Computational Approach to the Statistical Mechanics of Protein Folding

Ming-Hong Hao and Harold A. Scheraga*
Baker Laboratory of Chemistry, Cornell University,
Ithaca, NY 14853-1301,U.S.A.

October 27, 1995

Abstract

A statistical mechanical approach to the protein folding problem is developed based on computer simulations. The properties of proteins related to conformation and folding are determined from the density of states of the protein. A new simulation procedure, the Entropy Sampling Monte Carlo method, is used to determine accurately the density of states of the protein. To enhance the efficiency of sampling the conformational space of a protein, two techniques (a conformational-biased chain regrowth procedure and a jump-walking method) were introduced into the simulation. Applications of the approach to study a number of model polypeptides and a small protein, Bovine Pancreatic Trypsin Inhibitor, have been carried out. The results obtained demonstrate that the new approach is more powerful and produces richer information about the thermodynamics and folding behavior of proteins than conventional simulation methods.

I. Introduction

The emergence of supercomputers and advances in supercomputing technology have changed almost every field of scientific research. They not only shorten the time needed to carry out simulations, but also make it feasible to carry out many computational tasks that were previously impossible. Roughly speaking, computing in scientific research involves two types of activity: (1) devising new programs to increase the efficiency of existing simulation methods, and (2) developing new simulation methods to solve new problems or to treat known problems with higher accuracy. In this paper, we focus on a computational problem in the field of protein

folding, i.e. a study of the statistical mechanics of protein folding by computer simulation. Even though we have taken advantage of new techniques such as parallel computing, our paper is concerned primarily with the basic formulations for carrying out the desired simulations accurately. There is still much potential in our approach for utilizing more sophisticated programming techniques that would be helpful for treating larger proteins in a shorter time, but they are not the focus of the present paper.

Proteins are linear polymeric molecules formed from twenty different kinds of amino acid residues; the linear size of proteins is on the order of 10^2 residues per molecule. The structures of proteins are highly flexible, and, as a result, there is an astronomically large number of possible conformations for any given protein. In order for proteins to carry out their biological functions, they must fold to particular conformational states called "native structure". A theoretical understanding of the protein-folding problem, i.e. how proteins fold to their native states from a huge number of unfolded states, is a very challenging problem. As with many physical problems with a large number of states, the problem of protein folding can be addressed conveniently by statistical mechanical methods. While analytical treatments of protein folding based on simplified models have been helpful for our understanding the problem[1–6], a detailed treatment of the folding problem for **specific** proteins under concrete interactions has to depend on computer simulations.

The main task in the statistical mechanical treatment of protein folding is an accurate determination of the density of states of the given protein. Traditional Monte Carlo (MC) methods, adopted for simulating polymeric molecules, have been quite useful for studying some protein folding problems[7–12], such as the average properties of a protein in the folding process and the kinetic pathway of folding. However, traditional MC methods are not well suited for obtaining the free energy of a system. In particular, as far as simulating proteins is concerned, it is very difficult, if not impossible, to use traditional MC methods to obtain a true sample of the complete conformational space of a realistic protein as is required for a statistical mechanical treatment. There are two main factors that create this difficulty: one is the extremely large number of conformations of a protein, on the order of 3^{100}, which cannot be enumerated by any present-day computer; another is the very rugged shape of the potential energy surface of a protein which usually causes conventional simulation methods to become trapped in the valley of a particular energy minimum at relatively low temperature, making it almost an impossible task to sample the complete conformational space of a realistic protein.

In this paper, we describe a new simulation procedure which overcomes the above difficulties and makes the study of the statistical mechanics of protein folding more reliable. There are two essential elements leading to the success of this approach: (1) an iterative procedure that converges rather rapidly to the true density of states of the protein without sampling all the possible conformations, and (2) a feed-back mechanism that can overcome traps of local energy minima. The basics of the algorithm are explained in Section II; applications of the methods are discussed in Sections III and IV. To readers who are familiar with the field of protein folding, we emphasize that the goal of our approach is **not** aimed simply at folding proteins in some computer experiments, but rather at answering the questions as to whether or not a given polypeptide can fold to a unique native structure and as to what are the conditions for such a folding transition to occur.

II. Methodology

A. Lattice Representation of Protein Structures

A statistical mechanical treatment of the folding problem requires an extensive sampling of the conformations of the protein. It is, therefore, necessary to find a representation for the structure of a protein that can be rapidly and efficiently manipulated in a computer algorithm. The lattice-chain model offers such a representation. In 3D space, the primary lattice is the cubic lattice, in which the unit vector of the lattice can be used to represent a basic unit of the protein chain, i.e. a C^{α} - C^{α} virtual bond. The cubic lattice, however, is not very good in representing the intrinsic geometry of polypeptide chains because the C^{α} - C^{α} virtual bond angles of polypeptides have many different values other than the 90° provided by the cubic lattice. Moreover, the lattice is anisotropic in contrast to real space. Therefore, we first describe a systematic way of improving the accuracy of the lattice representation of protein structures.

The basic unit of the cubic lattice may be divided into 2, 3 or even a larger number of smaller units. Starting from the simple cubic lattice (Figure 1A), if the basic lattice unit is divided into two, a finer (cubic) lattice with unit length of 1/2 of that of the original lattice is obtained. Among the more available vectors in the new lattice, one can use a sequence of vectors of the type $(\pm 2, \pm 1, 0)$ and all their permutations (Figure 1B) to represent the C^{α} - C^{α} virtual bonds in the protein structure. This is called the (210) chain lattice which was first used by Kolinski and Skolnick[8,13]. When the basic vector of the simple cubic lattice is divided into three, an even finer (cubic) lattice with basic units of only 1/3 of that of the original lattice is obtained. On this lattice, more vector types are available and, to maintain

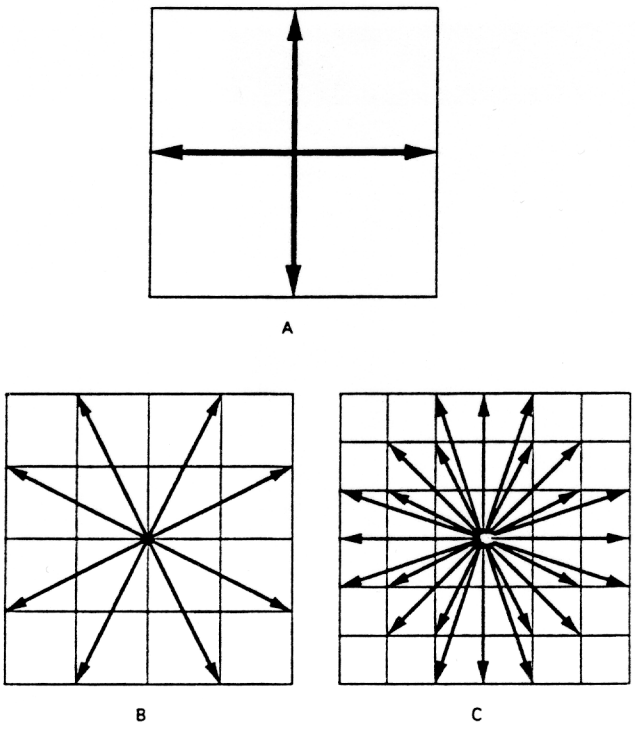

Figure 1: Two-dimensional illustration of the basic vectors in the simple cubic lattice (A), the (210) lattice (B), and the hyper lattice (C). Lattices B and C are derived from A by dividing the basic unit of the cubic lattice into smaller units and selecting a set of vectors.

the proper bond-length of the C^{α} - C^{α} virtual bond, one can use vector types $(\pm 2, \pm 1, 0)$, $(\pm 2, \pm 2, 0)$, $(\pm 2, \pm 2, \pm 1)$, $(\pm 3, 0, 0)$, and $(\pm 3, \pm 1, 0)$, including the permutation of the components in each type, to represent the C^{α} - C^{α} virtual bonds in the proteins. A two-dimensional illustration of the basic vectors in such a lattice is shown in Figure 1C. This lattice will be called, somewhat arbitrarily, the hyper lattice. Further divisions of the simple cubic lattice in a similar way are obviously possible.

It is clear that, as the division of the simple cubic lattice becomes finer, more points in a unit space will be available for representing the structure of the protein, the resulting lattice chain becomes less anisotropic, and a more accurate representation of the protein structure will be obtained. However, when the lattice becomes finer, the computational demand also increases. Therefore, there should be a compromise between accuracy and computational cost. In our studies, the (210) lattice chain model was used in simulations of model polypeptides, and the hyper lattice chain was used in simulations of the protein BPTI (Bovine Pancreatic Trypsin Inhibitor).

