Editor: Muhammad Sahimi, moe@iran.usc.edu

COMPUTER SIMULATIONS OF PROTEIN FOLDING

By Leili Javidpour

By studying the atomistic interactions involved, computer simulation has greatly helped scientists understand fundamental aspects of protein folding. Here, we review the advantages and shortcomings of some current computer simulation methods.

Computer simulations have helped scientists greatly understand the fundamental features of protein folding. In fact, much of our present knowledge about protein folding has been discovered or verified by computer simulations. We greatly rely on these simulations, because other experimental apparatuses can't reveal protein dynamics with enough spatial and time-interval resolution.[1] Additionally, computer simulations act as a flexible laboratory to test theoretical models, or even inspire us with their results, so that we're able to develop more realistic theories for different aspects of protein folding.

However, there are many ways that computer simulations still seem "limited" to scientists, and they aren't always the best option for finding out specific information about the protein-folding process. In this article, I provide some background information on proteins and what we know about the protein-folding process. I then discuss the role (and limitations) of computer simulations in discovering or verifying information regarding protein-folding theories, and consider when it's best to use computer simulations versus other scientific resources and approaches.

Protein Folding

Proteins are a large class of biomolecules that exist in nature. They're involved in nearly every function and mechanism happening in living cells. They even contribute to the production of new protein molecules. These biopolymers consist of 20 different amino acids as their building blocks. Amino acids have a carbon atom named alpha carbon as their core, in addition to an amide group (NH_3), a carbonyl group (COOH), and a side-chain group, all attached to the alpha carbon. Different side-chain groups define different amino acids. The simplest structure for the side chain is a single hydrogen atom, relating to the amino acid known as Glycine.[2,3]

An amide group from one amino acid and a carbonyl group from another could release a water molecule and form a chemical bond between the two amino acids. In the same way, a long chain of amino acids could form. This is, in fact, the way proteins are. A protein's amino acid sequence is also known as its *primary structure*—that is, the order in which amino acids are attached to each other, starting from the amino acid with a dangling amide group and ending with the amino acid that has a dangling carbonyl group. These two end residues (amino acids) are referred to, respectively, as the protein's N-terminus and C-terminus.

Proteins are biologically active in a special molecular configuration, with a special geometrical shape and exposed active parts that are the protein's so-called "folded state" or "native state," also known as their *tertiary structure* (see Figure 1). The process in which a protein begins as a nonstructured chain and finally adopts its folded state is called *protein folding*. Researchers have found the native structure in the atomic resolution of many proteins by using x-ray crystallography and nuclear magnetic resonance (NMR), for example. The Protein Data Bank[4] (see www.pdb.org) contains a database of these known structures.

Some common structures exist in different proteins' folded state. We can refer to these as *secondary structures*, and they include helices (α-helices, for example), beta structures, and turns. Helices and beta structures are stabilized mainly by a series of hydrogen bonds between carbonyl groups and amide groups of different amino acids in the protein sequence (see Figure 2). These structures are, in fact, from the classes of compact structures that a polymer could adopt.[5]

Why Study Protein Folding?

The study of protein folding is an important and highly active field of research. It's important to study this process's potential problems for many reasons. The most obvious reason is that a misfolded protein wouldn't have its biological functionality. The misfolding of proteins could happen normally in cells, and in fact there are some mechanisms for recycling misfolded proteins. However, if protein

Copublished by the IEEE CS and the AIP
1521-9615/12/$31.00 © 2012 IEEE

misfoldings happen much more frequently than usual, they could cause dangerous diseases such as Alzheimer's, Parkinson's, and prion diseases.[6]

On the other hand, if we know how a protein sequence folds, then we can design some proteins to have a special structure and do a special task in living cells. This field is usually referred to as *protein engineering*. This field actually has a good success rate for smaller proteins. However, the interactions in larger proteins can be quite complicated and difficult to predict, so there's much need for progress in this area.[7]

