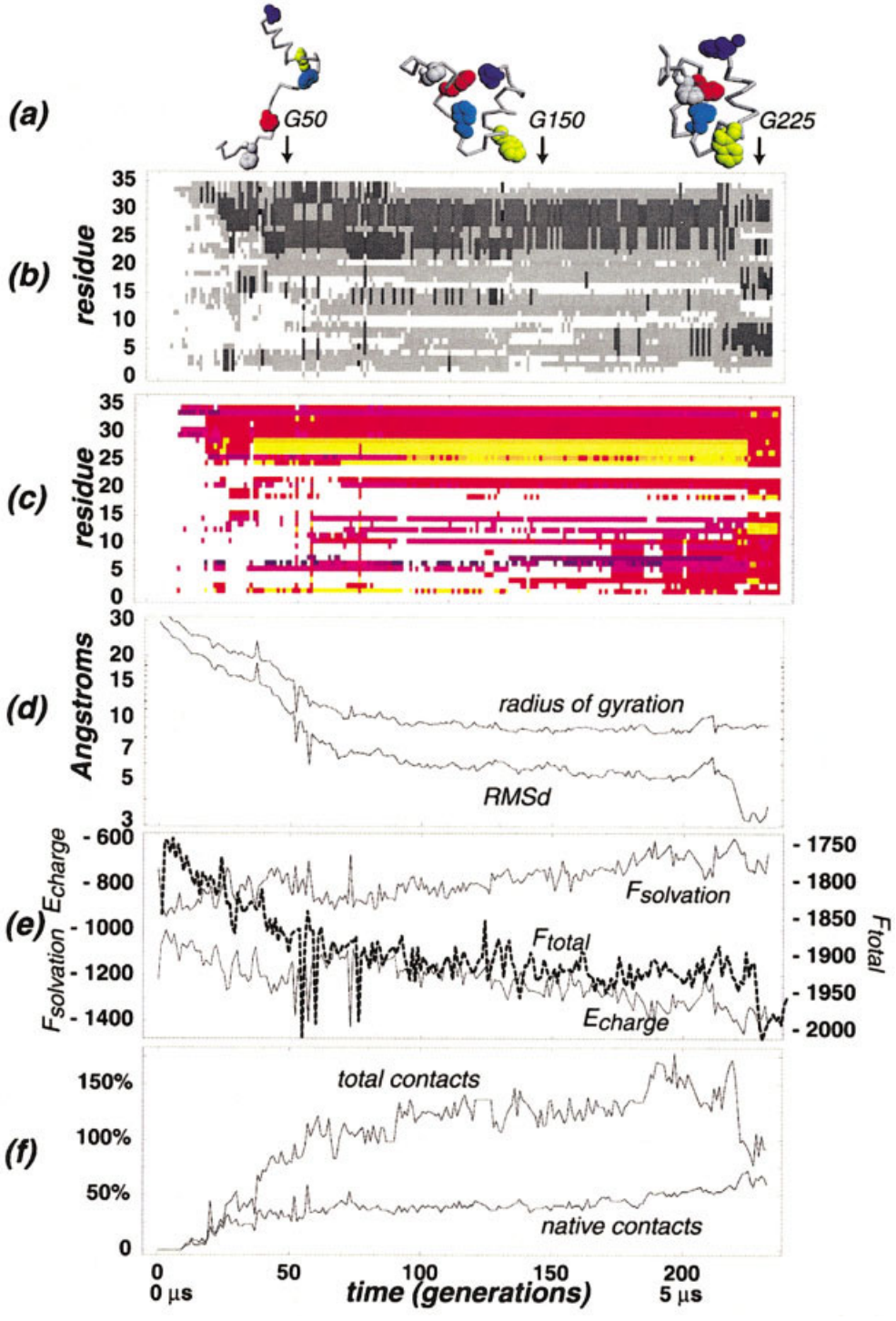


FIGURE 6 *Examination of the I to N transition of the villin headpiece in detail.* We see that in order to correctly fold, the protein must first unfold and open its conformation in order for it to form the missing native state interactions. Visualization is the same as in Figure 2a. See text for more details. The final state agrees reasonably well with the average refined NMR conformation from the Protein Data Base.[38]

