Motion Planning for Proteins: Biophysics and Applications*

Lydia E. Kavraki

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 1.0†

Abstract

This module introduces the concept of free energy and potential fields in the context of protein conformation spaces and motion planning. It then provides examples of applications of motion planning techniques to problems from structural computational biology.

Topics in this Module

- Free Energy and Potential Functions (Section 1: Free Energy and Potential Functions)
  - Free Energy (Section 1.1: Free Energy)
  - Potential Functions (Section 1.2: Potential Functions)
- Applications of Protein Motion Planners (Section 2: Applications of Roadmap Methods)
  - Kinetics of Protein Folding (Section 2.1: Kinetics of Protein Folding)
  - Protein-Ligand Docking Pathways and Kinetics (Section 2.2: Protein-Ligand Docking Pathways and Kinetics)

As we suggested in Robotic Motion Planning and Protein Motion, the main difference between modeling a macroscopic robot arm and a protein chain is that the protein is subject to forces resulting from differences in free energy between its states. The protein's **conformation space** does not consist only of colliding and non-colliding structures, but of structures on a continuum of free energy values. In this module, we will provide a very brief overview of free energy as it relates to protein structures, and then give some examples of how path planning techniques have been applied to solving problems in structural biology.

1 Free Energy and Potential Functions

1.1 Free Energy

In other modules, we have introduced the concept of a **native conformation** for any given protein, that is, the conformation of the protein that is observed, or expected to be observed, under physiological conditions of temperature, pH, and ion balance. What distinguishes this structure from other structures is that it has the minimum **free energy** of all accessible conformations. There are several different definitions of free energy depending on how the system is defined (for example, whether it is allowed to change in temperature, volume, and/or pressure). One common definition, applicable when temperature and volume are constant, is the Helmholtz Free Energy:

*Version 1.20: Mar 22, 2007 4:18 pm -0500
†http://creativecommons.org/licenses/by/1.0

$$F = U - TS$$

Figure 1: Helmholtz free energy

The quantity U is the **internal energy** of the system, both kinetic and potential, although for our purposes, we will usually think of changes in U as resulting from changes in potential energy. T is the absolute temperature of the system, and S is the **entropy** of the system, which is very difficult to predict computationally. Entropy is a measure of the number of accessible states to a molecule in a given state, and corresponds to a notion of disorder. In general, the probability of observing a particular state of a system (such as a protein in solution) **increases** exponentially as the free energy **decreases**, in accordance with the Boltzmann distribution:

$$P(E) \propto exp\left(\frac{-E}{k_B T}\right)$$

Figure 2: Boltzmann-like distribution.

E is a particular free energy, kB is the Boltmann constant, and T is the absolute temperature.

In practice, because entropy is very difficult to approximate computationally, potential energy is often used instead of free energy in molecular simulations and docking procedures. When the process is driven by potential energy, this is a reasonable approximation. Some processes are entropically driven, and results are usually poor when trying to model these processes using only potential energy.

1.2 Potential Functions

Potential functions are functions used to evaluate the feasibility of a particular structure of a molecule. Ideally, this would be done with quantum mechanics, in which case the energy function could report the true energy of a particular structure. In practice, quantum mechanical analysis of molecules the size of proteins is wildly intractable. As a compromise, biophysicists have developed artificial functions based on classical physics to approximate the true energy of molecular systems. Sometimes called **potential functions** or **molecular force fields**, these functions generally accept as input a molecular conformation, in the form of Cartesian coordinates for all atoms, and output an energy value. These energy values are generally only meaningful in relative terms: They provide information on what conformations of the molecule are more or less probable than others. The lower the energy value, then the more likely the conformation is to be observed. Most molecular potential functions have the form:

$$E\left(\vec{R}\right) = \sum_{\text{bonds}} B\left(\vec{R}\right) + \sum_{\text{angles}} A\left(\vec{R}\right) + \sum_{\text{torsions}} T\left(\vec{R}\right) + \sum_{\text{nonbonded}} N\left(\vec{R}\right)$$

$\vec{R}$ is the vector representing the conformation of the molecule, typically in cartesian coordinates

Figure 3: A generic potential function.

Approximate energy functions provide the basis for molecular simulations and some protein-ligand docking procedures, among other applications. In some docking problems, a potential function is used to evaluate how likely a particular pose of a small molecule (ligand) in the binding pocket of a protein is. The internal energy of the receptor and the ligand are considered along with the interaction energy between the two. Interaction energy usually consists of the non-bonded terms found in the internal energy function, summed of all pairs of atoms (r,l), where r is an atom of the receptor and l is an atom of the ligand. If the energy function approximates what is going on well enough then the docked conformation should have minimum energy value. Some docking programs use alternative forms of scoring functions, but in all cases, the object is to find the state of the complex that has the least free energy, and therefore there is a balance between making functions fast to compute and making them reasonably approximate free energy. Potential functions may also be used in simulations to study protein folding mechanisms and kinetics.