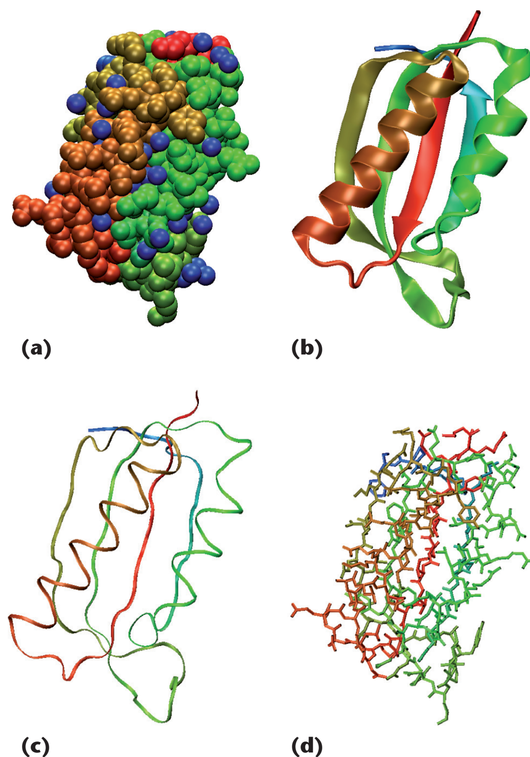

Figure 1. Example of a protein native structure, shown in different usual views. The protein's Protein Data Bank (PDB) ID is 1ris. (a) The protein's atomic structure. (b) A different view (known as the cartoon view) that shows the secondary structures, so that the protein structure is easier to see. (c) The ribbon view. (d) The bonds view, which shows all of the chemical bonds between the protein atoms. I produced all of these images using Visual Molecular Dynamics (VMD) software, and VMD is developed with National Institute of Health (NIH) support by the Theoretical and Computational Biophysics group at the Beckman Institute, University of Illinois at Urbana-Champaign (www.ks.uiuc.edu/Research/vmd).

Information Learned in the Lab and from Computer Simulations

Before a famous series of experiments by Christian Anfinsen and his colleagues,[8] people thought that protein folding was something that could only happen with living cells and organisms, and that it couldn't take place in a laboratory. Anfinsen showed in his experiments that after denaturation, proteins could gain their folded state in a well-designed experimental setup that's compatible with the physiological conditions regarding the solvent's temperature and presence of hydrogen (pH), for example. He concluded that all the information needed for protein folding is present in its amino acid sequence. Then several others in this field validated this conclusion.

I should briefly mention some of the more common interactions that stabilize proteins' native structure. One of the most important interactions is hydrogen bonding between amide and carbonyl groups of different amino acids. Hydrogen bonds have a strength of about 5 kilocalorie per mole (kcal/mol), which is nearly equivalent to 8 $k_B T$ at room temperature, where k_B is the Boltzmann constant and T is the temperature. Of course, a bond's strength depends on many different parameters, including neighboring atoms and solvent conditions. But its total strength is high enough to make it stable at room temperature.

Secondary structures are also formed by a group of such hydrogen bonds, and these structures are therefore energetically more stable. Hydrogen bonding can also occur between side chains. Other interactions in proteins are usually related to side chains. In fact, side chains are extremely important in determining protein structure, because they can greatly affect hydrogen bonding networks in a proteins' backbone, which also affects secondary and tertiary structures in the simplest way (by their volume size, for example, that could exclude other atoms). Different amino acid side chains could be polar, hydrophobic, or charged. These properties will influence their binary interactions. As an example, two hydrophobic side chains can be entropically attracted to each other in an aqueous solution, or two oppositely charged side chains will electrostatically attract each other. Also, some special amino acids have unique properties, such as cysteine amino acids, that are able to form disulfide bonds with each other.

Protein folding from an unstructured amino acid sequence occurs as a complicated process of forming a special interaction among many different possible interactions on the amino acid chain. Really the phenomenon is a probabilistic one—namely, a protein sequence won't always fold to the folded state, but it will fold with a probability that depends on the sequence itself, the solution's pH (that could in turn change the stability of different interactions in proteins), and the solution's temperature. This is why studying protein folding has become so closely related to its atomistic study. As a major consequence, computer simulations of proteins with atomistic and near-atomistic resolution have

become powerful tools for study. An interesting fact, by the way, is that natural proteins are much more stable than random sequences of amino acids—namely, they fold with a higher probability. This means that we can design some special protein sequences to have a high stability.[9]