Limitations of Our Methods to Predict Folding Rates and Mechanisms

Below, we summarize the approximations involved in our rate determination method above, and our justifications and reasoning of these approximations. First, we assume that the barrier crossing probability density is exponential. This does not mean that the total kinetics is single exponential, but that the time to cross an individual barrier is exponential. Second, we assume that transitions are correctly identified—i.e., that there are no false positives or false negatives. This second assumption is of greater concern. While we cannot know for certain that we are correctly identifying transitions, our ability to predict rates suggests that incorrect transition prediction is not an issue. (Perhaps this may be due to the fact that transitions were rarely detected in the larger proteins studied and were typically found at the beginning of the folding trajectories—see the section *Is Transition Detection Really Necessary?* below). However, we can mathematically address the consequences of incorrect transition detection and nonexponential kinetics, as we have done in a previous work.[8] Finally, in our error analysis above, we can calculate the statistical uncertainty of our rates. Even with only tens of successful folding trajectories, the statistical uncertainty is negligible.

Accuracy of Implicit Solvent Models for Protein Folding

For all of the folding simulations presented here, we have used the GB/SA method.[27] While *GB/SA* makes connections to physical arguments about the nature of interactions via internal vs external dielectrics, it is an empirical theory. Nevertheless, GB/SA performs very well at predicting the solvation free energy of small molecules,[27] and it is perhaps not surprising that it appears to be sufficiently accurate in the prediction of the folding rate of small proteins and peptides. Moreover, Caflisch and co-workers have also had successful results using even simpler implicit solvation mod-

FIGURE 5 *Anatomy of a folding trajectory of the villin headpiece.* (a) Significant representative conformations along the trajectory. The protein is visualized as a backbone trace with the aromatic residues (PHE7, PHE11, PHE18, TRP24, PHE36) space filled and colored gray, red, cyan, yellow, and blue respectively. (b) Secondary structure from DSSP[51] (black = helix, gray = turn, white = no structure). (c) Native contact density for each residue (blue = low, red = middle, yellow = high). (d) Radius of gyration and RMSD from the native state (the native state is defined from the average of a 10 ns traditional MD simulation at 300 K starting from the NMR structure[38]) are plotted; we use only α-carbons in this calculation and omit the first and last 2 residues in the RMS calculation (as they are unstructured in the refined NMR conformation). (e) Solvation free energy ($F_{\text{solvation}}$), charge/charge energy (E_{charge}), and total internal free energy (F_{total}) vs time. While F_{total} gradually decreases over the whole simulation, we see that $F_{\text{solvation}}$ has an initial decrease, but then gradually increases over the simulation, whereas E_{charge} consistently decreases. In fact, E_{charge} and $F_{\text{solvation}}$ are highly correlated ($R^2 = 0.92$) during this trajectory. (f) Fraction of all and native contacts vs time. In all frames, time is on the horizontal axis. It is most natural to report time in terms of 100 ps "generations" (see Figure 1); roughly, one can approximate time as[8] 250 processors $\times$ 0.1 ns/generation/processor = 25 ns/generation. We label conformations by their generation (e.g., "G225" in the upper right).