B. Entropy Sampling Monte Carlo (ESMC)

The statistical mechanical treatment of protein folding requires an accurate determination of the density of states of the protein. As discussed in the Introduction, this is a very difficult task with conventional MC methods. Therefore, we have developed a new MC procedure[14–16] based on the idea of entropy sampling Monte Carlo (ESMC) first proposed by Lee[17]. The ESMC method originated as the multicanonical MC method[18,19] but has a simpler and more general form than the earlier method. Even though the new procedure is named ESMC, the main body of the simulations actually does *not* use the exact entropy of the protein.

In contrast to conventional MC, the ESMC simulations are carried out based on a distribution of conformations in which the probability $P(E)$ of occurrence of states at energy level E is defined as

$$P(E) \propto N(E)\, e^{-J(E)} \tag{1}$$

where $N(E)$ is the number of conformations with energy E, i.e. the density of states, and $J(E)$ is a function of energy. The function $J(E)$ acts to scale the probability of occurrence of states at energy level E. Once the probability function $P(E)$ is given, the MC simulation is carried out according to the Metroplis procedure, but with the function $e^{-J(E)}$ replacing the usual Boltzmann probability $e^{-E/T}$. Thus, when an old conformation x with energy $E(x)$ is perturbed to a new conformation x' with energy $E(x')$, the acceptance probability is defined as

$$P(x \rightarrow x') = min\ [\ 1,\ e^{\{-J[E(x')]+J[E(x)]\}}\ ] \tag{2}$$

For numerical calculations, the energy space is discretized into bins with width ΔE. During the MC simulations, a histogram $H(E)$ is collected from the number of visits to the conformations in each bin. After a sufficient number of MC runs, the function $J(E)$ is updated by the following formula:

$$J_{new}(E) = \{ \begin{array}{ccc} J_{old}(E) + lnH(E) & if & H(E) > 0 \\ J_{old}(E) & if & H(E) = 0 \end{array} \tag{3}$$

This formula can be considered as the mathematical definition of the function $J(E)$. From Eq.(1), it is clear that, the larger the function $J(E)$, the smaller the probability of occurrence of states with energy E and, as a result, the histogram for that state obtained in the simulation will be smaller, and the increase of the new $J(E)$ after the updating will be smaller. Therefore, the moving direction of $P(E)$ in a series of interactions is toward the

uniformity of the function $P(E)$. The criterion of convergence of the simulation is simply defined as $P(E) = const$. When the convergence condition is reached, one obtains from Eq.(1):

$$N(E) = const \times exp[J_{conv}(E)]. \tag{4}$$

This determines the density of states of the protein from the converged function $J_{conv}(E)$.

We summarize here the distinctive features of the ESMC method. First, the procedure has a built-in mechanism of convergence to the true density of states of the system, and there is an unequivocal criterion to determine the convergence of the simulation. Second, in this method, the information gained from earlier simulations about the distribution of conformations of a protein is fed directly back to guide the following simulations. Finally, in the ESMC method, the probability of occurrence of states at different energy levels always moves toward the uniformity of the probability function. When the simulation is near convergence, both low- and high- energy states have nearly equal probability to be sampled; this creates an optimal condition for obtaining a representative sampling of all energy states. These properties form the basis of the algorithm to generate a complete density of states for proteins.

C. Conformational Biased Sampling

In simulations of proteins, a basic phenomenon is that, from a folded (unfolded) conformation, it is much easier to pick another folded (unfolded) conformation than to pick an unfolded (folded) conformation because of the polymeric nature of protein structures. If the sampled conformations are taken predominantly from one of the states, it is obviously impossible for the ESMC procedure to obtain a reliable estimate of the relative probabilities for all the conformational states. To increase the efficiency of sampling all conformational space of proteins in the ESMC procedure, we introduced a biased conformational generating procedure[20,21] into the simulation process. The basic idea is to use certain biases to increase the probability of sampling the compact conformations.

In the conformational-biased chain generation procedure[20,21], a new conformation is generated by discarding part of the old chain and then regrowing the discarded part of the chain residue by residue. In regrowing each residue, the local chain states (LCS) of the residue are given weight factors proportional to their Boltzmann factors. The choice of a certain LCS for

growing this residue is calculated by the probability:

$$w_i = \frac{e^{-\beta E_i}}{\sum_j \ e^{-\beta E_j}} \tag{5}$$

where E_i is the interaction energy of the residue in the ith LCS with all residues that are already on the chain, β is an inverse temperature parameter chosen to increase the weight of folded conformations[20], and the sum is evaluated over all possible LCS for the added residue. After a new conformation is generated by such a biased move, the probability of acceptance of the new conformation with respect to the old is calculated by Eq.(2), corrected for the different weights when both conformations were generated by the conformational bias procedure. The corrected acceptance probability is[20]:

$$P(x \rightarrow x') = min\ [\ 1,\ \frac{e^{-S[E(x')]}}{e^{-S[E(x)]}} \frac{\prod_I w_I}{\prod_I w'_I}\ ] \tag{6}$$

where the w_I are probabilities for growing the Ith residue.

The above conformational-biased chain generation step is introduced stochastically into the regular conformational-updating procedure based on local perturbations. In general, one regrowth of a part of the chain takes a time equivalent to several hundreds of local chain moves. However, the conformational-biased chain generation procedure is very effective in sampling new conformational space, and, in particular, in sampling compact low-energy conformations. As a result, the overall efficiency of the sampling algorithm increases.

D. Jump-and-Walking procedure

For sufficiently large polypeptides and proteins, the convergence of the ESMC simulation can be very slow. The main reason is that the structures of proteins occupy two very different conformational regimes, the statistical coiled state and the globular state. The transition between these two conformational regimes in an ordinary simulation procedure is usually slow and rare. When the estimated entropies $J(E)$ for the coiled and globular states are very different, either too large or too small, from their correct relative entropy $S(E)$, the updated entropy function according to Eq.(3) could change either too much or too little in an iteration in a certain conformational region. As a result, the updated function $J_{new}(E)$ would oscillate instead of steadily approaching a fixed point.

To overcome the above problem, we incorporate into the ESMC procedure the jump-walking technique[22,23] originally proposed for the standard MC

procedure. Our basic idea is to break a single long ESMC run in an iteration into many short runs, each of which starts with an independent new conformation. The new starting conformations of these short runs are not generated completely randomly, but rather are drawn from a conformational pool which is generated in a previous ESMC iteration. When the probabilities of occurrence of the conformations in the pool approach those defined by Eq.(1), the conformations drawn randomly from the pool will be nearly uniform over all energy levels. As a result, in an identical amount of computational time, the simulation with a number of short runs will produce a more uniform sampling over the whole conformational space than a single long run does, and the convergence of the simulations is greatly improved.

The jump-walking technique is implemented in the following way. First, starting from a normal ESMC simulation, randomly selected conformations in the simulation are saved to form a conformational pool. Then, in the following simulations, the procedure jumps occasionally, with a certain probability, to the conformational pool saved previously to choose a new conformation with which to continue the regular ESMC run. The jump is treated as a particular kind of conformational move. Let the estimated entropy functions for the current simulation and for the previous simulation from which the conformational pool was generated be $S_I(E)$ and $S_{II}(E)$, respectively. In a jump, the conformation x in the current step is switched to the conformation x' randomly chosen from the pool, and then the regular simulation procedure is resumed. If the transition between conformations x and x' were carried out by a *direct* move from the current simulation, the forward and backward transition probabilities, $\pi(x';x)$ and $\pi(x;x')$, would be determined by

$$\pi(x';x)/\pi(x;x') = e^{-S_I[E(x')}/e^{-S_I[E(x)]} \equiv e^{-\Delta S_I(x';x)} \qquad (7)$$

where $\Delta S_I(x';x) \equiv S_I[E(x')] - S_I[E(x)]$.

The new conformation chosen through the jump, however, is not selected completely randomly, but rather is determined by the distribution of conformations in the pool. The transitions between x and x' through the jump process are, therefore, affected by the relative statistical probabilities of the two conformations in distribution II. To correct for the effect of the unequal probabilities of conformations x and x' in distribution II, the forward and backward transition probabilities in a jump have to be determined by Eq.(8) instead of Eq.(7):

$$\pi(x';x)/\pi(x;x') = e^{-\Delta S_I(x';x)}/e^{-\Delta S_{II}(x';x)} \qquad (8)$$

where $\Delta S_{II}(x';x) \equiv S_{II}[E(x')] - S_{II}[E(x)]$. The probability of acceptance for the jump is finally expressed as:

$$P_j(x \rightarrow x') = min\ [\ 1,\ e^{[-\Delta S_I(x';x) + \Delta S_{II}(x';x)]}\ ] \qquad (9)$$

The above jump-walking procedure is incorporated into the ESMC simulations only when the whole conformational space has been covered in the initial simulations. The conformational pool is also updated from each iteration in the simulations.