We can also study proteins' dynamics with statistical mechanical tools. It's generally accepted that small proteins have a two-state folding process. For such proteins, we can define a transition-state ensemble and use these states for analyzing protein-folding dynamics. Larger proteins with different secondary structures and subdomains could have more complicated folding dynamics, but they still might have one or more intermediates that are stable enough to be detected in experimental measurements, for example.[10]

Proteins' folding speed is another important parameter. The formation of secondary structures is important in the process of folding, and might happen at initial stages. Some secondary structures might fold faster than others—for example, commonly α-helices fold much faster than β structures. So for computer simulations of protein folding, α proteins are favored because of their higher folding speed. The folding time of a protein depends on multiple parameters: larger proteins usually need more time to fold, and the type of secondary structures in the folded state could also affect the protein's folding time. However, generally speaking, proteins' lowest folding time is usually in the order of a microsecond.[11]

Folding temperature—usually defined as the temperature at which maximum heat capacity occurs—is another useful quantity. Above this temperature, unfolded states with high conformational entropy are dominant, and below this temperature, folded states with low potential energy are dominant. More stable proteins can retain their folded state at higher temperatures, and hence have a higher folding temperature. In some experimental literature, this temperature is also referred to as a protein's *melting temperature*.

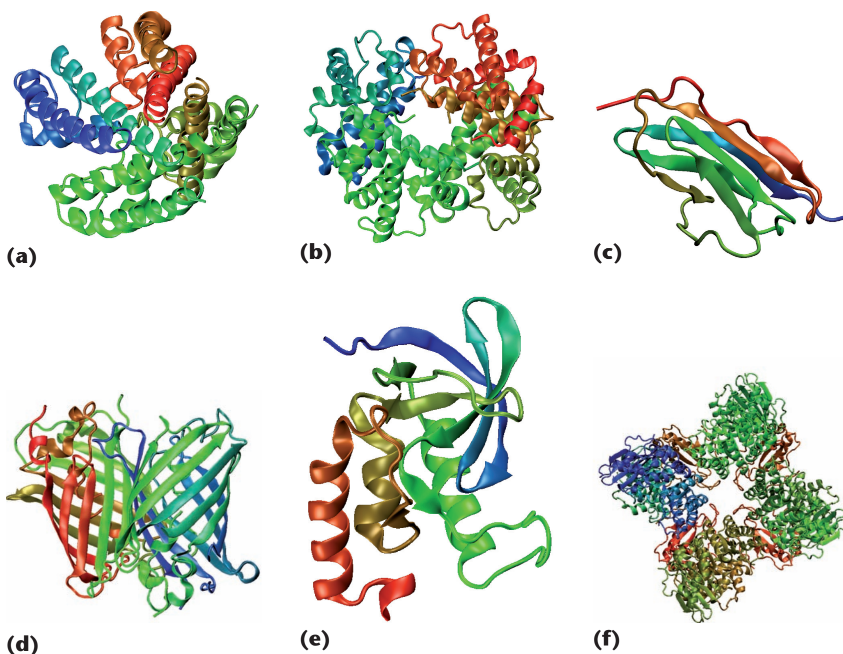

Figure 2. Proteins can have many different structures. Regarding secondary structures, some of them could be composed primarily of helices, such as (a) and (b), or beta structures, such as (c) and (d). Others, such as (e), are usually a mixture of these two main classes of secondary structures. Some structures—(a), (b), (d), and (f)—are in fact quaternary structures, in which a combination of amino acid chains, each folded to its tertiary structure, are assembled together to form a larger complex. The PDB IDs of the structures are (a) 1bl8, (b) 2hhd, (c) 1tit, (d) 1gfl, (e) 1eyd, and (f) 8ruc. I produced all of the images using VMD software.

Another important point is that proteins usually bury their hydrophobic side chains in their native state's interior, whereas polar side chains are probably on the folded protein's surface, and hence exposed to water molecules. It's believed that a common stage in protein folding is its *hydrophobic collapse*, in which hydrophobic amino acids attract each other and begin to form the protein's hydrophobic core.

It's also valuable to mention a concept known as the *existence of cores* in protein folding. In this view, some special set of contacts (attractive interactions) are considered crucial in protein folding. Namely, we know that their formation is extremely important and necessary in the folding process. For such discussions, ϕ-analysis is a valuable tool. In this method, we mutate the initial protein sequence's special residue(s), and then we analyze the change in the folding dynamics, mainly in the protein's folding rate. If the folding rate is changed, it's interpreted as a free-energy change of the transition-state ensemble, and in turn that's interpreted as being important for that special set of mutated residues in the transition-state ensemble.