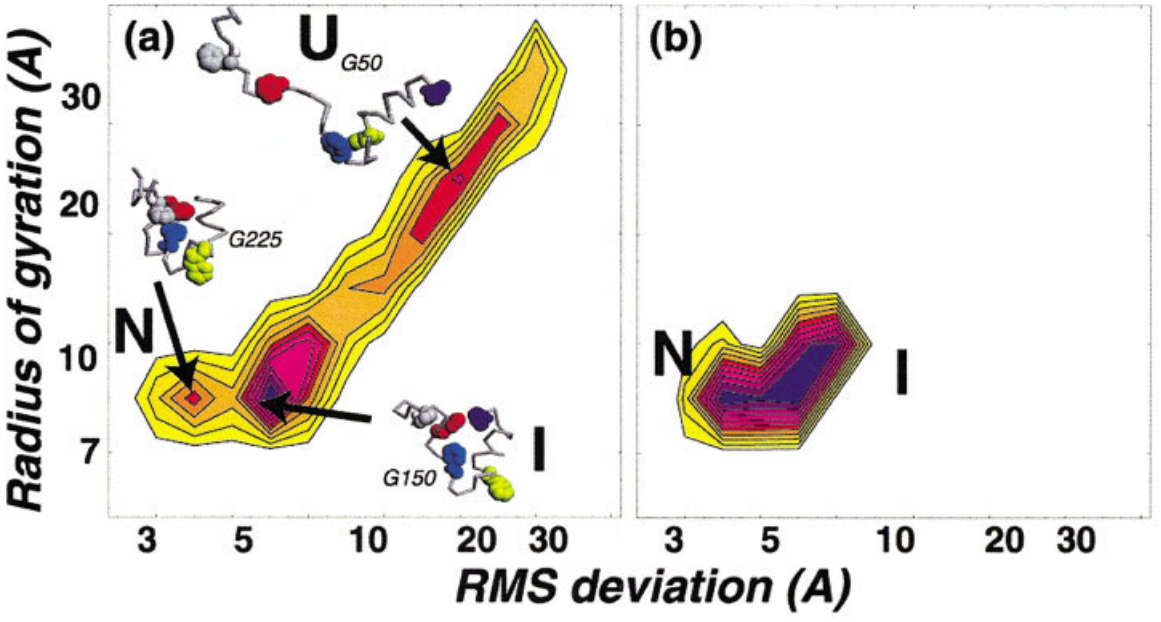


FIGURE 7 *Rough characterization of the underlying free energy landscape for the villin headpiece.* We plot the log of the probability of finding conformations with a given R_g and RMSD in (a) folding and (b) unfolding simulations. We find three distinct probability maxima (which correspond to free energy minima): an unfolded state, molten-globule-like intermediate, and the native state. This landscape generated from kinetic data qualitatively agrees with previous, more extensive thermodynamic calculations.[2]

els (distance-dependent dielectric with a surface area term following Still *et al.*[27]) in folding simulations.[31,45] However, it is unclear whether similar accuracy (in either rate prediction or even in reaching the native state) would be achieved using implicit solvation models for folding simulations of larger proteins.

Second, we stress that when employing any implicit solvent model for the faithful reproduction of kinetics, one must take into account the viscosity of the solvent (in addition to the dielectric and hydrophobic aspects). We have done so using Allen's stochastic integrator, as implemented in Tinker.[25] This scheme is an extension to Langevin dynamics, and includes both viscous drag and random forces in the force equation, to match the viscosity of the solvent and the random thermal fluctuations that the solvent would apply to the solute. However, unlike pure Langevin dynamics, this method scales the drag and the random force by the solvent-exposed area in order to only apply these solvation effects to atoms that are actually solvent exposed. Often, implicit solvation is run *without* any viscosity model (or viscosity considerably lower than water,[45] i.e., the viscosity parameter $\gamma \ll 90$/ps), which leads to differences in sampling and cannot lead to accurate rate predictions. This is, in some cases, considered to be an *advantage* of implicit solvation: one would expect that the speed of dynamics is inversely proportional to viscosity. However, since the magnitude of the random forces is also proportional to the viscosity, decreasing the viscosity diminishes the strength of these random forces. Counterintuitively, this may actually *decrease* the sampling as it is these very random forces that enable the system to cross free energy barriers.[18] Since our goal is the faithful reproduction of folding kinetics, we have chosen a viscosity damping parameter in order to match that of water.[46]

Third, it is interesting to consider the possible differences between implicit and explicit solvation models. While implicit solvation models can capture many important properties of the solvent, such as the dielectric effect, hydrophobicity, and viscosity, there are effects that are missing. In particular, any physical effect that arises from the *discrete* nature of water molecules, such as proteins hydrogen bonding to water, solvent-separated minima, or the drying effect, will be lost. Again, it is important to keep in mind that all models are approximations, and the relevant question is not whether a model is "correct" (since all models are incorrect at some level), but whether a given model is "correct enough" to capture the rele-