E. Parallelization of ESMC simulations and data analyses

It can be seen from subsection B that the main information used in the ESMC simulations is the histogram $H(E)$. In theory, whether the histogram is obtained from one long simulation or from a number of shorter and independent simulations does not affect the updated function $J(E)$ as long as the two different simulations include the same number of moves and are performed with the same probability function. Therefore, the ESMC simulation can be carried out in a straightforward parallel fashion. We have taken advantage of the KSR shared memory parallel computer at the Cornell Theory Center in our simulations. The parallel part of the algorithm is incorporated into the simulation in conjunction with the jump-walking steps. To start the simulation, a number of processors pick a starting conformation independently from the conformational pool. Then, each of the processors carries out its part of the ESMC runs. After all the processors finish their jobs, the sub-histograms obtained in the different processors are combined together to yield the total histogram. The updating of the function $J(E)$ is based on the total histogram according to Eq.(3). This parallelization is at the most coarse-grained level, and, with 50 processors, the speed-up of the simulations on the parallel computer is directly proportional to the number of the processors used.

We conclude this section by summarizing the procedures for analyzing the results from the ESMC simulations. The direct results from the ESMC simulation is the function $J(E)$. The relative density of states of the protein is defined by $J_{conv}(E)$ in Eq.(4). Since the entropy $S(E)$ of the protein at a given energy level E is related to the density of states by $S(E) = kln[N(E)]$, the function $J_{conv}(E)$ is equal to the relative entropy $S(E)$ of the protein. Once the relative entropy is known, all other thermodynamic properties of the protein can be calculated. For example, the relative free energy at different energy levels, $F(E,T)$, is calculated from $S(E)$ as

$$F(E,T) \ = \ E \ - \ TS(E) \qquad (10)$$

where T is the canonical temperature. The average energy of the protein at a given temperature can be calculated as

$$< E >= \sum_E E\, exp[S(E) - E/T] / \sum_E exp[S(E) - E/T] \qquad (11)$$

Indeed, the average value of any mechanical property M can be calculated from $S(E)$ as

$$< M >= \sum_E \overline{M}(E) exp[S(E) - E/T] / \sum_E exp[S(E) - E/T] \qquad (12)$$

where $\overline{M}(E)$ is the mean value of the property M at the given energy level E. $\overline{M}(E)$ can be obtained from the same ESMC simulation without costing much extra computer time.

III. Simulations on Model Polypeptides

The study on model polypeptides serves several purposes: (1) the chain representation of the model polypeptide can be made simpler, (2) the potential functions that describe molecular interactions of model polypeptides can purposely be chosen to study certain features of a protein, (3) model polypeptides bear basic similarities to real proteins, and (4), therefore, model polypeptides can be used to study the fundamental characteristics of protein folding and to test the effectiveness of the algorithm with less computational demand.

The model polypeptides are represented by the (210) lattice chain (described above). The side chains are represented by a single sphere attached to the backbone C^α atom. Four types of interactions are considered in the model: (a) The excluded volumes of the C^α atoms are treated as hard spheres with a radius of $\sqrt{2}$. (b) Side chains are divided into three types: polar (P), hydrophobic (H) and neutral (U). Interactions between H-H type residues are represented by the potential $V(r) = \epsilon_H[(\sigma/r)^{12} - 2(\sigma/r)^6]$, and between H-P and P-P type residues by $V(r) = \epsilon_P(\sigma/r)^{12}$, where r is the distance between the centers of the side-chain spheres, σ is the characteristic separation of the spheres, and ϵ_H and ϵ_P are the strengths of the two types of interactions, respectively. Neutral residues have only an excluded volume. (c) Two non-neighboring bond vectors are associated with a dipolar interaction energy, $-\epsilon_{dp}$, when the two vectors are anti-parallel and the distance between the centers of the bonds is smaller than a given distance, r_d; this term accounts for the energies of main-chain hydrogen bonds. (d) Interaction energies between neighboring residues are accounted

for by a bond-angle potential for the angles between neighboring $C^\alpha - C^\alpha$ bonds. Bond angles may be two types: those with and those without a preferred state. For the bond angles with a preferred state, an energy of $-\epsilon_{ang}$ is associated whenever it adopts the preferred value; otherwise, its energy is zero. For bond angles with no preferred state, the energy of all its states is zero.

The (210) lattice chain provides a good representation for random conformations as well as for the β-type structures of proteins. But its representation for α-helical structures is not very satisfactory. The present study is therefore restricted to β-type proteins (when the real protein is studied later, the hyper lattice is used since it can represent all types of local structures well). The following energy and geometry parameters are used in the simulations: $\epsilon_H = 1.3$, $\epsilon_P = 0.7$, $\epsilon_{dp} = 0.6$, $r_d = 4.0$, $\sigma = 2.8$, and the bond-angle parameter ϵ_{ang} is tested with two values: 1.5 and 1.0. The native bond angle is chosen so that the distance between atoms 1 and 3 is $\sqrt{16}$, i.e. the value for the β structure on the (210) lattice. Distances are measured in lattice units, temperature in units of $\epsilon_H/(1.3k)$, where ϵ_H is defined above, and k is the Boltzmann constant.

A number of model polypeptides with different sequences have been studied[14–16]. For the purpose of comparison, we will discuss two 38-residue polypeptides in this paper, one with an optimized sequence:

S_1: UHPHPHPHP$\underline{U}$PHPHPHP$\underline{H}$UU$\underline{H}$PHPHPHP$\underline{U}$PHPHPHPHU

and another with a random sequence:

S_2: UHPPHHHPP$\underline{H}$PHPPPHH$\underline{P}$HH$\underline{H}$HPHPHPH$\underline{H}$PPHHPPHPU

where U, P and H are the residue types defined above. In the above sequences, the bond angles at the positions where the center atom is underlined are angles with no preferred state, while all other bond angles prefer the $\sqrt{16}$ state. Turns in the conformations of the above model polypeptides will prefer to occur at the positions with underlined residues because there is no bond-angle energy bias at those positions. The dipolar interactions lead to a preference for anti-parallel regularly extended conformations. The long-range side-chain interactions vary in different sequences. While S_2 is a random sequence in terms of side-chain long-range interactions, S_1 is a designed sequence in which [in the targeted (two-layered four-stranded β-sheet) structure] all side chains that are in contact are H type residues while side chains that point to the outside are P type residues. In other words, in the targeted structures, both long-range and short-range energies are minimized; therefore, this sequence can be considered as satisfying the "principle of minimum frustration"[6].

In our ESMC simulations, one conformation updating consists of a sequence

of random local perturbations[7,8], in which **every bond** of the chain makes on average one two-bond spike-like rotation (for the chain ends it is a flip of the end bonds), one four-bond rearrangement and one eight-bond rearrangement, followed by a conformational-biased partial chain regrowth. The resulting conformation is considered as a new conformation in the conformational updating process. The energy space is divided into bins of 0.25 energy unit. The initial rough estimates of the function $J(E)$ for the model polypeptides were obtained by a number of short iterations, ranging from 10 to 20 until the lowest-energy conformation has been sampled (as judged by the fact that, in further iterations, no new lower-energy states appear). Each iteration consists of 40,000 chain updates. After the rough estimate of the entropy function was obtained, a series of long iterations, each consisting of 10^6 chain updatings, was carried out to attain the convergence of the entropy function. In these long MC runs, the sampled conformations are saved at every 2,000 chain updates to form the conformational pool (which contains a total of 500 conformations). Starting from the second long iteration, the MC simulation incorporates the jump-walking procedure which draws a new conformation from the conformational pool for every 2,000 conformational updates. On the KSR computer, 20,000 conformational updates for the 38-mer model polypeptides required about one half hour on one processor. Using 50 processors, we could carry out a million chain updates (in a long run) within one hour.

Densities of States of Model Polypeptides

We now discuss the information and significance of the results obtained from the ESMC simulations.