It's also crucial to mention that protein folding could differ in significant ways for in vivo and in vitro situations. For example, macromolecular crowding in cells could change proteins' folding dynamics, or chaperons and chaperonins could help proteins fold and prevent them from being aggregated.[12]

Computer Simulations

Now that I've explained a bit about protein folding, you can see that it's a very complex process to study, and there are many approaches we can use in an attempt to learn more about this process. For computer simulations, we can use many different models of proteins to study their folding. As with any other scientific field, depending on our available computational power, we can use different levels of accuracy for the model that we plan to use. However, to study protein folding with the best models, we would need quantum mechanical features, and that means that we would need computational power that's currently unreachable to simulate a small amino acid sequence for even a nanosecond of real time. So for now we must ignore some level of reality in our model to be able to study the problem. For more accurate and quantitive results, we must use more exact models. But if we're studying a general behavior for many different kinds of proteins, then it's likely that we could perform our simulations by using a greatly simplified model of proteins.

Models for Protein Structure

Lattice models are from the simplest models of proteins' structure. In these models, each amino acid is usually represented by only one particle or bead, and it's restricted to move only on a specific lattice site. The lattice could be in two or three dimensions, and we generally use a square or cubic lattice. For such models, the most-used interaction between the beads is based on the hydrophobic and polar (HP) model that we'll describe later. Interestingly, some general aspects of protein folding can even be captured by such two-dimensional HP lattice models.

In a more realistic class of nonlattice models, beads aren't restricted to a lattice. Other models exist that consider 2, 3, 4, or even 6 beads and more for a single residue. In such models, the residues' atomistic detail isn't taken into account, and a group of atoms are shown by a bigger united atom. Because of the side chains' importance in proteins' behavior, all of such models have one bead or more for representing the side chain of each amino acid. Some of them even consider different numbers of beads for the different side chains that they're modeling, depending on the size and/or special interactions of those side chains. Obviously, the more simplified models in this section will lose the more conformational details of the amino acid sequences that they're modeling. For example, models having less than 3 beads for each residue in the backbone wouldn't be able to measure Ramachandran angles (ϕ and ψ) for different protein configurations that they're modeling. We could use these angles to realize with more accuracy proteins' different secondary structures. So if for a special study we need these conformational details, we should have at least 3 beads for the backbone part, and possibly at least 1 bead for the side-chain part of each amino acid.

If a nonlattice model is well designed for a special protein, we can even use it to detect the difference in folding dynamics between mutants of an amino acid chain (in a ϕ-analysis simulation, for example). However, the most realistic models for proteins are full atomistic models, which might include hydrogen atoms. Such models are also useful for problems that need accurate analysis of interactions between two different molecules, one of which might be a protein or an enzyme with an active site. To benefit this detailed representation of a protein structure and obtain protein-specific results, we'll also need accurate force fields for the interaction between different binary (or more) sets of atom types.

Another important feature in the proteins' simulation is the way that we model solvent molecules. For a more realistic simulation, we need to carry out a simulation that considers solvent molecules explicitly. In fact, solvent molecules have many different—and significant—effects on protein dynamics. Their entropy, their differing interactions with different atoms and side chains, and their random forces all could influence proteins greatly. On the other hand, an explicit solvent simulation could also be very expensive. Thus, implicit solvent models are quite popular among the scientific community, and researchers use a few tricks to mimic different effects of solvents on proteins implicitly (which we discuss in the next section).

Interactions

For the interaction between beads in simpler models, the usual approach is to split all beads (and, more importantly, the side-chain beads) into two different groups of hydrophobic and polar, in accordance with the HP model. Then the interaction between different bead types is defined and used in the simulations. Although in an aqueous solution, hydrophobic particles are effectively attracted to each other, in HP models the most attractive interaction relates to the interaction between two H beads. In contrast, the interaction between other beads might be considered simplistically, and include only a purely repulsive part to consider the beads' excluded volume effect. In fact, this is the most-used approach for considering

implicitly solvent molecule effects. Some more complicated approaches are also sometimes used to account for such solvation effects, one of which is the *solvent-accessible surface area*. Of course, these models can't include all the important effects of solvent molecules, but they surely account for one or more of the most important effects.