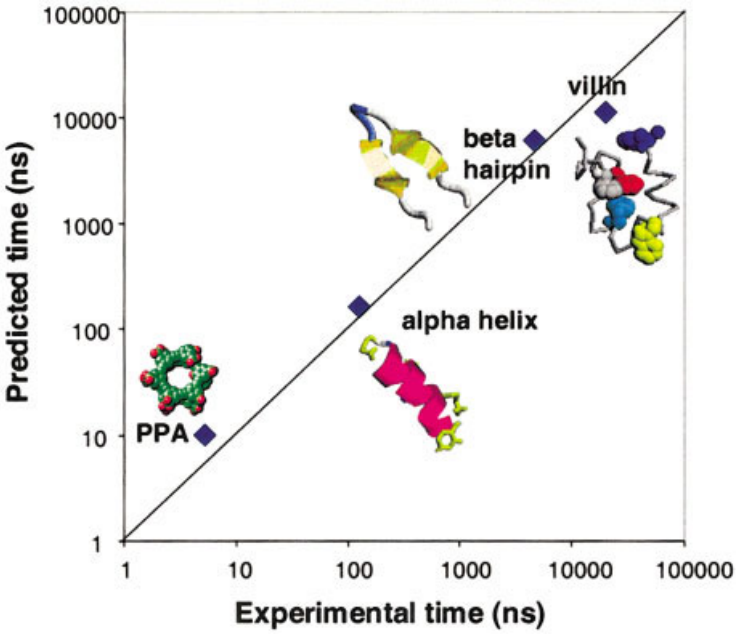


FIGURE 8 *Comparison of theoretical rate predictions from @Folding@Home and the according experimental folding rate determinations.* We compare the folding rates for the proteins and polymers described in this review: PPA, polyalanine-based helices, the C-terminal β-hairpin from protein G, and the villin headpiece. If our folding rate prediction were perfect, all points would lie on the diagonal line. The agreement strongly suggests that our method can accurately predict the absolute folding rate for small proteins, peptides, and foldamers.

vant physics to faithfully reproduce and predict the physical effect of interest. For the small proteins we have examined, it appears that the model we have employed is indeed "correct enough" for predicting rates (see Figure 8 and Table I). This implies that either the discrete nature of water is not relevant for folding, that folding rates are fairly robust to such inaccuracies of the model, or that there is a convenient cancellation of errors. In order to discern between these two possibilities, one must resimulate these proteins with explicit solvation models and compare the rate and mechanistic predictions; if these predictions agree, then perhaps the potential gain in accuracy of explicit solvent models would indeed not be relevant for folding kinetics. Furthermore, we stress that explicit solvent models make approximations as well,[47] and there is no reason why an arbitrary explicit solvation model would necessarily be better than a well-designed implicit model.

Finally, it is important to consider that the question of the validity of implicit solvation models goes beyond a simple debate of the validity of particular computational methodology, but also impacts the way in which one thinks of protein structure in general. If explicit solvation were critical to protein folding, then it is likely that one should not think of protein structures without the requisite cloud of water molecules it interacts with, as it is the very discrete and potentially structural aspects of the water that play a large role in folding. However, if implicit solvent models are sufficiently accurate, this suggests that a structural picture of a protein alone (*implicitly* considering the effects of water, such as hydrophobicity, etc., but not with a discrete, structural form in mind) is indeed sufficient.

Alternative Methods to Simulate Dynamical Events on Long Time Scales Using Low Viscosity Simulation

Water is a relatively viscous solvent. Indeed, in quantitative terms, the damping force of water is on the order of $\gamma = 100$/ps. It is intriguing to consider whether one can simulate the effective result of long time-scale events by simulating the effect of much lower viscosity solvents, say $\gamma = 1$/ps, while keeping all of the other properties of a water-implicit solvation model unchanged. This is appealing since this is trivial to perform with implicit solvation models and this ability to explicitly set the viscosity of the solvent in the model may represent one of the great strengths of using implicit solvent models.