Figure 2 and 3 show the relative entropy of states of sequences S_1 and S_2, respectively, obtained from the simulations. In these simulations, while the long-range interaction parameters were kept constant, the bond-angle potential was varied between two values to assess the effect of the relative strength of short- and long-range interactions on the folding transition of proteins. The insets of each figure show the standard errors of the MC simulation results in representative cases. The standard errors are defined as the root-mean-square variations of the simulated data points in the last four consecutive iterations. It can be seen from the insets that the standard deviations of the simulation data are very small, only about 0.2 entropy units with respect to the entropy values which are on the order of 10. Therefore, the accuracy of the MC results is very good. It is useful to note that, in the earlier ESMC simulations without introducing the jump-walking algorithm[14], the standard deviations of the simulated data were about 5 times larger than the present results. The jump-walking procedure

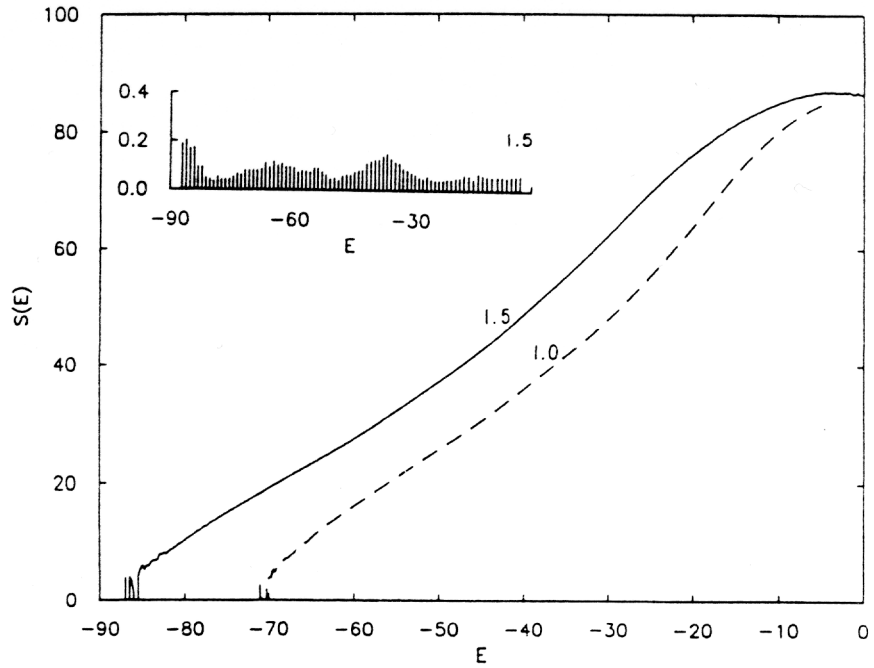

Figure 2: Relative entropy (in units of k) of states of sequence S_1 as a function of energy (in units of kT). The number on each curve pertains to the strength of the bond-angle potential in each case. The inset shows the standard errors (in units of kT) of the simulation data for the case labeled.

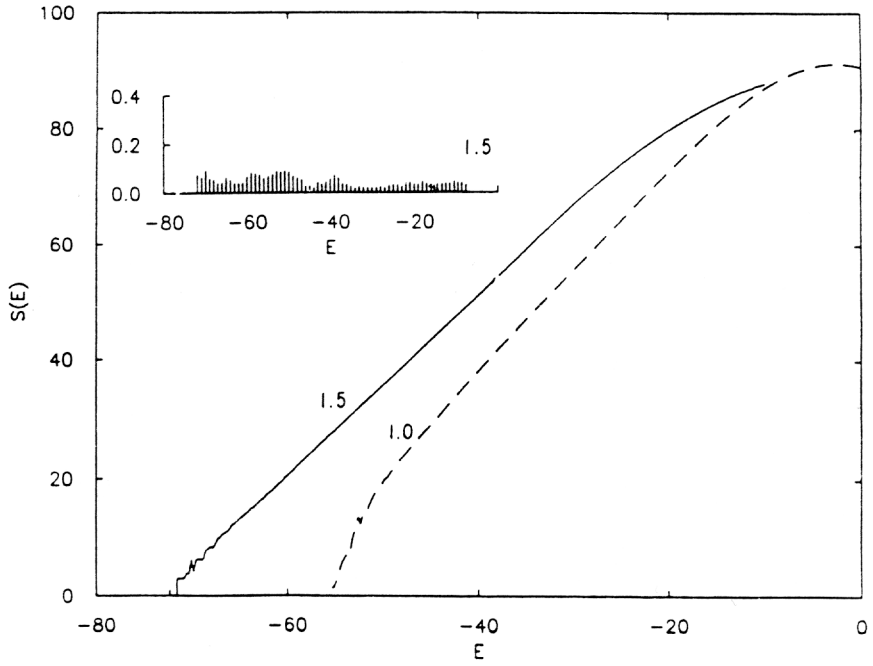

Figure 3: Same as Figure 2 but for sequence S_2.

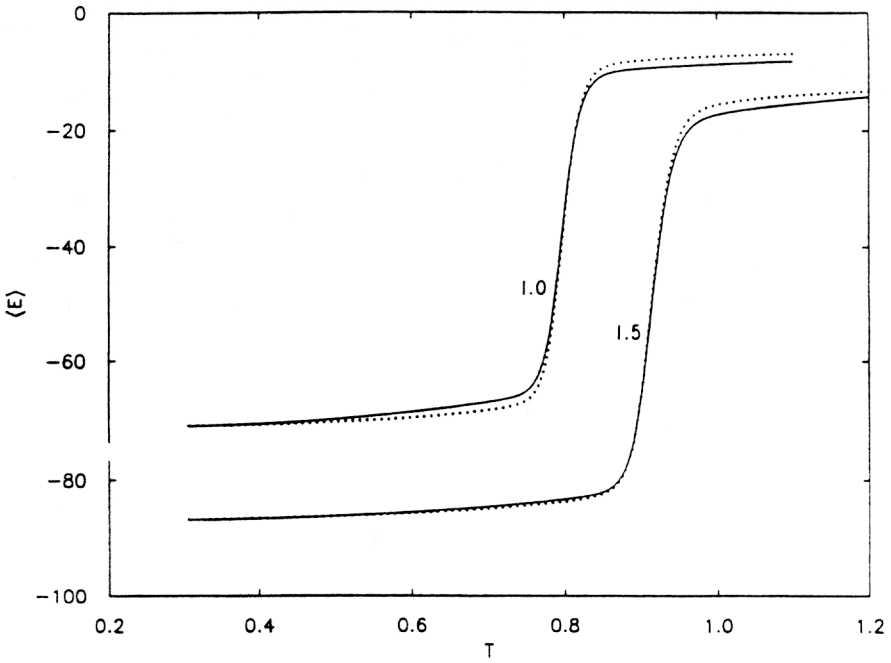

Figure 4: The thermal folding/unfolding curves of sequence S_1. The number on each curve is the strength of the bond-angle potential. The dotted curves were calculated by the theoretical model described in the text.

improves the convergence rate and the reproducibility of the simulation significantly.

The simplest way to describe the folding transition is to calculate the thermal folding-unfolding (or caloric) curve, which is the average energy of the protein as a function of temperature. Figure 4 and 5 show the calculated thermal folding-unfolding curves for the two sequences based on the results shown in Figs. 2-3. It can be seen from these Figures that both model polypeptides have one predominant folding transition at some folding temperature. Below the transition temperature, the average energy of a given polypeptide is very close to the lowest energy of the molecule; therefore, most residues must be in the native state below the transition temperature. Furthermore, all the transitions are rather sharp in the sense that the folding occurs in a narrow temperature region. The dotted curves associated with each curve in Figure 4 and 5 are obtained by an analytical model based on six mean-field energy parameters extracted from the simulation data for each sequence under a given potential. The analytical model is discussed below.