To represent the interaction between two beads, we can use different simple potentials, including hard-core repulsion and different kinds of Lennard-Jones potentials (which are models that approximate the interaction between a pair of neutral atoms or united atoms). Sometimes, when we're entering a specific important interaction into the model, such as hydrogen bonding or electrostatic interactions, we add a simplified potential to mimic their effect between specific particles.

We use the most accurate potentials in the full atomistic models. We can obtain such force fields using quantum mechanical calculations, and we can also use the results of experimental measurements from the structure of biomolecules as feedback for the force fields' accuracy. There are different, frequently used force fields such as AMBER potentials (and many others), each of which might be more appropriate for a particular class of macromolecules.

For nearly all of the aforementioned potentials, however, we can use a Go model.[13] In such potentials, we use our knowledge of the studied protein's native structure, along with some special interactions that appear in the folded state—known as *native contacts*—that aren't allowed to break after their formation. But the other strong attractive interactions between system particles are allowed to break if they formed during the simulations. It's obvious that such a model helps a protein achieve its folded state after a while. There are, of course, strong discussions about verifying the use of Go potentials. The main reasoning is based on the aforementioned existence of cores for protein folding. These potentials can greatly reduce the computational power needed for simulations, which is why they're widely used. Thus, for many problems, simulations yield much better results than experiments. However, if we're going to simulate proteins in special situations where a theory doesn't completely explain our experimental observations, using a non-Go potential might be more adequate.[14]

Computer Simulation Methods and Analysis

Molecular Dynamics (MD) and Monte Carlo (MC) methods are the two main classes for the simulation of proteins, similar to other molecular systems.[15] In an MC simulation, it's sometimes possible to sample the configurations of the protein's phase space with a lower computational effort. However, in MC, we can't get any information from the simulation itself about the real time needed for conformational changes that are happening in the simulation. Still, it's possible that the ratio of MC time steps for different processes to happen might be reliable and compatible with the ratio of their happening in real time. Thus, MC could be an effective and useful method for studying proteins' free-energy surface, the effect of a special parameter on folding or unfolding rates, or any other problem that doesn't need to know the system's exact dynamics.

In an MD simulation, we numerically integrate Newtonian equations of motion for all particles in the protein structure model that we defined. Forces between different particle types should be defined, too. The initial state we choose could be a random coil, or even the protein's folded state, depending on our simulation's purpose. It should satisfy all conditions of a real protein chain, such as the chain's continuity, its amino acids' chirality, or the transconfiguration of bonds between neighboring amino acids. The initial velocities are usually chosen from a Maxwell-Boltzmann distribution at the simulation's temperature. The time step for numerical integration is usually about a femtosecond ($10^{-15}s$) or more, and usually less than $10^{-13}s$. The time step considered should be about one or two orders of magnitude less than the system's smallest time scale. Potentials used for modeling chemical bonds and angles between the atoms usually define the smallest time scales in an MD.

As we discussed previously, solvent molecules have many different effects on protein dynamics. Such effects can arise from random forces of solvent molecules to the protein atoms. In implicit solvent simulations, a usual approach for modeling this effect is to use the Langevin equation.[9]

In contrast to MC, but with the same computational effort, MD might be restricted to a smaller region of phase space. However, except for the time-integration error that we ignore regarding the equations of motion, the model's dynamics that we're simulating are its real dynamics. The maximum time scale we can reach in an MD depends on the complexity of the potentials we're using. More accurate force fields will need more computational power.

It's interesting to know that the first complete simulation of the folding of a small protein occurred in 1998 for

a villin headpiece subdomain—which is a 36-residue peptide with explicit solvent molecules—for 1 microsecond on a massively parallel supercomputer at that time.[16] Other big groups have since made notable strides using supercomputers to simulate proteins.[17] Using such powerful clusters could help us greatly improve our understanding of protein folding, especially if we're able to use state-of-the-art protein models for the time scales that are compatible with what's needed for their folding.