1.3 Terms of energy functions

1.3.1 Bonds

The bond energy term corresponds to the stretching and compressing of the length of a bond. In most energy functions this term reduces bonds to simple harmonic oscillators, yielding a quadratic equation:

$$E_{\text{bonds}} = K_b(b - b_0)^2$$

where K_b is an empirically determined constant that depends on the atom types, b is the current bond's length, and b_0 is the bonds length in equilibrium, which again depends on the atom types. In this case you can think of the bond as a spring, it has an equilibrium length that it wants to remain at. If the bond length varies from the equilibrium length, the energy increases.

1.3.2 Bond Angles

The bond angle energy corresponds to changes in the angle between bonds. As with bond length, the bond angles have an equilibrium value, and any deviation increases the potential energy. Once again this can be modeled by a simple quadratic term.

$$E_{\text{angles}} = K_\theta(\theta - \theta_0)^2$$

where K_θ is an empirically determined constant, θ is the current bond angle, and θ_0 is the equilibrium angle.

1.3.3 Torsions

Torsions are created by series of three bonds, and consist of rotations of the bonds on either end with respect to the axis of the middle bond. In molecular structure certain torsional angles are preferred over others and the energy function reflects this. Usually it is described by a Fourier series expansion. The simplest being a single term:

$$E_{\text{tor}} = K_{\text{tor}} \left[1 + s \times \cos\left(n\omega\right)\right]$$

$K_t or$ is a constant, $s = +1$ or 1, n is the periodicity and ω is the angle

Figure 4: A typical torsional energy term for a potential function.

A more complicated three term expansion can also be used:

$$E_{\text{tor}} = V_1\left(1 + \cos\left(\omega\right)\right) + V_2\left(1 - \cos\left(2\omega\right)\right) + V_3\left(1 + \cos\left(3\omega\right)\right)$$

V_1, V_2, V_3 are constants, and ω is the angle

Figure 5: A more complicated torsional energy term for a potential function.

1.3.4 Van der Waals Interactions and Steric Clash

Strictly speaking, Van der Waals interactions are weak attractive interactions between atoms at an ideal separation from each other. The atoms transiently induce each other's electron distribution into complementary dipoles, allowing a weak electrostatic attraction between them. In molecular potential fields, Van der Waals attractions are usually combined with steric clash (extremely high energies due to overlapping atoms) in a Lennard-Jones potential, such as this Lennard-Jones 12-6 function:

$$E_{L-J}(i,j) = -4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right]$$

where ϵ_{ij} is a constant characteristic of the two atom types, σ_{ij} is the average diameter of the two atoms, and r is the distance between the atom centers.

Figure 6: A typical Lennard-Jones 12-6 potential.

1.3.5 Electrostatic Interactions

Electrostatic interactions are usually computed using some variant of Coulomb's Law, which assumes that atoms behave as point charges located at their centers. A typical Coulombic term looks like this:

$$E_{\text{elec}} = \frac{q_i q_j}{D r_{ij}}$$

$q_i q_j$ are the charges of the atoms, D is the effective dielectric constant, $r_i j$ is the distance between the two atoms

Figure 7: Electrostatic energy is computed using a version of Coulomb's Law.

The dielectric constant is a function of the medium through which the two charges interact. The difference between the dielectric constant of water and that of pure protein is substantial, so some models attempt to take it into account. One of the simplest assumes that the farther apart two charges are, the more likely they are to have water between them. This is called a distance-dependent dielectric, because it scales with the distance between the atoms involved.

1.3.6 Other Classes of Interactions

While all of the previous terms are almost always included in energy functions, there are a handful of other terms that are common, but not present in every function. These include hydrogen bonding, solvation and cross terms.

Hydrogen bonds (which are not true bonds in the strict, electron-sharing sense) are unusually strong electrostatic interactions, usually between a hydrogen atom and an electronegative atom such as oxygen or nitrogen. They play an important role in determining and maintaining the structure of biomolecules including proteins and nucleic acids. Some energy functions account for hydrogen bonding in the electrostatic term. Other functions include a separate hydrogen bonding term which is most often a Lennard-Jones-like 12-10 potential:

$$E_{\text{hydro}} = K_{ij} \left[\left(\frac{C_{ij}}{r} \right)^{12} - 2 \left(\frac{D_{ij}}{r} \right)^{10} \right]$$

K_{ij}, C_{ij} and D_{ij} are constants, r is the distance between the two atoms

Figure 8: A hydrogen-bonding 12-10 term for potential functions.

The solvent that a molecule is in can have a large effect on how it moves. Explicitly representing solvent molecules, however, is a computational cost that most methods try to avoid. Usually the solvent model is separate from the energy function. There are several different ways of approximating solvent interactions including the Generalized Born Model and the Poisson-Boltzmann method. Most force fields do not have an explicit solvent term.