It should be noted that the same type of folding curves can be obtained by conventional MC simulations as well; for the purpose of folding a protein, it is not necessary to use the ESMC method. However, the folding curves

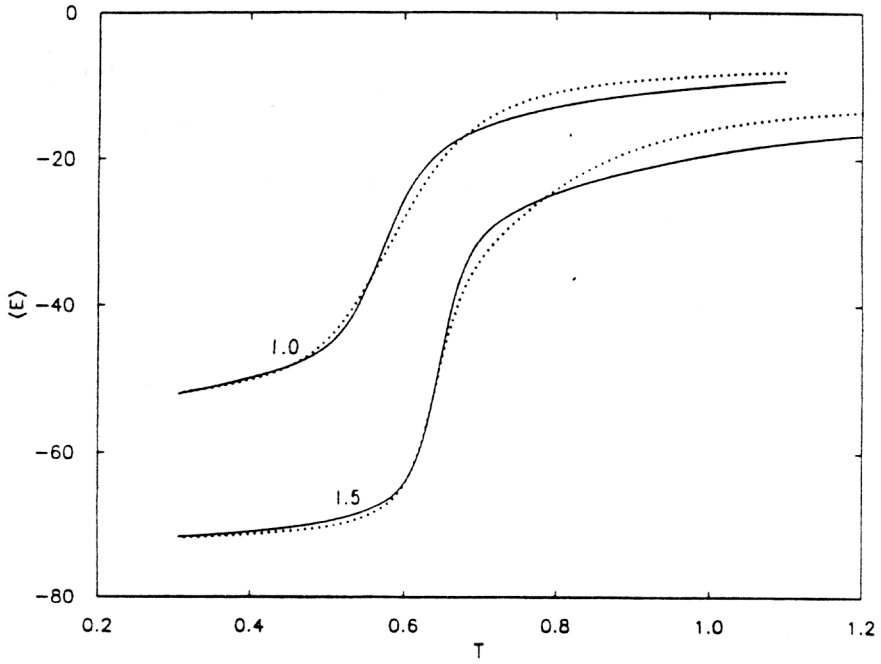

Figure 5: Same as Figure 4 but for sequence S_2.

shown in Figs. 4 and 5 leave many unanswered questions. For example, from the folding curves, it appears that both polypeptides (with different sequences under different potentials) undergo a similar folding transition. Do the folding transitions of both of these polypeptides lead to unique structures? What is the chance that a polypeptide folds to amorphous globules? Does an energy barrier exist in the folding? Is the folding transition first-order or continuous? Only the results from ESMC simulations contain the information that can resolve such questions.

Characteristics of the folding transitions

Whether or not a folding transition is two-state or continuous can be determined readily from the entropy curve. This is accomplished by evaluating the second derivative of $S(E)$ with respect to E. Denoting $d^2S(E)/dE^2$ by $\beta'(E)$, then entropy curves can be classified into four types[24]: (a) $\beta'(E) < 0$ everywhere but there is a finite maximum; (b) same as (a) except that there is a single point where $\beta'(E) = 0$; (c) same as (a) but there is a finite region of the curve where $\beta'(E) = 0$; and (d) $\beta'(E) > 0$ in a finite region and $\beta'(E) > 0$ everywhere else. For polypeptides with the (a)-type entropy curve, its folding transition is a continuous process without a barrier, for polypeptides with (b) and (c) type entropy curves, the folding transitions will be exactly second- and first-order, respectively, when the systems are extrapolated to infinitely large size, and for (d)-type entropy curves, the folding transition is first-order with a free energy barrier

between the folded and unfolded states.

According to the above rules, we can see from Figs. 4 and 5 that the type of folding transitions of the polypeptides depend on the sequence as well as the potential. For the optimized sequence S_1, its folding transition can be determined as first-order. This folding behavior of S_1 does not appear to depend very much on the strength of the bond-angle potential (see Ref.16 for more examples). For the random sequence S_2, its entropy curve under different strengths of bond-angle potential is either linear or convex in a finite region. Because the model polypeptide is a finite system, it is expected that the folding transition of S_2 is continuous without encountering a free energy barrier.

The clearest way to illustrate the difference between a two-state folding transition and a continuous folding is to calculate the relative free energy of different states according to Eq.(10). Figures 6a and 6b compare the free energy curves at the transition temperature for sequence S_1 and S_2, respectively, under the given potentials. It can be seen that, in sequence S_1 at the transition temperature, there is a high energy barrier separating the folded and unfolded states. For sequence S_2, however, the free energy curve shows a flat profile. The shape of the free energy curve is a critical factor in determining whether or not the folding transition will lead to a unique native structure or to a random globule. In polypeptides with sequences like S_1, the free energy barrier separates the unfolded and folded states in the folding transition. When the polypeptide folds from the unfolded state, it has a very large possibility to fold to the native state because this is a downhill movement in the free energy landscape and all the non-native states have smaller probabilities of occurrence. In contrast, for a polypeptide with a sequence like S_2, at the transition temperature there are no dominant states, and the polypeptide may fold to any one of many different conformations. As a result, the probability that such a polypeptide folds to the lowest-energy state is vanishingly small.

Order parameters for characterizing the folding transition of proteins

The simulated entropy curves for a polypeptide or protein can be fitted with an analytical expression, from which a set of energy parameters specific for each polypeptide can be extracted. This set of energy parameters is a quantitative measure that characterizes the thermodynamic properties of the given polypeptide. The analytical expression is based on a mean-field theory which has been described elsewhere[2,16]. The parameters include ϵ_0, ϵ_1 and ϵ_2, which characterize the mean-field energy of individual residues

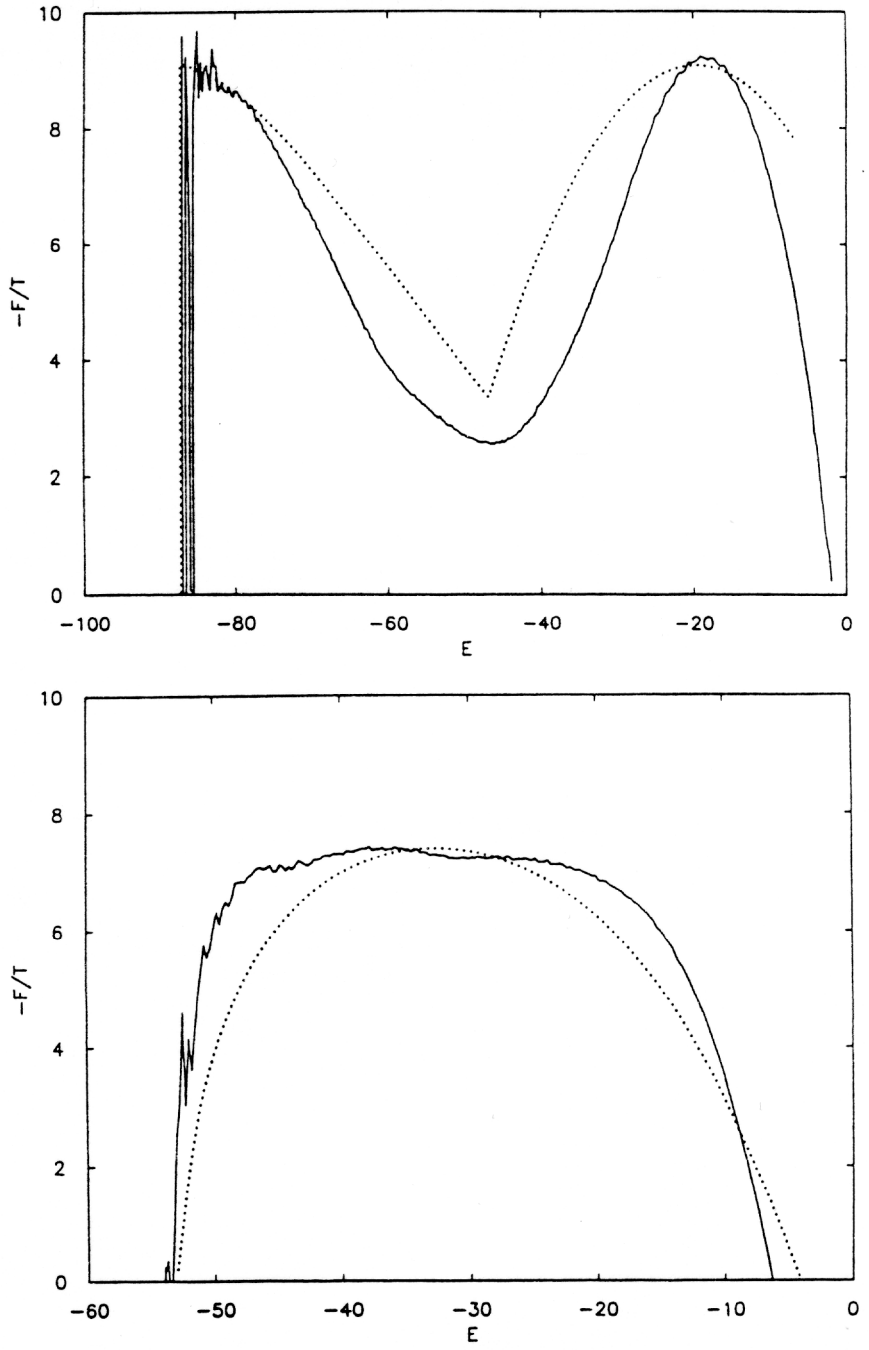

Figure 6: The free energy of states as a function of energy at the folding temperature (T). a.(top) Sequence S_1 at a bond-angle potential of 1.5 (T = 0.92); b.(bottom) Sequence S_2 at a bond-angle potential of 1.0 (T = 0.56). The dotted curves in the Figures were calculated based on the theoretical model described in the text.