But what if we didn't have access to such supercomputers? We might use simpler models, and really there are still enough unknown features that we can study on some level without exact models. Thus, we should balance the accuracy of the model that we use with our available computational resources. We should also consider the time scales needed to study the phenomena that we're interested in. For example, if we're observing a single folding event of a small peptide with an α-helix native state, a simulation length of tenths of a nanosecond might be sufficient. But if we're interested in some other phenomena on a time scale of hundreds of nanoseconds, then we should try simpler models to catch our needed simulation times.[18,19]

Discontinuous molecular dynamics—also known as discrete molecular dynamics (DMD)—is a simple model along these lines that has been successful in explaining many different protein behaviors, including fibril formation.[14,18–20] In this method, we use discrete potentials, and hence the forces between particles are zero or infinite. Thus, the system state's change is caused by collisions between particles—that is, when the first two particles reach a distance that's at the next nearest step in their potential. This is why DMD is also called an *event-driven simulation*. In contrast, usual MD is a time-driven simulation, which means that the simulation stops after a constant time interval, so the simulation proceeds based on time. Used potentials in DMD can have as many steps as we wish, but we should bear in mind that more steps in potentials means more events, and thus more stops in the simulation, which requires more computational power to finish a constant simulation time.

In addition to the traditional methods of MC and MD at a special temperature, there are many newer techniques that could help us sample the phase space with a smart choice of different simulation conditions (such as temperature). For example, some kind of acceptance or rejection for such changes in the simulation condition could help us greatly improve our accessible regions in complex biomolecules of proteins' phase space. These types of methods are usually a combination of MD and MC methods. Using such methods might help us greatly improve the accuracy of results that we get from our simulations. Some examples of such techniques include replica-exchange molecular dynamics, umbrella-sampling methods, and a weighted-histogram-analysis method.[13,19] These methods also could have some recipes for measuring the average of different statistical mechanical quantities from different simulation cases.

I should also mention that many useful software systems exist for protein simulation, examples of which include Gromacs and Not (just) Another Molecular Dynamics program (NAMD). Some of these tools have the ability to work on parallel computers with good efficiency, and some have been developed by large groups and are highly optimized. Thus, if the model you're going to use for your protein is compatible with these software programs, then you can use them. However, if you're going to use a special model or if you need to have great control on the code you're using to be able to change the model or your measurements in any way that you wish, then it might be better for you to develop the code for your specific problem.

What Results Can Simulations Provide?

Simulations are in fact a laboratory for us to verify our theories or even to help us develop new insights. Although the time and spatial resolution in a computer simulation is great enough, we can measure many different system-model properties. Here, I mention some of the most common measurements that are usually the most helpful in analyzing protein-folding dynamics.

For smaller proteins, we can assume a two-state folding process. In such cases, one of the useful parameters to measure in a simulation is the protein's folding rate—that is, the inverse of the average time needed for a completely unfolded protein to fold to its native state. Another useful parameter is the protein's unfolding rate—the inverse of the time needed for a folded protein to unfold. If the studied protein has some stable intermediates for its folding, then the average times for changing between the unfolded state, folded state, and any intermediates will help us analyze the protein's dynamics.

Evaluating the simulated proteins' free-energy surface using advanced techniques could also be quite useful, and this lets us measure many different statistical quantities. For example,

we could then study various parameters' effect on proteins' stability and folding dynamics.

As another example, we could study the simulated protein's transition-state ensemble. Some groups, for example, could define a state for a two-state folding protein to be in this ensemble if it could fold and unfold with the same probability of 1/2. Then we could analyze this ensemble, to see whether there's a folding nucleus for the studied protein.[10]

Many groups have studied the protein-folding problem. Using computer simulations to study the problem is a great help, except for the present shortcomings with regards to limitations of time and space resolution. However, by using the high computational power of computer clusters that are being developed in more institutes every year, over time the community will be able to extend our knowledge and ability to predict the folded state of an amino acid sequence to larger and more complex proteins.

Acknowledgments

I thank Muhammad Sahimi for his critical feedback, which greatly improved this article.