Other terms that describe the interaction between bonds and angles, angles and torsions and so on are included in some force fields. For example to model the interaction between bonds and angles:

$$E_{\mathrm{s-b}} = K_{r\theta}\left(r - r_0\right)\left(\theta - \theta_0\right)$$

$K_{r\theta}$ is a constant, θ_0 is the equilibrium angle value, r_0 is the equilibrium bond length, r is the bond lenght θ_0 is the angle value

Figure 9: A potential energy term depending on both bond lengths and angles.

1.4 Parameters

All of the terms presented above include one or more atom-type-dependent constants, or parameters. Determining these parameters is the major problem in developing a new potential function. These parameters are typically found by fitting calculated results to experimental data. Detailed quantum analysis of small molecules may also be used to set some constants. Regardless of how it is determined, it is important to remember that all potential fields are approximations, and most are best suited for some types of proteins over others.

1.5 An Example: The CHARMM All-Atom Empirical Potential

CHARMM (Chemistry at HARvard Macromolecular Mechanics) refers to both a program for macromolecule dynamics and mechanics and the energy function developed for use in that program. CHARMM is a popular force field used mainly for the study of macromolecules. In the most recent version, the parameters were created using experimental data and supplemented with ab initio results. The CHARMM energy function has the form:

$$\begin{aligned} U\left(\vec{R}\right) &= \sum_{\text{bonds}} K_b\left(b - b_0\right)^2 + \sum_{\text{UB}} K_{UB}\left(S - S_0\right)^2 + \sum_{\text{angle}} K_\theta\left(\theta - \theta_0\right)^2 \\ &+ \sum_{\text{dihedrals}} K_\chi\left(1 + \cos\left(n\chi - \delta\right)\right) + \sum_{\text{impropers}} K_{\text{imp}}\left(\phi - \phi_0\right)^2 \\ &+ \sum_{\text{nonbond}} \epsilon\left[\left(\frac{R_{min_{ij}}}{r}\right)^{12} - \left(\frac{R_{min_{ij}}}{r}\right)^{6}\right] + \frac{q_i q_j}{\epsilon_1 r_{ij}} \end{aligned}$$

K_b, K_{UB}, K_θ, K_χ, K_{imp} are constants, b is the bond length, b_0 is the equilibrium bond length, S is the UB 1,3-distance, S_0 is the ideal UB 1,3-distnace, θ is the angle value, θ_0 is the equilibrium angle value, χ is the dihedral angle value, n is the periodicity, ϕ is the improper angle value, ϕ_0 is the ideal improper angle value, ϵ is the Lennard-Jones well depth, $R_{min_{ij}}$ is the distance at the Lennard Jones minimum, q_i and q_j are the atoms' charge ϵ_1 is the effective dielectric constant, r_{ij} is the distance between the atoms

Figure 10: The CHARMM all-atom empirical potential function

For more information on CHARMM and the CHARMM force field, please see The CHARMM website.[1]

[1] http://www.scripps.edu/brooks/charmm_docs/charmm.html

2 Applications of Roadmap Methods

2.1 Kinetics of Protein Folding

The two standard methods of simulating protein motion are molecular dynamics simulation (MD) and Monte Carlo simulation (MC). In MD, a molecule or system of molecules is given an initial set of atomic momenta, placed in a potential field, and allowed to evolve over time following Newton's equations of motion and the forces exerted on it by the field. In MC, a series of perturbations is applied to a single molecule. After each perturbation, if the estimated energy of the molecule has decreased, the perturbed conformation is used for the next step of the simulation. If the energy has increased, the perturbed conformation might be accepted, with a probability that drops off sharply as the energy change increases. Otherwise, the perturbed conformation is rejected and the previous conformation is perturbed again. Properly implemented MC or MD simulations, run for long enough, should generate a series of conformations with a Boltzmann-like distribution of structures (see the first section of this module for a reminder of what the Boltzmann distribution looks like).

The problem with these methods is that they are very slow. A single MD simulation of a few nanoseconds of motion for an average-size protein, performed on a cluster of processors, can take days, and such simulations are of limited reliability due to approximations of energy and to the extremely short time periods that can be simulated in a reasonable amount of CPU time. Simulations must be repeated to determine what a reasonable, average behavior might look like. Some protein rearrangement events take place on a scale of microseconds, milliseconds, or even seconds, so a trajectory of a few nanoseconds cannot hope to capture these low-frequency events.

The field of chemical kinetics is concerned with the rate at which chemical processes take place, and therefore, the pathways and mechanisms by which they occur. In protein biochemistry, one of the major open questions is the protein folding problem: Given a protein, and its folded (native) and unfolded (denatured) structures, what is the mechanism by which the protein folds into its native state? Currently (2006), it is possible to determine in the laboratory the rate at which a protein folds and sometimes the form of its transition state, the highest energy conformation(s) it assumes in the process of folding. These laboratory measurements can be compared to those inferred from simulation, and the quality of the simulation can thereby be indirectly estimated.