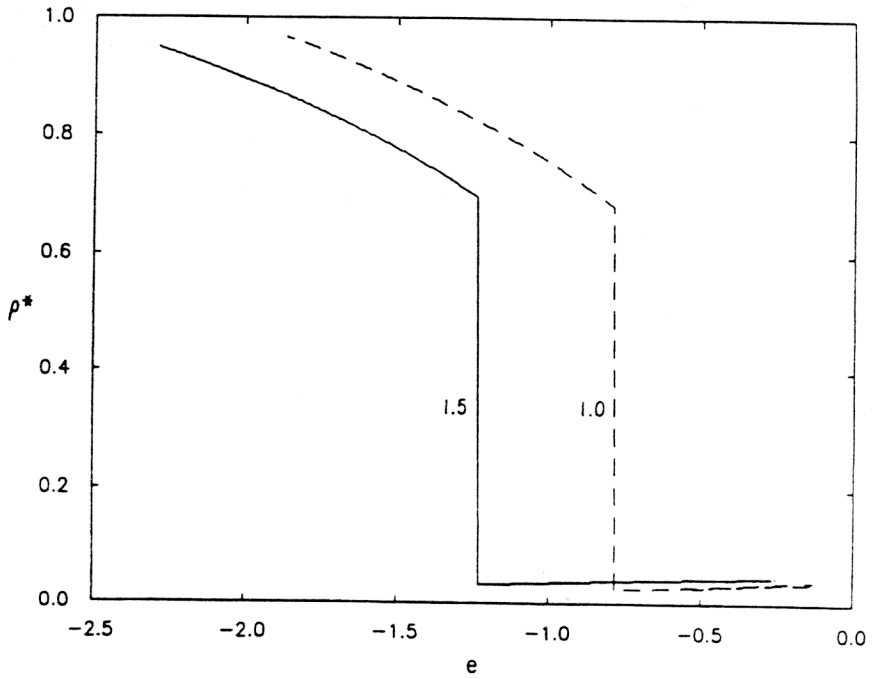

Figure 7: The most probable fraction of native state of sequence S_1 as a function of the energy per residue under the bond-angle potentials indicated by the labels on each curve.

in the protein, and δ_0, δ_1 and δ_2, which characterize the distribution of the energy of the individual residue. This theory states that the entropy of the protein at different energy levels can be defined by the formula:

$$S(E) = N\{-\rho^* ln\rho^* - (1-\rho^*)ln(\frac{1-\rho^*}{\nu})\} - N\{\frac{[E/N - \epsilon_0 - \epsilon_1\rho^* - \epsilon_2\rho*^2]^2}{2[\delta_0 - \delta_1\rho^* - \delta_2\rho*^2]}\} \quad (13)$$

To find the value of ρ^* at each E in the above equation one first maximizes the right side of Eq.(13) at the given E with respect to ρ^*, and then uses the resulting ρ^* to calculate $S(E)$.

These mean-field parameters for a given polypeptide can be extracted from the simulation data by fitting these data to the analytical expression by a least-square minimization. Once these parameters are obtained, all other thermodynamic properties of the polypeptide can be calculated based on analytical expressions and these parameters. The dotted curves in Figs. 4-6 were calculated by such a procedure.

The most interesting property from the analytical expression is the variable ρ^*, which bears the physical meaning of the most probable fraction of

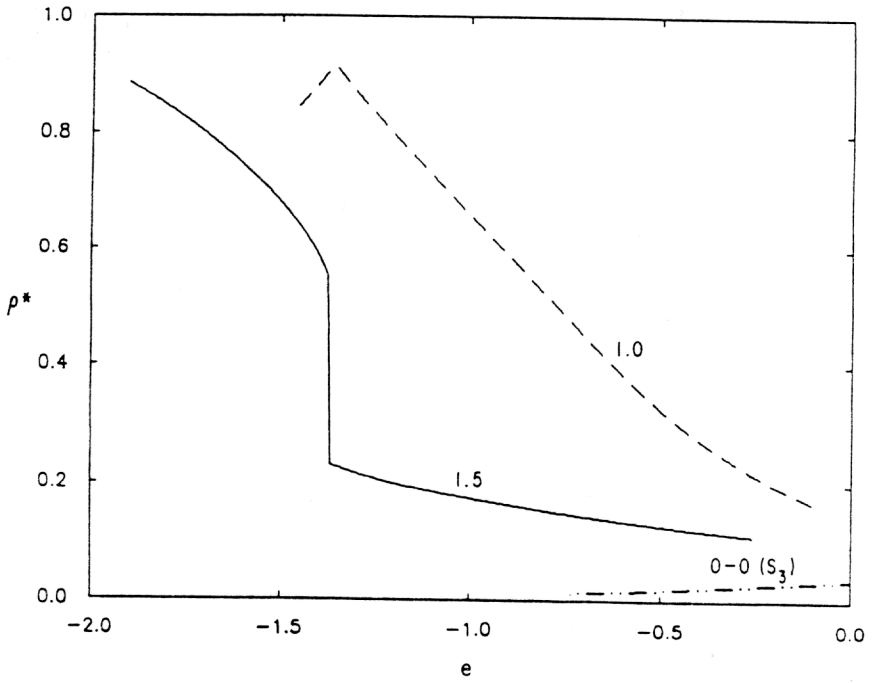

Figure 8: Same as Figure 7 but for sequence S_2. See text for curve S_3.

residues in the native state among the total residues of the protein. Evidently, ρ^* is the kind of order parameter by which one monitors how close a polypeptide is to the native state. Figures 7 and 8 show the plots of the most probable fraction of residues in the native state as a function of the energy per residue for the two different polypeptides. These Figures illustrate how differently the folding transitions of the two polypeptides behave. For sequence S_1, the order parameter of the polypeptide experiences a jump from the unfolded (high-energy) state to the native (low-energy) state. For sequence S_2, the order parameter mostly varies continuously. The curve labeled by 0-0(S_3) in Fig. 8 is presented here as a special reference. This is the order parameter of a 54-residue sequence under no bond-angle and dipolar interactions[16]. In this very special case, the order parameter never varies significantly as the average energy of the polypeptide decreases. The folding of this polypeptide is therefore like a glass transition: the folding does not result in an increase in the order of the conformation.

All the above information is contained in the density of states, or the entropy of states, of the protein, which has been obtained from an ESMC simulation. Therefore, an ESMC simulation is more informative than an ordinary MC simulation.

IV. Simulations on Bovine Pancreatic Trypsin Inhibitor (BPTI)

In order to study real proteins by the simulation approach, two necessary

requirements are: (1) a realistic representation of the protein structure, and (2) a potential function that correctly defines the structure of the protein.

The representation of the protein structure is a relatively easy problem. In the present study, we have used the hyper lattice chain (Fig. 1C) to represent the backbone of the protein. The side chains of the amino acid residues in the protein are represented by either two pseudo-atoms for large side chains (i.e. GLU, GLN, LYS, ARG, PHE, TRP and TYR) or one pseudo-atom for the rest of the residues except for GLY for which no side chain is included. The positions of the pseudo-atoms which represent the side chains of a residue are determined by the local backbone conformation, i.e. by the two C^{α} - C^{α} vectors connected to the C^{α} atom of the residue. A list of all the side-chain positions for all residues is contained in a database in the computer algorithm. To establish a lattice-chain representation of the native structure of a protein, the backbone of the protein is first fitted by the basic lattice vectors described earlier, grown one residue at time. Then, the side chains of all residues are attached on the C^{α} atoms of the residues. Finally, the whole lattice structure of the protein is optimized by minimizing the rms deviations of all the pseudo-atoms of the lattice chain and of the crystal structure, with additional constraints of the excluded volume of all the pseudo-atoms. The resulting lattice-chain structure is very close to the target crystal structure of the protein, the rms deviation between two such structures being about only 1 $\mathring{A}$ for most proteins. Figure 9 compares the backbone conformation of the crystal structure of BPTI and its lattice-chain representation; the difference between these two structures is seen to be very minor.