If one were to decrease the viscosity by 100 times, could one simulate 10 ns and expect to get 1000 ns = 1 μs of sampling? This question has been addressed in many models and systems. For example, Klimov and Thirumalai[48] have shown that (for a coarse-grained protein model) the rate of folding increases with decreasing viscosity to a point ($\gamma \sim$ 1/ps) at which the rate decreases with decreasing viscosity. This nonmonotonic dependence can be understood in terms of the dual role of the solvent viscosity: viscosity retards motion through the solvent, but also creates the random forces that are needed to drive the system over energy and free energy barriers. Thus, the rate should be optimal at intermediate viscosity. If this peak in the rate vs viscosity curve does peak at $\gamma \sim$ 1/ps, then one should expect an increase in sampling at viscosities in between 1/ps and 100/ps, and for thermodynamic properties, this increased sampling should be beneficial. However, it is still unclear whether kinetic properties would be unchanged by significant changes in the viscosity.

Alternative Methods to Simulate Dynamical Events on Long Time Scales Using Large-Scale Distributed Computing

In this review, we have discussed protein folding simulations using ensemble dynamics, our parallel replica-like method intended to handle free energy barrier crossing problems. The greatest weakness of this method rests in the need for transitions and for the accurate identification of these transitions. Our suggested means to identify transitions, looking for energy variance spikes during dynamics, has the benefit of being a purely thermodynamic method and thus does not use any information of the protein native state or any folding-related hypothesized reaction coordinates. However, if transitions are incorrectly identified, the validity of the resulting data is put under question. Considering that great computational resources are needed to generate the folding simulations presented here, this limitation could be very expensive computationally—the failure to accurately identify transitions may mean that the resulting data set is invalid.

It is interesting to consider a simpler method, which does not have the liabilities described above. Namely, instead of loosely coupling simulations (i.e., restarting simulations after transitions have been detected), one could simply run a large number M of completely *independent* simulations. For single exponential kinetics, we would still gain an M times speedup (as described above). However, even if the reaction under study did not have single exponential kinetics, independent trajectories might still have value. Indeed, in a sense, a set of thousands of simulations each on the tens of nanosecond time scale is a data set that stands on its own. For example, one could interpret the results for single-exponential kinetics, by examining the fraction $f(t)$ that fold in time t and fitting a rate with the slope. However, one would not be limited to this exponential kinetics analysis, and this data could be reanalyzed a *postiori* to test new hypotheses or kinetic models. Considering the great computational cost of producing these data sets, this more "pure" method for simulating kinetics has a great appeal. Indeed, we have reexamined the folding of villin with uncoupled trajectories[49] in this manner (B. Zagrovic, et al. J Mol Biol, 2002, in press) and it will be interesting to determine how the uncoupled simulations differ (e.g., in rates and mechanism) from those presented here.

Is Transition Detection Really Necessary?

The discussion above regarding the possibility of using independent trajectories and still gaining a speed increase linear with the number of processors raises the question, Must one bother with transition detection as used here? Another way to examine this question is to ask with what frequency were transitions detected in the examples described here. For the helix folding simulations, 2 or 3 transitions were detected before the simulation reached the folded state. The first transition accompanied the first formation of helical structure and the other transitions occurred during propagation. For the larger molecules, transitions were even more infrequent. For example, the β-hairpin and villin simulations typically had a single transition that occurred early in the folding process, accompanying the collapse of the protein chain.

Thus, we find that for the larger molecules, transition detection was likely not necessary, since the transitions occurred earlier and thus the simulations were essentially running independently (as suggested in the subsection above). We suggest that the transitions were not needed since these larger molecules fold with single exponential kinetics, and thus have a single rate-limiting step. The helices are potentially different: the rates of nucleation and propagation of helices in our model are not highly separated (e.g., see the trajectories in Figure 2) and thus transition detection may be needed in the helix case, but not for the β-hairpin or villin molecules.