Roadmap-based algorithms to study this problem began with work by Latombe, Singh, and Brutlag in 1999 [9], in which they attempted to use a PRM to find and study protein binding pockets. This work led directly to that of Song and Amato, Apaydin and Latombe, and Singal and Pande, all presented below. Existing methods as of early 2006 are presented.

2.1.1 A PRM-Based Approach

The research group of Nancy Amato has been working on roadmap-based methods to study the process of protein folding [2][1][12][13]. They start with the known native structure of a protein, and incrementally find conformations more and more different from the native state, and build a roadmap using these conformations. The goal is to find a large ensemble of pathways between the native structure and unfolded structures, and to study these pathways and their properties. In their work, the degrees of freedom of the protein are assumed to be the ϕ and ψ backbone dihedral angles of each amino acid residue. The side chains are assumed to be rigidly attached to the backbone. In their initial work, they generated new conformations for their roadmaps by adjusting the backbone dihedral angles in the folded conformation randomly with various standard deviations. This approach worked well for very small proteins, but did not scale well.

A later sampling approach was based on counting **native contacts**. For the purposes of their method, a native contact was defined as any two alpha-carbons within 7 Å of each other in the folded state of the protein. A new conformation is generated at each step of the sampling phase of the roadmap construction by randomly perturbing some existing sample. The resulting structure was accepted with a probability as follows:

$$P(\text{accept q}) = \begin{cases} 1 & \text{if } E(q) < E_{min}, \\ \frac{E_{max}-E(q)}{E_{max}-E_{min}} & \text{if } E_{min} \leq E(q) \leq E_{max}, \\ 0 & \text{if } E(q) > E_{max}. \end{cases}$$

where q is the candidate node, $E(q)$ is the energy of the structure at q, E_{min} is an energy so low that structures with this energy are always accepted, and E_{max} is an energy so high that all structures with this energy are rejected.

Figure 11: The acceptance criterion for newly sampled conformations.

The energy, E, used in this research includes a term favoring known secondary structure contacts, and a Lennard-Jones 12-6 term as presented in the previous section. The parameters of the Lennard-Jones term are selected to favor interactions between H and O atoms, and thus hydrogen bonds. The energy thresholds for acceptance are decided by experiment. If accepted, a structure is placed in a bin corresponding to the number of native contacts present, with one bin for each possible number of native contacts from 1 to n. The procedure begins by sampling around structures in the 100% native contacts bin (initially, just the native structure). Once the next bin, with one fewer contacts, is full, samples are made by perturbing structures in that bin. Thus, the sampling generally proceeds from structures with all native contacts to structures with very few native contacts.

Once the samples are generated, an attempt is made to connect the k nearest neighbors of each node to the node itself. A series of conformations on the line connecting the two nodes are tested, and if their energy is below a threshold, the edge is included in the roadmap. The weight of the edge depends on the sequence of energies of the conformations computed along the connecting line. The probability of moving from the (i-1)th structure to the ith along the line is given by:

$$P_i = \begin{cases} e^{\frac{-\Delta E_i}{k_B t}} & \text{if } \Delta E_i > 0, \\ 1 & \text{if } \Delta E_i \leq 0. \end{cases}$$

where P_i is the probability of moving from conformation $i-1$ to conformation i along the edge, ΔE_i is the difference in energies between conformation $i-1$ and conformation i, k_B is the Boltzmann constant, and T is the absolute temperature.

Figure 12

The weight of an edge is the sum of the logarithms of the probabilities, and is intended to represent the energetic feasibility of making the transition from the conformation represented by one node to the next. This assumes that moving from one node in the roadmap to an adjacent one consists of a series of move along the edge, each associated with a probability. Note that in reality, the path taken by a protein transitioning between two structures need not be a straight line in conformation space.

Once the roadmap is computed, the shortest paths between structures can be found using Djikstra's algorithm, and the folding paths can be extracted and studied. In particular, the order of secondary structure formation can be predicted by a consensus method. The order of secondary structure formation is determined

for all paths in the roadmap from unfolded to folded structures, and the most common order is predicted as the true order of formation, which is a coarse, high-level expression of the folding mechanism. On a set of proteins used to test their method, the predicted formation order matched laboratory-determined formation order in all cases where it was available. Because of the coarseness of this notion of the folding mechanism, a statistical analysis of all pathways makes sense.

In their most recent work [13], these researchers have refined the method by which new structures are generated in the sampling phase of roadmap construction. This method is based on **rigidity analysis**. For each structure, each degree of freedom is determined to be independently flexible, dependently flexible, or rigid, using an algorithm called the **Pebble Game**[8]. Independently flexible degrees of freedom rotate with no deterministic effect on others. Dependently flexible degrees of freedom force other degrees of freedom to change in a set way. Rigid degrees of freedom are essentially locked in place unless the constraints change. In perturbing an existing structure to generate a new sample, degrees of freedom are perturbed with a strong bias towards perturbing independently flexible degrees of freedom and against perturbing rigid degrees of freedom. Because the new structures are generated by physically realistic motions, it is expected that they will generally be lower energy and more representative of real structures than if they were generated by completely random perturbation of the degrees of freedom.