The derivation of a correct potential function for proteins is one of the fundamental problems in theoretical studies of protein folding. A complete treatment of this problem is beyond the scope of this article. In what follows, we *outline* a procedure for obtaining the potential function for simulating the folding of proteins. The potential function includes a bond-angle potential for local interactions:

$$E_{ang} = a_1(\nu)e^{-(\theta-90)^2/5^2} + a_2(\nu)e^{-(\theta-120)^2/20^2}, \tag{14}$$

where ν represents the type of the residue where the bond-angle is situated and θ is the value of the bond-angle, a hydrogen-bond interaction between the peptide vectors:

$$E_{HB} = H(\theta_{ij}, d_{ij}) \tag{15}$$

where $H = -0.6$ if $\theta_{ij} < 15^o$ or $\theta_{ij} > 165^o$ and $d_{ij} < 6.5\mathring{A}$, and $H = 0$ otherwise, and nonbonded interactions between all pseudo-atoms:

$$E_{nb} = \mu_{ij}[(\frac{r_{0,ij}}{r_{ij}})^8 - sgn(\frac{r_{0,ij}}{r_{ij}})^4]. \tag{16}$$

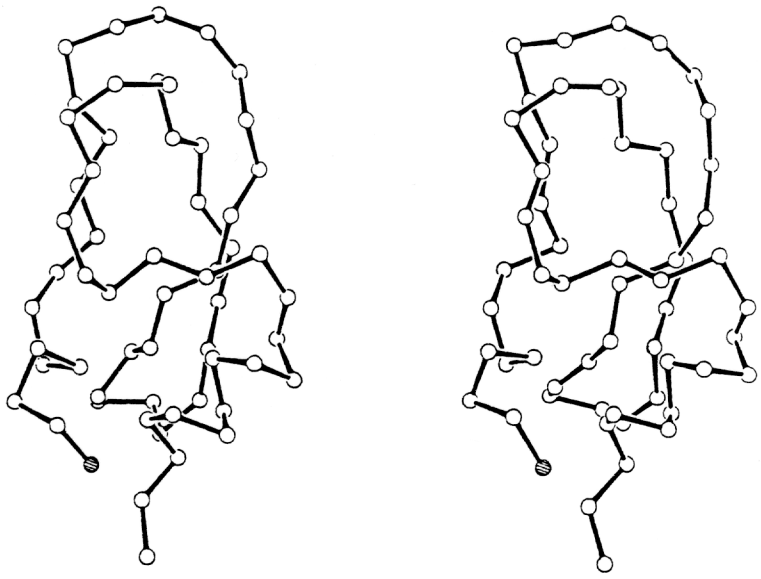

Figure 9: Comparison of the C^{α} chain conformation of the crystal structure of BPTI (left) and the lattice-chain representation of its structure (right).

where $sgn = \pm 1$ depending on i and j. The nonbonded interaction is a lump-sum of all sorts of pair-wise interactions between nonbonded atoms. The total energy of a protein is a sum of all the individual terms of the different energy components.

The energy parameters [such as $a_i(\nu), \mu_{ij}$, etc.] were optimized in the following way. Starting with a set of subjectively chosen energy parameters, an ensemble of conformations of the protein was generated by the Metropolis MC method at a number of temperatures. Using this ensemble of conformations, we minimized the following constrained function H with respect to all the energy parameters,

$$H = 1/N \sum_{i=1}^{N} \Delta E_i + K\sigma$$

$$subject\ to\ \ \Delta E_i < 0\ for\ all\ i \tag{17}$$

where N is the total number of conformations, ΔE_i is the difference in energies between the native structure and the ith generated conformation, K is a weighing factor, and σ is the root-mean-square deviation among all the values of ΔE_i. After an optimized set of energy parameters was obtained, we generated a new ensemble of conformations based on the new energy parameters, and carried out the minimization again. The above process was repeated until the optimized energy parameters did not change

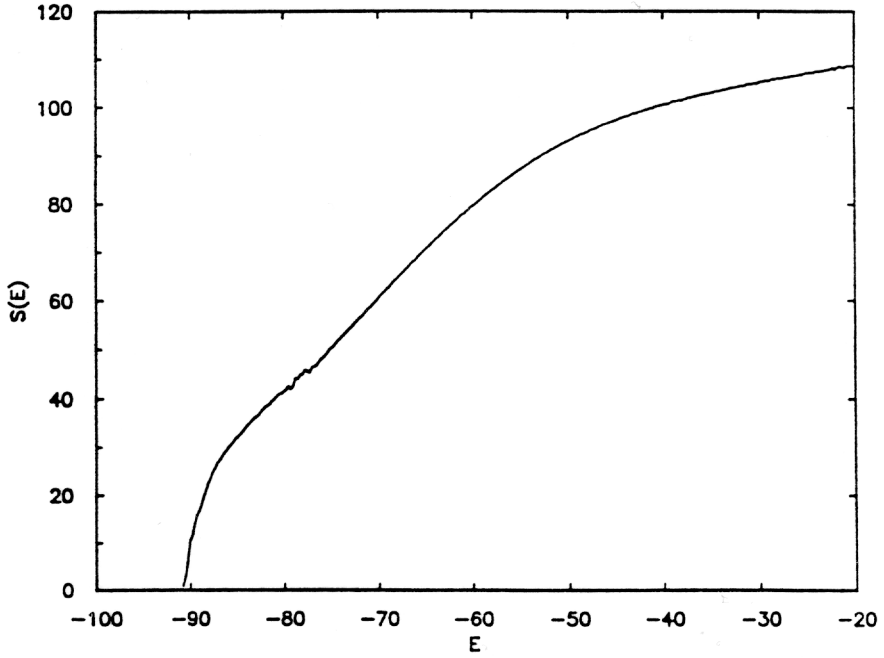

Figure 10: The relative entropy of BPTI at different energy levels E.

significantly. In this initial work, mainly for testing the ESMC procedure, we have used only one protein, BPTI, as the training protein for optimizing the energy parameters. In general, in order for the energy parameters to be applicable to other proteins, the training molecules should include different types of proteins.

With the optimized potential function, we have simulated BPTI with the ESMC procedure. The protein BPTI contains 58 residues. In the lattice-chain representation, the protein contains 132 particles, including both the C^{α} atoms and the pseudo-atoms from the side chains. Figure 10 shows the entropy (the logarithm of the density of states) of BPTI as a function of energy obtained from the simulations. The entropy curve has a finite segment ($-83 < E < -65$) where the curve is concave; therefore, the folding of BPTI will occur by means of a two-state transition. The clearest way to show the two-state characteristics of the folding transition is to calculate free energy of the protein. Based on the data shown in Fig. 10, the free energy of the BPTI was then calculated according to Eq.10. The type of the folding transition is determined by the free energy curve at the transition temperature. The state of the protein is identified by the energy: unfolded states of the protein have higher energies and folded states have lower energies. Figure 11 shows the free energy as a function of energy at the transition temperature. In this Figure, the folded and unfolded states can be clearly distinguished from the two separated minima of the free energy curves in two energy regions.

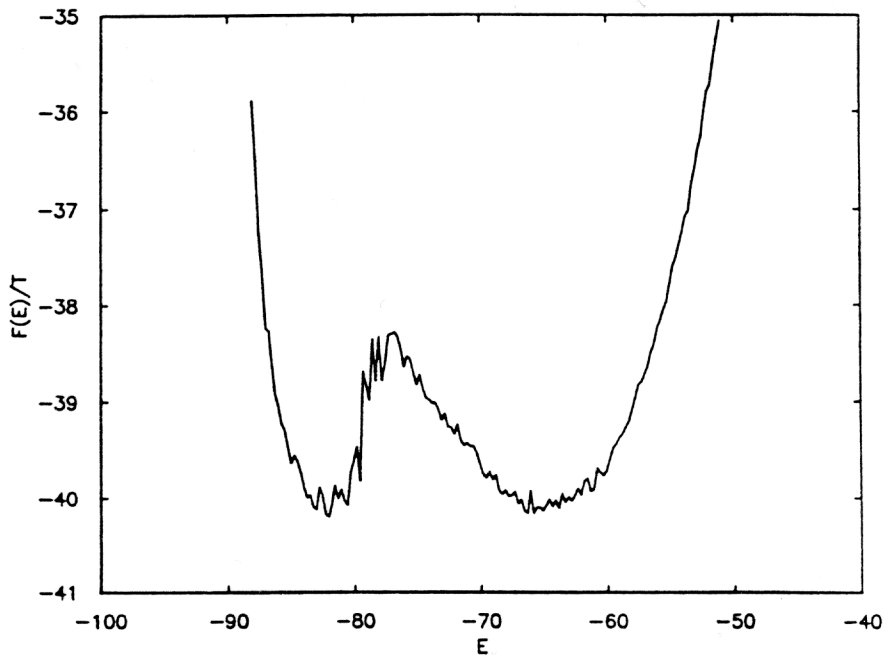

Figure 11: The relative free energy of BPTI at different energy levels E at the transition temperature (T = 0.55).

