CS5238 Combinatorial Methods in Bioinformatics 2004/2005 Semester 1

Lecture 12: Energy computation and motion analysis for proteins (II) – Oct 29, 2004

Lecturer: Prof Jean-Claude Latombe *Scribe: Chia Hoo Hon, Pramila Nuwantha Ariyaratne, Wang Lu*

12.1 Introduction

Molecular motion occurs in all biochemical processes. The study of molecular motion during the process of proteins binding may facilitate the study of diseases such as mad cow disease, which is caused by misfolding. In the pharmaceutical industry, it may help create more effective drugs by studying how the drug molecules are able to bind to proteins.

12.1.1 Drawbacks of existing computational techniques

One may study molecular motion through various computational techniques. Two common techniques are Monte Carlo and Molecular Dynamics simulation. However, they have two major drawbacks: each simulation run produces a single pathway, while in reality molecules tend to move along many different pathways as illustrated in Figure 12.1.

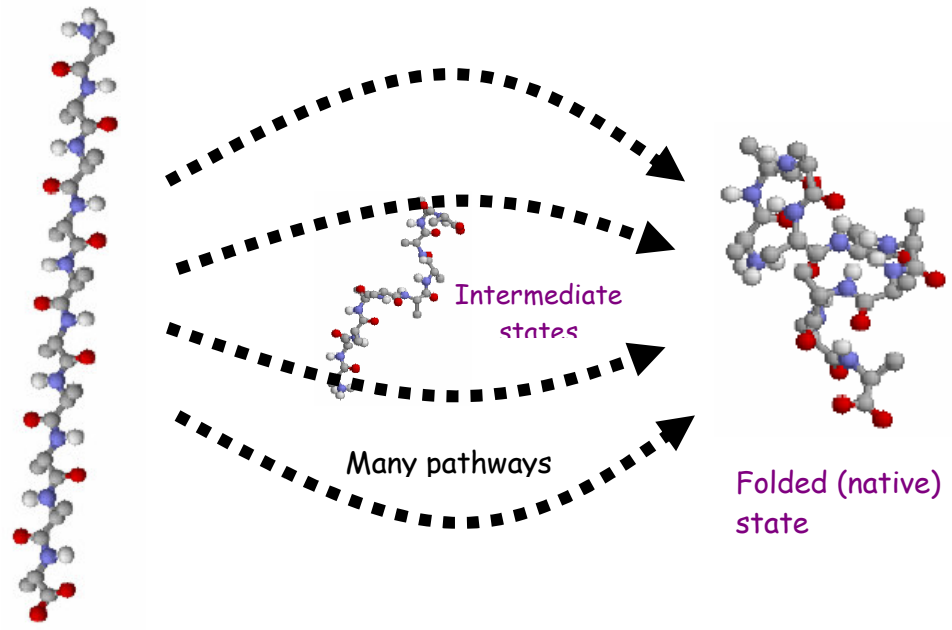

Figure 12.1: Protein Folding occurs along many distinct pathways

In the study of molecular motion, we are interested in the ensemble properties of the molecules' movement along different pathways. Examples of ensemble properties are

probability of folding, P_{fold}, order of formation of secondary structure elements, average time to escape, folding rate and key intermediates.

The other major drawback is that they tend to waste a lot of time in the local minima.

A new approach to overcome the two major drawbacks is to make use of roadmap-based representation. Roadmaps are capable of compacting representation of many motion pathways. They have coarse resolutions which are relative to Monte Carlo and Molecular Dynamics simulations. We will be looking into five of the algorithms in the following sections.

12.2 Motion Planning Approach to Flexible Ligand Binding

Amit P. Singh, Jean-Claude Latombe and Douglas L. Brutlag proposed in 1999 to use the traditional robot motion planning with probabilistic roadmaps to study the ligand protein binding.

A flexible ligand can be naturally modeled as an "articulated robot" with a free base as illustrated in Figure 12.2.

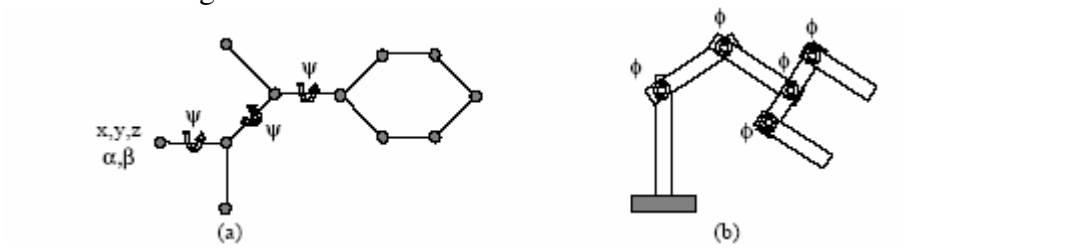

Figure 12.2: (a) A ligand with 8 degrees of freedom (3 coordinates (x, y, z) and 2 angles (α,β) for the root atom plus one torsional angle (ψ) for each non-terminal atom (b) A 2-dimensional fixed base articulated robot with 5 degrees of freedom (one rotational angle (Φ) for each joint

The approach is to utilize robot motion planning to determine potential paths that a ligand may naturally take based on the energy distribution of its workspace. This examines the possible motions that ligand is induced by the energy landscape of its immediate environment. With the use of probabilistic roadmap planners (PRM), it is able to represent energetic constraints in conformational space. PRM allows us to estimate the naturally induced motion of the ligand.