In practice, rigidity sampling appears to allow construction of high-quality roadmaps with many fewer samples than were necessary without it. It thus improves the overall efficiency of calculating protein behavior using this roadmap method.

2.1.2 Stochastic Roadmap Simulations

Numerous variants of MD and MC have been developed in an effort to speed up the process or focus the simulations on particular motions of interest. One method, called the **Stochastic Roadmap Simulation (SRS)**[3][4][5][6] uses a PRM-like structure to approximate a large number of simultaneous MC simulations very rapidly, allowing the analysis of an ensemble of trajectories. This method followed very directly from the first roadmap studies of molecular properties by Singh and Latombe.

The SRS method proceeds as follows:

- N samples are made uniformly at random by selection of a random value for each dihedral angle.
- The k nearest neighbors for each sample are found.
- For each sample, a transition probability is calculated to each of its nearest neighbors, depending on their energy difference as follows:

$$P_{ij} = \begin{cases} \frac{1}{d_j} e^{\frac{-\Delta E_{ij}}{k_B T}} & \text{if } \frac{\varepsilon_j / d_j}{\varepsilon_i / d_i} < 1, \\ \frac{1}{d_i} & \text{otherwise,} \end{cases}$$

where ε_i and ε_j are Boltzmann factors (exponentially proportional to energy) for conformations i and j, k_B is the Boltzmann constant, T is the absolute temperature, d_i and d_j are the neighbor counts for nodes i and j, and ΔE_{ij} is the energy difference between structures i and j.

Figure 13: Transition probabilities for SRS.

The energy, E, in this method is based on the hydrophobic-polar (H-P) energy model, and includes two terms, one favoring hydrophobic interactions, and the other depending on the solvent-excluded volume. Note the difference between the transition probabilities calculated by this method and those calculated by the method presented in the previous section. These probabilities depend only on the

energies of the endpoints of an edge, whereas those of the other method depend on the energy along the path between the endpoints. The probabilities of the SRS method are faster to calculate, and, assuming that the system is at equilibrium, more likely to be consistent with the actual distribution of conformations.

- Each sample is given a self-transition probability as follows, so that the sum of outgoing edge probabilities for each node is 1:

$$P_{ii} = 1 - \sum_{i \neq j} P_{ij}$$

where P_{ii} is the self-transition probability for node i, and P_{ij} is the probability of transitioning to state j if the current state is i.

Figure 14: Self-transition probabilities ensure that the total transition probability is 1.

The transition probabilities are defined as they are to be consistent with a Boltzmann-like distribution of energies, and therefore with standard Monte Carlo simulation probabilities. The authors demonstrated that each continuous path in the roadmap may be interpreted as a Monte Carlo simulation, and that, if a very large number of samples and edges are made, the aggregate behavior of these Monte Carlo simulations can be analyzed to estimate properties of the protein such as **folding rates** and **transition states**. Essentially, SRS is a way to generate large amounts of Monte Carlo simulation data in a short time. The developers or this method have provided a proof that, for a sufficiently large SRS and a sufficiently long Monte Carlo simulation, the distribution of conformations is expected to be equal.

To study protein folding using SRS, we observe that some set of nodes in the roadmap represent conformations in or very close to the folded state (native structure). We will refer to this set of nodes as F. For every node in the roadmap, we can compute an expected number of state transitions (or Monte Carlo steps) to go from that state to a node in F, with the base case that any node in F is defined to be at distance 0 from F. Given a precomputed SRS, we can compute this statistic for each node as follows:

$$t_i = 1 + \sum_{v_j \in \mathscr{F}} P_{ij} + \sum_{v_j \notin \mathscr{F}} P_{ij} t_j, \forall v_i \notin \mathscr{F}$$

where t_i is the expected number of transitions to get to a node in $\mathscr{F}$ from node v_i. P_{ij} is the probability of transitioning from node to node v_j if already at node v_i.

Figure 15: The expected number of transitions to reach a node in the folded state starting from node i.

This implies a system of linear equations on the variables ti. This system can be solved by an iterative numerical method such as Jacobi iteration. The solution is an estimate for each node of the average number of Monte Carlo steps necessary to achieve a folded conformation.