CONCLUSIONS

Comparison to Experiment

With a wide range of molecules under study, from the nonbiological PPA helices to the 36-residue villin headpiece, we have simulated a set of molecules with a range of folding times spanning over four orders of magnitude, from nanoseconds to tens of microseconds. Since the primary means of comparison to experiment is the comparison of rates determined by simulation and experiment, we concentrate on our prediction of rates. Figure 8 shows a striking agreement between predicted and experimental rates (see Table I for details). Of course, with just four molecules simulated, it is unclear whether this agreement is simply fortuitous. In order to more fully address this question, we plan to simulate the folding kinetics of additional molecules, including larger and more slowly folding proteins. Indeed, more recent work on a small $\beta\beta\alpha$-fold (C. Snow et al., Nature, 2002, in press) and villin (B. Zagrovic et al., J Mol Biol, 2002, in press) also result in strong agreement with experimental rates.

With this quantitative agreement with experiment, it is also interesting to ask how do our results reflect upon the quality of modern force fields? On the surface, one might conclude that our agreement with experiment is evidence that force fields are sufficiently accurate. We stress that the only question that can truly be addressed by our work is whether force fields are sufficiently accurate to reproduce experimental rates and structures. Ignoring for the moment the possibility that the agreement may be fortuitous, the agreement between our simulations and experiments suggest that force fields are sufficiently accurate to predict the folding rates of small proteins. Indeed, this accuracy can be quantified in terms of the strong correlation ($R^2 = 0.996$) and low p value (0.000026) of the logarithms of the predicted to experimental rates. However, this statement should definitely not be overgeneralized—it is unclear whether the analogous rate prediction for large protein folding would be similarly accurate or whether these results are fortuitous (such that the simulation of additional proteins would weaken the correlation). We are currently addressing this question by examining the folding of different and larger proteins.

What Have We Learned About the Protein Folding Mechanism?

The question of "how proteins fold" has been asked for decades, and remains a difficult problem due to the complexities and difficulties of computational and experimental methods. However, the methods presented here have allowed us to understand, for the first time, the folding mechanism for some small fast folding proteins, in atomistic detail with experimentally validated rates (Figure 8 and Table I). We have been able to discern the mechanism of a few particular proteins, but it is unclear whether we can expect these to generalize to larger and more complex proteins. An understanding of the mechanism of larger proteins will likely require further direct simulation. However, considering the diversity of mechanistic results found even in these small proteins, it seems reasonable to consider that there may not be a single, universal folding mechanism. Indeed, evolution may be mechanistically agnostic and may have selected proteins for function, without concern for folding mechanism. This could lead to a variety of protein folding mechanisms (even for sequences which fold to the same structure), and thus there may not be a *single* answer to the question of "how proteins fold."

Future Perspectives

The ensemble dynamics technique coupled with distributed computing has allowed us to break fundamental computational barriers in the dynamics of complex systems, such as protein and polymer folding. However, one need not build a distributed computing infrastructure to gain the benefits of our methods. Indeed, with a cluster accessible to almost any group (e.g., a hundred PCs), one can simulate 100 ns in a day (assuming 1 ns/processor). This is a significant advance over state of the art of traditional parallel molecular dynamics.[6] Of course, the combination of our method on top of traditional parallel MD (i.e., using traditional MD to speed up individual simulations to the maximum scalability of parallel MD and then using our method to statistically sample runs) may lead to the greatest advance, especially on massively parallel architectures with millions of processors, such as IBM's proposed million processor Blue Gene supercomputer.

Moreover, this technique should have broad applicability to any dynamical system that progresses by crossing free energy barriers, especially in the most intractable problems with high free energy barriers. It could also serve to augment existing computational methods, such as path sampling[20] (which requires simulating a fast trajectory over the relevant free energy barriers) or the determination of transition states using pfold analysis[50] (which is currently hindered by simulations dwelling in transiently stable intermediate states).

Finally, with the ability to reach time scales for protein folding in all-atom simulations (i.e., hundreds of microseconds), it is natural to ask whether the potential sets are adequate for folding. Indeed, due to the great number of calculations involved, distributed computing networks will most likely play an important part in providing sufficient computational power to extensively test and validate new potential sets. Considering the omnipresent role of force fields in structural biology, ranging from simulations, to informatics, to x-ray and NMR refinement, the ability to quantitatively test force fields will likely play a critically important role in structural biology and virtually all related fields.