References

1. B. Schuler and W.A. Eaton, "Protein Folding Studied by Single-Molecule FRET," *Current Opinion in Structural Biology*, vol. 18, no. 1, 2008, pp. 16–26.
2. A.R. Fersht, "Structure and Mechanisms in Protein Science: A Guide to Enzyme Catalysis and Protein Folding," 9th ed., Freeman, New York, 2008.
3. C. Branden and J. Tooze, *Introduction to Protein Structure*, 2nd ed., Garland Publishing, New York, 1998.
4. H.M. Berman et al., "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, 2000, pp. 235–242.
5. J.R. Banavar et al., "Unified Perspective on Proteins: A Physics Approach," *Physical Rev. E*, vol. 70, no. 4, 2004; doi:10.1103/PhysRevE.70.041905.
6. C.M. Dobson, "Protein Folding and Misfolding," *Nature*, vol. 426, no. 6968, 2003, pp. 884–890.
7. K.A. Dill et al., "Protein Folding Problem," *Ann. Rev. Biophysics*, vol. 37, no. 1, 2008, pp. 289–316; doi:10.1146/annurev.biophys.37.092707.153558.
8. C.B. Anfinsen, "Principles that Govern the Folding of Protein Chains," *Science*, vol. 181, no. 4096, 1973, pp. 223–230.
9. S. Takada, Z. Luthey-Schulten, and P.G. Wolynes, "Folding Dynamics with Nonadditive Forces: A Simulation Study of a Designed Helical Protein and a Random Heteropolymer," *J. Chemical Physics*, vol. 110, no. 23, 1999; doi:10.1063/1.479101.
10. L. Mirny and E. Shakhnovich, "Protein Folding Theory: From Lattice to All-Atom Models," *Ann. Rev. Biophysics and Biomolecular Structures*, vol. 30, 2001, pp. 361–396.
11. J. Kubelka, J. Hofrichter, and W.A. Eaton, "The Protein Folding Speed Limit," *Current Opinion in Structural Biology*, vol. 14, no. 1, 2004, pp. 76–88.
12. F.U. Hartl and M. Hayer-Hartl, "Concepts of Protein Folding In Vitro and In Vivo," *Nature Structural & Molecular Biology*, vol. 16, 2009, pp. 574–581.
13. C. Clementi, H. Nymeyera, and J.N. Onuchic, "Topological and Energetic Factors: What Determines the Structural Details of the Transition State Ensemble and 'En-route' Intermediates for Protein Folding? An Investigation for Small Globular Proteins," *J. Molecular Biology*, vol. 298, no. 5, 2000, pp. 937–953.
14. L. Javidpour and M. Sahimi, "Confinement in Nanopores Can Destabilize Alpha-Helix Folding Proteins and Stabilize the Beta Structures," *J. Chemical Physics*, vol. 135, no. 12, 2011; doi:10.1063/1.3641482.
15. M.P. Allen and D. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, 1987.
16. Y. Duan and P.A. Kollman, "Pathways to a Protein Folding Intermediate Observed in a 1-Microsecond Simulation in Aqueous Solution," *Science*, vol. 282, no. 5389, 1998, pp. 740–744.
17. V.A. Voelz et al., "Molecular Simulation of Ab Initio Protein Folding for a Millisecond Folder NTL9 (139)," *J. Am. Chemical Soc.*, vol. 132, no. 5, 2010, pp. 1526–1528.
18. H.D. Nguyen and C.K. Hall, "Molecular Dynamics Simulations of Spontaneous Fibril Formation by Random-Coil Peptides," *Proc. Nat'l Academy of Science*, Nat'l Academy of Sciences, vol. 101, no. 46, 2004, pp. 16180–16185.
19. L. Javidpour, M.R. Rahimi Tabar, and M. Sahimi, "Molecular Simulation of Protein Dynamics in Nanopores. II. Diffusion," *J. Chemical Physics*, vol. 130, no. 8, 2009; doi:10.1063/1.3080770.
20. L. Javidpour, M.R. Rahimi Tabar, and M. Sahimi, "Molecular Simulation of Protein Dynamics in Nanopores. I. Stability and Folding," *J. Chemical Physics*, vol. 128, no. 11, 2008; doi:10.1063/1.2894299.

Leili Javidpour is a postdoctoral associate at the Institute for Research in Fundamental Sciences. Her research interest is in computational biophysics. Javidpour has a PhD in physics from the Sharif University of Technology. Contact her at javidpour@ipm.ir or javidpour@alum.sharif.edu.

cn *Selected articles and columns from IEEE Computer Society publications are also available for free at http://ComputingNow.computer.org.*