12.2.1 Roadmap Construction

The nodes are generated by sampling conformations of the ligand uniformly at random in the parameter space around the protein. The sampling process is bias towards to regions of low energy. The energy E is computed based on the following:

$E = E_{interaction} + E_{internal}$

$E_{interaction}$ = electrostatic + van der Waals potential

$E_{internal} = \Sigma_{\text{non-bonded pairs of atoms}}$ electrostatic + van der Waals

The randomly generated ligand configuration is accepted as a node with the following probability:

$$P(\text{accepted}) = \begin{cases} 0 & \text{if } E_{config} > E_{max} \\ \dfrac{E_{max} - E_{config}}{E_{max} - E_{min}} & \text{if } E_{min} \le E_{config} \le E_{max} \\ 1 & \text{if } E_{config} \le E_{min} \end{cases}$$

This method of probabilistic collision checking will result in a denser sampling of low-energy regions of conformational space.

The following algorithm is used to construct the roadmap from the set of S randomly generated nodes:

For each node i (0 <= i < S)

1) Sort all the remaining nodes (i.e. with index > i) based on their distance from i
2) While the number of edges at node i < N
   Use the local path planner to connect i to its first un-tested nearest neigbour (i.e. a node to which an edge has not yet been attempted with the local path planner)

For each of the accepted path, the local planner will compute a weight representing the energetic favorability of the path. The weight reflects the difficulty of traversing the path which means that the higher value for the paths, it indicates the paths crossing over large energy barriers.

The energy of each discretized configuration along the path is used to determine the overall probability of traversing the path in a particular direction. Figure 12.3 illustrates a simple energy contour for a straight line path between two nodes in a 1-dimensional configuration space.

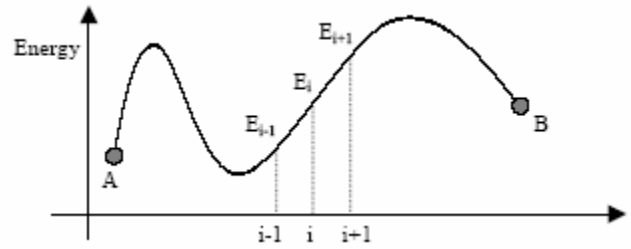

Figure 12.3

A line segment between nodes A and B is discretized into consecutive configurations that are less than ε units apart.

For any successive discretized configurations along the straight line path (i-1, i, i+1 with energies E_{i-1}, E_i, and E_{i+1}), he following equation is used to determine the probability of moving from configuration i to i+1:

$$P(i \text{ to } i+1) = \frac{e^{-(E_{i+1} - E_i)/kT}}{e^{-(E_{i+1} - E_i)/kT} + e^{-(E_{i-1} - E_i)/kT}}$$

From the above equation, the total weight of the path between A and B is as follows:

$$\text{Weight of local path} = \sum_i -\log[P(i \text{ to } i+1)]$$

The path weight is not the same in both the directions, though both the weights can be computed simultaneously.

12.2.2 Querying Roadmap

The sum of the local path weights between any one node and all the other remaining nodes of the roadmap reflect the kinetic and dynamic properties of the motion between these nodes.

For any randomly generated initial and goal configurations, the minimum weight path between them can be computed. Firstly, find the nodes in the roadmap that can be easily reached from these two arbitrary configurations. A graph search is then performed to find the minimum weight path between the two nodes in the roadmap.

As two weights are recorded for each edge representing the two different directions of motion, the minimum weight path can be computed in either of the direction. The graph search algorithm used is Dijkstra's single source shortest-path algorithm. The time complexity of this algorithm is $O(n^2)$ where n is the number of nodes.

The algorithm will terminate on either of the two conditions; the end node is reached or when all nodes are discovered. The latter condition allows the distribution of all paths entering or leaving a given node to be computed. With the computation of minimum weight for all paths, estimation can be derived of the average difficulty of all paths entering or leaving a given node. The two values can be correlated with the kinetic rates of binding (K_{on}) and dissociation (K_{off}) for the node.