We can also define a set of nodes representing conformations close to the stable denatured state of the protein as the unfolded state, U. Given both of the sets U and F, we can define a quantity called the transmission coefficient, τ, for each node. The transmission coefficient expresses the probability that a structure at a particular node will proceed to a state in F before it reaches a state in U–in other words, it is the probability that a given structure will fold before it unfolds. This is often called the folding probability, or Pfold, in more recent research. The quantity, τ, is calculated for each node using the following relation:

$$\tau_i = \sum_{v_j \in \mathscr{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathscr{U}} P_{ij} \cdot 0 + \sum_{v_j \notin (\mathscr{F} \cup \mathscr{U})} P_{ij} \cdot \tau_j.$$

where τ_i is P_{fold}, the probability that a random walk on the SRS starting from node i will reach a node in the folded state, $\mathscr{F}$, before it reaches a node in the unfolded state, $\mathscr{U}$. P_{ij} is the probability of transitioning to state j if the current state is i.

Figure 16: The folding probability for a node i. This is the probability than a simulation starting at node i reaches a folded state before reaching an unfolded state.

As before, this relation implies a system of linear equations, this time on τi, the τ-value of each node. Again, it can be solved iteratively, and the result is a Pfold (τ) value for each node. Pfold is an interesting statistic in studying the mechanism of protein folding because structures with a true (as opposed to simulation-derived) Pfold of 0.5 have equal probability of going to the folded or unfolded states, and therefore each one is the highest energy structure on some folding pathway. These are the structures that constitute the **transition state ensemble (TSE)** of the protein, and study of these structures may provide insight into the mechanism by which the protein folds.

2.1.3 Markovian State Models

Markovian State Models (MSM)[10][11] are roadmaps constructed by running many molecular simulations (Monte Carlo and molecular dynamics) and merging the trajectories. The method starts with a simulation that runs from the folded to unfolded state. It then picks a structure at random (call it c) from this trajectory and run a new simulation. If this new simulation reaches the unfolded state, then the next trajectory from which we will pick a structure will consist of the old trajectory from the folded state to c, and the new trajectory from c to the unfolded state. If the new trajectory reaches the folded state, we do the opposite. If neither happens in a reasonable time, we reject the new trajectory and start over. This is repeated a set number of times, and each time a trajectory is accepted, all states from the new part of the trajectory are added to the growing roadmap as nodes, and each transition from the trajectory is added as an edge. When it is added, each edge is labeled with a transition probability of 1 and a transition time equal to the timestep of the simulation.

The goal of this method is to move roadmap methods closer to MD and MC simulations, and in particular to incorporate a notion of time, which follows from the use of simulation techniques in the sampling of new conformations.

Once all of the simulations have been run, nodes that are within some cutoff distance of each other by some similarity metric must be merged. To merge two nodes, one node is removed, and its edges are transferred to the other node. If this results in two edges between the same pair of nodes, the transition probabilities and times are defined as follows:

$$P_{ij}^{\text{new}} = P_{ij}^{1} + P_{ij}^{2}, \qquad t_{ij}^{\text{new}} = \frac{P_{ij}^{1} t_{ij}^{1} + P_{ij}^{2} t_{ij}^{2}}{P_{ij}^{1} + P_{ij}^{2}}.$$

where P_{ij} is the probability of transitioning to state j if in state i and t_{ij} is the expected time to to transition from state i to state j.

Figure 17: Expressions for new transition probabilities and transition times when merging nodes in constructing a MSM.

Once all merges are complete, the transition probabilities for each edge are normalized to the range [0,1] such that the sum of all outgoing edge probabilities from a node is 1. Given the roadmap, Pfold values and folding times can be calculated using the edge probabilities and step times. The approach is the same as with SRS: A system of equations is set up, but instead of Pfold, the value of interest is the expected time for a simulation starting at node i to reach a folded state, called the **mean first passage time (MFPT)**. The system of equations is solved using standard numerical methods, as with SRS. On tests of a small protein, called tryptophan zipper beta hairpin, or TZ2, the predicted folding rates agreed well with experiment.

An important fact of all roadmap methods that attempt to extrapolate properties of the entire protein folding landscape is that there is inherent sampling error. The energy landscape of a protein is a continuous function, which roadmap methods attempt to approximate through discrete sampling. The researchers who developed the MSM method also developed a method to estimate the error of the folding rates estimated based on MSMs [11]. While a complete description is beyond the scope of this module, the details are available in the 2005 paper by Singhal and Pande linked in the Recommended Reading section below. The error analysis allows them to generate a probability distribution for the folding times for each node in the MSM. Useful in its own right because it gives us an idea of how confident we can be in folding times generated by a given MSM, this analysis is especially useful for focusing sampling during the generation of an MSM.

The variance of the distribution of the folding time for each node provides an estimate of the error. If at each stage of simulation instead of choosing a node at random to start the next simulation, we select the node with the greatest contribution to our estimate of the error in folding time, we effectively focus our efforts where they will decrease the error most. In this way, MSMs with less overall error may be generated with using simulations.