The result in Figure 11 indicates that there is a free energy barrier between the unfolded state and folded state of the protein BPTI. Because of this barrier, a jump must appear in the transition from unfolded state to the native state. In the previous section, it was pointed out that a two-state transition is the favored course leading to a unique folded structure. It is seen that the protein BPTI, under a properly designed potential, has the statistical mechanical characteristics which lead it to fold to its unique native structure. This result coincides with the folding behavior of model polypeptides of optimized sequences (see Section III). The similarity in the folding behavior of the protein and the model polypeptide with an optimized sequence would suggest that protein sequences are not random but optimized. Indeed, theories proposed by a number of authors[25–27] suggest that protein sequences are optimized in biological evolution. Obviously, a criterion for the optimal selection of a protein sequence must be that the resulting protein is able to fold to its unique native structure.

An important issue in modeling proteins is how the conformations in the modeled "native state" of the protein determined from the simulation are related to the known crystal structure of the protein. The closeness of a conformation to its native structure can be measured by the ratio, denoted as Q, of the number of native contacts in the conformation to those in the crystal structure. By definition, the Q value of the crystal structure is unity, and, for other conformations, the larger the Q value of a conformation, the closer the conformation is to the crystal structure. There is a linear

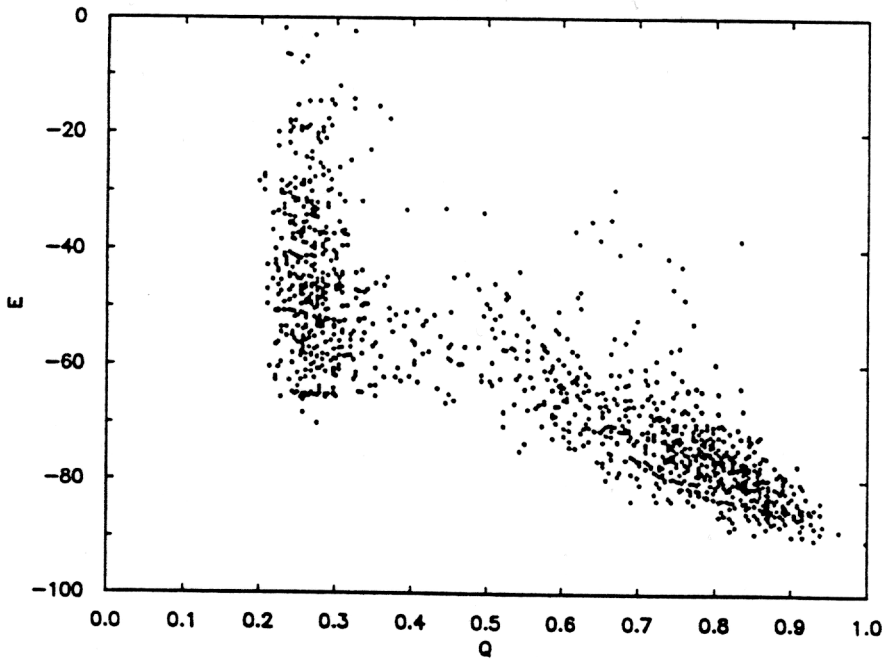

Figure 12: The correlation between the energies E and the fractions of native contacts Q among nonbonded pseudo-atoms in a sample of the conformations of BPTI.

relationship between this Q measure and the rms deviation measure used to determine the similarity of two structures (M.-H. Hao and H.A. Scheraga, unpublished observations). In Fig. 11, the states of the protein are defined by the energy of the molecule. To determine the closeness of the states of the protein to the crystal structure, one only needs to examine the relation between the energy and the Q value of the protein. Figure 12 shows the correlation of the energies and the Q values of an ensemble of conformations of BPTI generated in an MC run. The general trend of this diagram clearly indicates that, the lower the energy of a structure, the larger is its Q value. Since the "native state" in Figure 11 has a Q value between 0.8 and 1, this state is therefore very close to the crystal structure of BPTI.

V. Conclusions

In this paper, we have presented a computational approach for studying the statistical mechanics of protein folding. It is demonstrated that this procedure is very powerful for determining the characteristics of the folding transitions of proteins. Traditional MC simulations are suitable for studying some folding processes of proteins. However, when a polypeptide fails to be folded in a simulation, it may be due either to the limitation of the ability of the algorithm or to the intrinsic nature of the molecule. Traditional simulations cannot resolve such issues. The statistical mechanical

approach developed in this work treats the folding problem in a more fundamental way: it determines the density of states of the protein, identifies the characteristics of the folding transition, thereby elucidating the foldability of the protein. To a sufficient extent, this approach separates the limitation of our folding algorithm from the intrinsic foldability of a protein model and enables us to distinguish good models from bad models. This approach opens a new way for studying the problem of protein folding and will contribute to our theoretical understanding of the problem as well as to practical applications, e.g., to determine whether a potential function is good enough for predicting the native structures of known proteins and to design new polypeptide sequences that fold to unique stable structures.

The results of this paper clearly demonstrate how computer simulations can help us in understanding fundamental phenomena in biological systems, in this particular case, the folding of a protein to its biologically active state. Simulation studies of biological macromolecules are usually very computationally demanding. The availability of supercomputers not only shortens the time needed to carry out a simulation, but also motivates us to design more complicated simulations to attack new problems which would not have been practical on smaller computers.

Acknowledgement

This work was supported by grants from the National Institutes of Health (GM-14312) and the National Science Foundation (DMB-90-15815). Support was also received from the Association for International Cancer Research and the Cornell Biotechnology Center. The simulations in this work were carried out on the KSR parallel computer at the Cornell National Supercomputer Facility, which is a resource of the Cornell Theory Center, which is funded in part by the National Science Fundation, New York State, the IBM cooperation, and members of its Corporate Research Institute. The KSR facility was funded in part by the National Institutes of Health.

References

1. K.A. Dill, Biochemistry, **24**, 1501 (1985).
2. J.D. Bryngelson, P.G. Wolynes, Proc. Natl. Acad. Sci. USA, **84**, 7524 (1987)
3. E.I. Shakhnovich and A.V. Finkelstein, Biopolymers, **28**, 1667 (1989).
4. T. Garel and H. Orland, Europhys. Lett., **6**, 307 (1988).

5. E.I. Shakhnovich, A.M. Gutin, Biophys. Chem., **34**, 187 (1989).

6. J.D. Bryngelson, P.G. Wolynes, Biopolymers, **30**, 177 (1990)

7. J. Skolnick and A. Kolinski, J. Mol. Biol, **221**, 499 (1991).

8. A. Kolinski, M. Milik and J. Skolnick, J. Chem. Phys., **94**, 3978 (1991).

9. E. Shakhnovich, G. Farztdinov, A.M. Gutin, and M. Karplus, Phys. Rev. Lett., **67**, 1665 (1991);

10. P.E. Leopold, M. Montal, and J. N. Onuchic, Proc. Natl. Acad. Sci. USA, **89**, 8721 (1992).

11. M. Fukugita, D. Lancaster and M. G. Mitchard, Proc. Natl. Acad. Sci. USA, **90**, 6365 (1993).

12. C. J. Camacho and D. Thirumalai, Proc. Natl. Acad. Sci. USA, **90**, 6369 (1993).

13. A. Kolinski and J. Skolnick, J. Chem. Phys., **97**, 9412 (1992).

14. M.-H. Hao, H.A. Scheraga, J. Phys. Chem., **98**, 4940 (1994).

15. M.-H. Hao, H.A. Scheraga, J. Phys. Chem., **98**, 9882 (1994).

16. M.-H. Hao, H.A. Scheraga, J. Chem. Phys., **102**, 1334 (1995)

17. J. Lee, Phys. Rev. Lett, **71**, 211 (1993); Erratum **71**, 2353 (1993).

18. B. Berg and T. Neuhaus, Phys. Rev. Lett., **68**, 9 (1992).

19. B. Berg and T. Celik, Phys. Rev. Lett., **69**, 2292 (1992).

20. J.I. Siepmann and D. Frenkel, Mol. Phys., **75**, 59, (1992).

21. J. J. de Pablo, M. Laso and U.W. Suter, J. Chem. Phys, **96**, 2395 (1992).

22. D.D. Frantz, D.L. Freeman, J.D. Doll, J. Chem. Phys., **93**, 2769 (1990)

23. C.J. Tasi, K.D. Jordan, J. Chem. Phys., **99**, 6957 (1993).

24. A. Huller, Z. Phys., **B 93**, 401 (1994).

25. E.I. Shakhnovich, A.M. Gutin, Nature, **346**, 773 (1990).

26. A. Sali, E.I. Shakhnovich and M. Karplus, J. Mol. Biol., **235**, 1614 (1994).

27. H.S. Chan and K.A. Dill, J. Chem. Phys., **99**, 2116 (1994).