It is our honor to have this work be part of the memorial issue for Peter Kollman. Peter was a great inspiration as a scientist and a senior colleague—generous with time and with praise and with numerous useful and encouraging suggestions. Indeed, in many ways our work on folding dynamics was inspired by his work with Yong Duan on the villin headpiece, as it opened our eyes to see just how close the field was to simulating time scales relevant for folding and thus to finally directly simulate protein folding.

We would also like to thank Kevin Plaxco for a critical reading of the manuscript, Robert Baldwin and Susan Marqusee for their comments about α-helices, Martin Gruebele and Jeff Moore for their comments about PPA folding kinetics, Dan Raleigh and collaborators for their unpublished results on the experimental folding time for the 36-residue villin fragment, Jay Ponder for allowing our use of the Tinker MD code in the Folding@Home client, and Bill Swope and Jed Pitera for their advice with modifying and extending Tinker. We would also like to thank Scott Griffin and the other members of the Intel distributed computing team for their help with the Folding@Home infrastructure and support of our work.

Much of this method was developed, tested, originally implemented, and run on the T3E at the National Energy Research Scientific Computing (NERSC) center at Lawrence Berkeley National Labs. We also thank the Advanced Biomedical Computing Center, National Cancer Institute (NCI), Frederick, Maryland, for its assistance and use of its SP3 and SV1. The helix and PPA production calculations we performed at NERSC and NCI. The other production calculations (beta hairpin and villin) were run on Folding@home. We would especially like to thank the tens of thousands of Folding@Home contributors, without whom this work would not be possible (a complete list of contributors can be found at http://Folding.Stanford.edu).

JC and EJS acknowledge support in the form of Computational Science Graduate Fellowship (DOE). SL is a James Clark fellow and acknowledges the support of a Stanford Graduate Fellowship. MS acknowledges the support of a Fannie and John Hertz fellowship and a Stanford Graduate Fellowship. YMR acknowledges the support of a Stanford Graduate Fellowship. CS and BZ each acknowledge support from a HHMI predoctoral fellowship. This work was supported by grants from the ACS PRF (36028-AC4), NSF MRSEC CPIMA (DMR-9808677), NIH BISTI (IP20 GM64782-01), ARO (41778-LS-RIP), and Stanford University (Internet 2), as well as by gifts from the Intel and Google corporations.

REFERENCES

1. Dill, K. A.; Chan, H. S. Nat Struct Biol 1997, 4, 10–19.
2. Brooks, C. L.; Gruebele, M.; Onuchic, J. N.; Wolynes, P. G. Proc Natl Acad Sci USA 1998, 95, 11037–11038.
3. Dobson, C. M.; Sali, A.; Karplus, M. Angew Chem Int Edit Engl 1998, 37, 868–893.
4. Prusiner, S. Proc Natl Acad Sci USA 1998, 95, 13363–13383.
5. Nelson, J. C.; Saven, J. G.; Moore, J. S.; Wolynes, P. G. Science 1997, 277, 1793–1796.
6. Duan, Y.; Kollman, P. A. Science 1998, 282, 740–744.
7. Shirts, M. S.; Pande, V. S. Science 2000, 290, 1903–1904.
8. Shirts, M. S.; Pande, V. S. Phys Rev Lett 2000.
9. Voter, A. F. Phys Rev B 1998, 57, 13985–13988.
10. Williams, S.; et al. Biochemistry 1996, 35, 691–697.
11. Thompson, P.; Eaton, W.; Hofrichter, J. Biochemistry 1997, 36, 9200–9210.
12. Nelson, J. C.; Saven, J. G.; Moore, J. S.; Wolynes, P. G. Science 1997, 277, 1793–1796.
13. Elmer, S.; Pande, V. S.; J Phys Chem B 2001, 105, 482–485.
14. Yang, W. Y.; Prince, R. B.; Sabelko, J.; Moore, J. S.; Gruebele, M. J Am Chem Soc 2000, 122, 3248–3249 (2000).
15. Blanco, F. J.; Serrano, L. Eur J Biochem 1995, 230, 634–649.
16. Munoz, V.; Thompson, P. A.; Hofrichter, J.; Eaton, W. A. Nature 1997, 390, 196–199.
17. Bryngelson, J. D.; Onuchic, J. N.; Socci, N. D.; Wolynes, P. G. Proteins Struct Funct Genet 1995, 21, 167–195.
18. Chandler, D. J Chem Phys 1978, 68, 2959–2970.
19. Pande, V. S.; Rokhsar, D. S. Proc Natl Acad Sci 1999, 96, 9062–9067.
20. Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. J Chem Phys 1998, 108, 1964–1977.
21. Doniach, S.; Eastman, P. A. Curr Opin Struct Biol 1999, 9, 157–163.
22. Elber, R. Curr Opin Struct Biol 1996, 6, 232–235.
23. Pande, V. S.; Rokhsar, D. S. Proc Natl Acad Sci 1999, 96, 1273–1278.
24. Pande, V. S.; Grosberg, A. Y.; Tanaka, T.; Rokhsar, D. S. Curr Opin Struct Biol 1998, 8, 68–79.
25. Pappu, R. V.; Hart, R. K.; Ponder, J. W. J Phys Chem B 1998, 102, 9725–9742.
26. Jorgensen, W. L.; Tirado-Rives, J. J Am Chem Soc 1988, 110, 1666–1671.