2.2 Protein-Ligand Docking Pathways and Kinetics

So far, we have looked at applications of roadmap methods that deal with the single-body problem of protein folding. The first use of roadmaps in molecular modeling, however, was to study the two-body system of protein-ligand docking. The docking problem itself is, given a small molecule and a protein, to predict whether they will bind to form a complex, and if so, what will be the geometry and stability (binding affinity) of this complex. This problem is path-independent, and so does not lend itself to motion planning approaches. Roadmaps can be used, however, to study the question of how a ligand reaches or exits the binding pocket of a protein, what the energy profile of this process looks like, and the rate at which the ligand binds and dissociates.

Typically, in modeling protein-ligand docking with a roadmap, the protein is held rigid and induces a force field in which the ligand is free to rotate, translate, and change conformation. The first work in this area, by Latombe, Singh, and Brutlag [9], led a few years later to the SRS framework. An SRS for ligand docking pathways can be constructed be starting with the ligand in the bound state, and generating samples for its conformation, location, and orientation in, around, and outside the binding pocket of the protein. These paths can then be studied individually to examine features of the binding process, or as an aggregate to get properties such as binding affinity or escape time, which is represented in an SRS by the weight of

paths away from the bound state.

To validate their method for studying docking, the developers of SRS showed [4] that the escape times (in Monte Carlo steps) calculated for ligands leaving proteins with particular mutations in their binding sites were consistent with the expected effect of the mutations: Mutations expected to increase the binding affinity led to longer escape times, and mutations expected to decrease the binding affinity led to shorter escape times. They also showed that SRS could be used to distinguish between the binding site of the protein and other pockets on its surface. Ligands had significantly greater estimated escape times from the true binding site than from spurious ones.

Cortes et al. [7] developed a tree-based sampling method for studying protein-ligand docking pathways. The algorithm is based on the dynamic-domain RRT planner (see Robotic Path Planning and Protein Modeling for an introduction to RRTs and other tree-based motion planners), in which, when sampling a random point toward which to expand, the location of that point is restricted to be within some distance of the existing tree, rather than anywhere in the whole space. The sampling method is based on the geometry of the system being studied: The major factors contributing to the energy of a conformation are reduced to geometric criteria. Hydrogen bonds and hydrophobic interactions are modeled by distance constraints. Steric clash is handled by treating atoms as hard spheres and performing collision checks using a fast collision checker called BioCD, developed by the same research group. Only structures satisfying all geometric constraints are subjected to an energy minimization procedure. The geometric constraints help ensure that structures to be added to the tree are already fairly low-energy, ensuring that the minimization can be done quickly, and that time is not wasted minimizing unrealistic structures.

This work was applied to studying the **enantioselectivity** of various proteins. **Enantiomers** are molecules that are non-superimposable mirror images of each other. Although they contain the same atom types and connectivity, enantiomers of a given chemical cannot be interconverted without breaking and reforming bonds. Molecules may contain multiple sites where this kind of asymmetry exists, in which case the molecule may exist as a whole family of **diasteroemers**. Most biological molecules have at least one asymmetric center, and are therefore said to be **chiral**, and in most cases, only one diastereomer or enantiomer exists in appreciable quantity. The chemistry of a pair of enantiomers is identical **except** when they are interacting with other chiral molecules, in which case it is important that the correct diastereomer is present for the desired interaction.

Enantioselectivity is the ability of a protein to distinguish between the two enantiomers of a molecule. Since proteins are chiral, they exhibit enantioselectivity for enantiomeric ligands. In the tree-based method of Cortes et al, the amount of time it takes their planner to find an unbound state for a ligand turns out to be correlated with the difficulty of maneuvering the ligand into and out of the binding pocket. Thus, computation times for finding a path out of the binding pocket are much less for the preferred enantiomer of the ligand than for the other enatiomer, often by a factor of 10 or more.

Recommended Reading

- A PRM-Based Approach
  - Amato, N. M. and G. Song. PDF[2] . Using motion planning to study protein folding pathways. Journal of Computational Biology 9:149-168, 2002.
  - Amato, N. M., K. A. Dill, and G. Song. PDF[3] . Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. Journal of Computational Biology 10:239-255, 2003.
  - Thomas, Shawna, Guang Song and Nancy M. Amato. PDF[4] . Protein Folding by Motion Planning. Physical Biology 2:S148-S155, 2005.
  - Thomas, Shawna L., Xinyu Tang, Lydia Tapia, and Nancy M. Amato. PDF[5] . Simulating Protein Motions with Rigidity Analysis. Proceedings of the 2006 ACM International Conference on

[2] http://doi.acm.org/10.1145/369133.369239
[3] http://www.liebertonline.com/doi/abs/10.1089%2F10665270360688002
[4] http://cnx.org/content/m11449/latest/ http://www.iop.org/EJ/article/1478-3975/2/4/S09/ph5_4_s09.pdf
[5] http://www.springerlink.com/link.asp?id=w06g6wg14p1j4w2x

Research in Computational Molecular Biology (RECOMB), pp. 394-409.