12.2.3 Predicting binding sites

In order to predict the potential binding site on the receptor, the node generation technique is modified by over-sampling regions of low regions of low energy near the protein surface. The modified technique is as follows:

- Sorts the initial set of randomly generated nodes in order of increasing energy
- For each P lowest energy nodes, Q extra nodes are created around it by sampling a region of configuration space close to this initial node
- A new minimum energy node is selected among the Q extra samples and the process is iterated R times
- Initial set of P low-energy nodes are selected so that their centres of mass are at least 5 Å apart.

The result of iteration will result in P distinct regions of conformational space that are heavily over-sampled. The number of extra samples generated in each of the P regions is Q*R. This approach will report each of these P regions and the lowest energy nodes contained within them as potential active sites.

12.3 Motion Planning to Map Protein Folding Landscapes

This algorithm, proposed by N.M. Amato, K.A. Dill and G. Song in 2003, models Protein folding as a classical motion planning and attempts to solve it by using Probabilistic Road Maps. The implementation of PRM here differs from the classical implementation in fact that each sampled vertex is accepted according to 'energy computations' rather than collision detection.

The idea behind PRM is quite simple. Nodes which correspond to a specific state in configuration space are generated randomly. Then edges between these nodes are built using a local search method. Then optimal path between start node (state) and goal node (state) is found using standard graph traversing algorithms.

12.3.1 Protein Modeling

Protein structures are modelled similar to a rigid robot. The degrees of freedom are limited to phi-psi angles in the backbone. Bond length and bond angles are assumed to be fixed and side chains are modelled as sphere and bear no degrees of freedom.

12.3.2 Node Generation

As configuration space is multidimensional, uniform sampling will prove to be virtually impossible. Therefore the algorithm uses a different sampling strategy to generate nodes more effectively.

A native contact is a pair of C_{alpha} atoms of hydrophobic residue that are with in 7 A of each other in native state. Contact number of a conformation is defined as number of native contacts it has. The algorithm categorizes each sampled conformation in to a 'bin' according to its contact number. The sampling is continued until all bins have a minimum number (10) of conformations.

Initial samples are obtained by adjusting the phi-psi angles of the native state slightly. Each generated sample q is accepted according to the propobility,

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

Where E_{min} = 50000KJ/mol and E_{max} = 89000KJ/mol. As can be seen my the equation, low energy conformations are favoured.

Each node sampled is placed in a 'bin' according to its contact number. The node generation is continued by randomly selecting $N_{frontier}$ nodes from lowest filled bin as seeds for next round. New nodes will be created by sampling from normal distribution with seed node as the origin and each of 3 ^{o},5 ^{o},10 ^{o},20 ^{o},40^o as the standard deviation. Each new node which passes acceptance test is placed in appropriate bin. The process continues until all bins are filled.

12.3.3 Edge Generation

For each node in the roadmap, the k (20) closest neighbours (Euclidian distance) are found, and connected using a simple local search. The intermediate conformations between two nodes are found by uniformly interpolating. If two nodes can be connected, the edge between those two is added to the road map, and the weight of the edge depends on the sequence of intermediate conformations. For each pair of consecutive conformations c_i and c_{i+1} the probability of moving from c_i to c_{i+1} is

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

Where $\Delta E_i = E(c_{i+1}) - E(c_i)$

Then the weight of two edges in the roadmap is calculated by summing the negative logarithm of probabilities within each pair of conformations in the sequence.

$$w(q_1, q_2) = \sum_{i=0}^{n-1} -log(P_i)$$

12.3.4 Finding Pathway

Path from denatured conformation to native structure can be found by merely applying Dijkstra's single source shortest path algorithm to find the path with lowest cumulative weightage. If there is no node in the roadmap that corresponds to initial conformation, it is added manually similar to other nodes.

12.3.5 Energy Computations

The algorithm is very much dependent on energy computation to for both sampling strategy and node weightage.

The potential energy of a conformation depends on two factors, main chain restraints and van der waals forces. The total potential energy of a conformation is expressed as,

$$U_{tot} = \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} + \sum_{atom\ pairs} (A/r_{ij}^{12} - B/r_{ij}^6),$$

where first potion of the equation deals with main chain interactions and second potion models van der waals forces. For experiments K_d was set to 100kJ/mol and distances $d_0 = d_c = 2A$.

Entropy and free energy is used to analyze paths obtained by road map, as well as estimate folding rates. Free energy calculation consists of 3 components, hydrogen bond interaction, entropy and the hydrophobic form.

Contribution from hydrogen bonds is calculated by F_{hb} = -0.86 kcal/mol x N_{hb}, where N_{hb} refers to number of hydrogen bonds present in current conformation.

Entropy loss due to hydrogen bond formation is calculated as $\Delta s = \sum_i^{N_{hb}} \log \mathrm{ECO}_i$ and total free energy change as $F_{entropy}$ = 6.0 cal/mol/K x (300K) x ΔS.