27. Qiu, D.; Shenkin, P. S.; Hollinger, F. P.; Still, W. C. J Phys Chem A 1997, 101, 3005–3014.
28. Andersen, H. C. J Comput Phys 1983, 52, 24–34.
29. Pande, V. S.; Rokhsar, D. S. Proc Natl Acad Sci 1999, 96, 1273–1278.
30. Young, W. S.; Brooks, C. J Mol Biol 1996, 96, 560–572.
31. Ferrara, P.; Apostolakis, J.; Caflisch, A. J Phys Chem B 2000, 104, 5000–5010.
32. Zagrovic, B.; Sorin, E. J.; Pande, V. J Mol Biol 2001, 313, 151–169.
33. Dinner, A. R.; Lazaridis, T.; Karplus, M. Proc Natl Acad Sci USA 1999, 96, 9068–9073.
34. Zhou, R.; Berne, B.; Germain, R. Proc Natl Acad Sci USA 2001, 98, 14931–14936.
35. Garcia, A. E.; Sanbonmatsu, K. Y. Proteins 2001, 42, 345–354.
36. Bryant, Z.; Pande, V. S.; Rokhsar, D. S. Biophys J 2000, 78, 584–589.
37. Raleigh, D.; et al. Private communication.
38. McKnight, C. J.; Matsudaira, P. T.; Kim, P. S. Nat Struct Biol 1997, 4, 180–184.
39. Thompson, P. A.; Eaton, W. A.; Hofrichter, J. Biochemistry 1997, 36, 9200–9210.
40. Pande, V. S.; Grosberg, A. Y.; Tanaka, T.; Rokhsar, D. S. Curr Opin Struct Biol 1998, 8, 68–79.
41. Ptitsyn, O. B. Adv Protein Chem 1995, 47, 83–229.
42. Finkelstein, A. V.; Shakhnovich, E. I. Biopolymers 1989, 28, 1667–1680.
43. Pande, V. S.; Rokhsar, D. S. Proc Natl Acad Sci USA 1998, 95, 1490–1494.
44. Pande, V. S.; Grosberg, A. Y.; Tanaka, T. J Chem Phys 1995, 103, 9482–9491.
45. Ferrara, P.; Caflisch, A. Proc Natl Acad Sci 2000, 97, 10780–10785.
46. Cramer, C. J.; Truhlar, D. G. Chem Rev 1999, 99, 2161–2200.
47. Jorgensen, W. L. J Am Chem Soc 1981, 103, 335.
48. Kilmov, D.; Thirumalai, D. Phys Rev Lett 1997, 79, 317–320.
49. Zagrovic, B.; Pande, V. S. 2002, in preparation
50. Du, R.; Pande, V. S.; Grosberg, A. Y.; Tanaka, T.; Shakhnovich, E. I. J Chem Phys 1998, 108, 334–350.
51. Kabsch, W.; Sander, C. Biopolymers 1983, 22, 2577–2637.