- Strochastic Roadmap Simulations
  - Apaydin, M.S., A. P. Singh, D. L. Brutlag and J.-C. Latombe. PDF[6] . Capturing Molecular Energy Landscapes with Probabilistic Conformational Roadmaps. Proceedings of the 2001 IEEE International Conference on Robotics and Automation, pp. 932-939.
  - Apaydin, M. S., C.E. Guestrin, C. Varma, D.L. Brutlag, and J.-C. Latombe. PDF[7] . Stochastic roadmap simulation for the study of ligand-protein interactions. Bioinformatics, 18(s2):18-26, 2002.
  - Apaydin, M. S., D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe and C. Varma. PDF[8] . Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. Journal of Computational Biology 10:257-281, 2003.
  - Chiang, Tsung-Han, Mehmet Serkan Apaydin, Douglas L. Brutlag, David Hsu and Jean-Claude Latombe. PDF[9] . Predicting Experimental Quantities in Protein Folding Kinetics using Stochastic Roadmap Simulation. Proceedings of the 2006 ACM International Conference on Research in Computational Molecular Biology (RECOMB), pp. 410-424.
- Markovian State Models
  - Singhal, N., C. D. Snow and V. S. Pande. HTML[10] . Using path sampling to build better Markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. Journal of Chemical Physics 121:415-425, 2004.
  - Singhal, Nina and Vijay S. Pande. HTML[11] . Error analysis and efficient sampling in Markovian state models for molecular dynamics. Journal of Chemical Physics 123:204909, 2005.
- Docking Pathways and Kinetics
  - Cortes, J, T. Simeon, V. Ruiz de Angulo, D. Guieysse, M. Remauld-Simeon and V. Tran. PDF[12] . A Path Planning Approach for Computing Large-Amplitude Motions of Flexible Molecules. Bioinformatics 21(s1): i116-i125, 2005.

References

[1] K. A. Dill Amato, N. M. and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. *Journal of Computational Biology*, 10:239–255, 2003.

[2] N. M. Amato and G. Song. Using motion planning to study protein folding pathways. *Journal of Computational Biology*, 9:149–168, 2002.

[3] A. P. Singh D. L. Brutlag Apaydin, M.S. and J.-C. Latombe. Capturing molecular energy landscapes with probabilistic conformational roadmaps. *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 932–939, 2001.

[4] C.E. Guestrin C. Varma D.L. Brutlag Apaydin, M. S. and J.-C. Latombe. Stochastic roadmap simulation for the study of ligand-protein interactions. *Bioinformatics*, 18(s2):18–26, 2002.

[6] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=932670
[7] http://bioinformatics.oxfordjournals.org/cgi/reprint/18/suppl_2/S18
[8] http://portal.acm.org/citation.cfm?id=565196.565199
[9] http://cnx.org/content/m11449/latest/ http://www.springerlink.com/link.asp?id=mw24x00572l733u6
[10] http://scitation.aip.org/journals/doc/JCPSA6-ft/vol_121/iss_1/415_1.html
[11] http://scitation.aip.org/journals/doc/JCPSA6-ft/vol_123/iss_20/204909_1.html
[12] http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/suppl_1/i116

[5] D. L. Brutlag C. Guestrin D. Hsu J.-C. Latombe Apaydin, M. S. and C. Varma. Stochastic roadmap simulation: an efficient representation and algorithm for analyzing molecular motion. *Journal of Computational Biology*, 10:257–281, 2003.

[6] Mehmet Serkan Apaydin Douglas L. Brutlag David Hsu Chiang, Tsung-Han and Jean-Claude Latombe. Predicting experimental quantities in protein folding kinetics using stochastic roadmap simulation. *Proceedings of the 2006 ACM International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 410–424, 2006.

[7] T. Simeon V. Ruiz de Angulo D. Guieysse M. Remauld-Simeon Cortes, J and V. Tran. A path planning approach for computing large-amplitude motions of flexible molecules. *Bioinformatics*, 21:i116–i125, 2005.

[8] D.J. Jacobs and M.F. Thorpe. Generic rigidity percolation: The pebble game. *Physical Review Letters*, page 40518211;4054, 1995.

[9] Latombe J.C. Singh, A.P. and D.L. Brutlag. A motion planning approach to flexible ligand binding. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.

[10] C. D. Snow Singhal, N. and V. S. Pande. Using path sampling to build better markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *Journal of Chemical Physics*, 121:415–425, 2004.

[11] Nina Singhal and Vijay S. Pande. Error analysis and efficient sampling in markovian state models for molecular dynamics. *Journal of Chemical Physics*, 123:204909, 2005.

[12] Guang Song Thomas, Shawna and Nancy M. Amato. Protein folding by motion planning. *Physical Biology*, 2:S148–S155, 2005.

[13] Xinyu Tang Lydia Tapia Thomas, Shawna L. and Nancy M. Amato. Simulating protein motions with rigidity analysis. *Proceedings of the 2006 ACM International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 394–409, 2006.