Hydrophobic effect is calculated by $F_{hydrophobic}$ = -2.19kcal/mol * N_{hydro}, where N_{hydro} is the number of C_{alpha} atoms in hydrophobic residue that are less than 7A apart.

Results

Running Time and Roadmap Statistics				
PDB	res	nodes	edge	time (h)
1GB1	56	5126 (5506)	70k	3.71
1BDD	60	5471 (9106)	104k	7.03
1SHG	62	5427 (5502)	59k	2.89
1COA	64	7975 (8407)	104k	6.87
1SRL	64	8755 (8822)	111k	4.95
1CSP	67	6735 (6852)	72k	4.67
1NYF	67	6219 (6332)	70k	3.42
1MJC	69	5990 (6142)	62k	4.30
2AIT	74	8246 (8477)	92k	7.11
1UBQ	76	8357 (10667)	119k	9.44
1PKS	79	7685 (10257)	95k	9.32
1PBA	81	8085 (10747)	114k	10.40
2ABD	86	7330 (12577)	149k	14.20
1BRN	110	6601 (10607)	108k	15.80

As it can be seen from the table road map building is time wise expensive. However once the road map is built, succeeding queries can be answered efficiently.

12.4 Stochastic Roadmaps

The idea, proposed by M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.C. Latombe and C. Varma in 2003, behind this algorithm is the treat the space as a markov chain where traversing from one node to another occurs according to a probability. Unlike monte-carlo simulation, Stochastic Roadmap simulation considers many parallel paths at a given time and greatly speeds up computation.

12.4.1 Roadmap Building

Road map building procedure in this algorithm is very much similar to that of any other algorithm. Nodes in configuration space are sampled uniformly at random. The weight of each edge depends on the probability that molecule will move from first state to the

second, which is considered proportional to energy difference between two conformations.

Probability that molecule will travel from state i to j is given by

$$P_{ij} = \begin{cases} \frac{1}{d_j} \exp(-\Delta E_{ij}/k_B T) & \text{if } \frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1; \\ \frac{1}{d_i} & \text{otherwise;} \end{cases}$$

..where ε_i and ε_j are Boltzmann factors of v_i and v_j and d_i and d_j are number of neighbours of v_i and v_j.

12.4.2 Markov Chain Modeling

Given that there is a probability associating movement from 1 stage to another, the road map can essentially be considered a markov chain, provided the probabilities of each edge is normalized such that their sum equals to 1. Therefore we can estimate expected number of transitions to move from a given state to folded state without explicitly running simulations.

$$t_i = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j \quad \text{for every } v_i \notin \mathcal{F}.$$

Where *F* is set of folded states.
The values of *t* can be calculated by solving system of equations by Gaussian elimination or any other method.

12.4.3 Computing P_{fold}

P_{fold} expresses how probable it is for a certain state to reach folded state as opposed to unfolded state. If value is greater 0.5 then from that state it is more likely reach a folded state than unfolded state.

P_{fold} value can be efficiently calculated using markov chain modeling.

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

T_i value would correspond to P_{fold} value of node v_i. *F* and *U* represent sets of folded and unfolded states respectively.

Values of P_{fold} obtained via SRS and Monte Carlo generally in agreement. However time wise SRS appear to be much more effective. It takes about 1.5 hours to generate a roadmap with 5000 nodes with P_{fold} values for all those nodes using SRS. Where as in the same hardware it requires 5-6 hours to estimate P_{fold} for one node using MC. Overall SRS shows a performance increase of at least four order magnitude compared to MC.

12.5 Using Path Sampling to Construct Roadmaps

In 2004, N. Singhal, C.D. Snow and V.S. Pande proposed transforming the simulation data gathered from transition path sampling algorithms into a probabilistic roadmap that includes transition time data. They suggested that their method would improve on the current roadmap techniques by sampling points using molecular dynamics, thereby greatly increasing the probability that the configurations that are included are kinetically relevant.

12.5.1 Method and Theory - Sampling of paths

This method is a modified version of the shooting algorithm that has been shown to efficiently generate a sample of uncorrelated transition paths leading from the initial region to the final region.

Step1: From previous data, high temperature unfolding simulations, direct MC or Langevin simulations, we must obtain some initial path between the initial and final regions. We can label the points along this path as $\{P_0, P_1,\ldots, P_n\}$, where n is the length of the path.

Step2: We generate new paths by picking a random point along the current path, P_i, and "shooting" a new path from it by starting a new simulation from this point. Points are recorded along this path every t_{int} and are labeled $\{nP_0, nP_1,\ldots, nP_m\}$.

```
i = 0;
set some simulation time cutoff;
while (i<n)
{
        if (P_i != the initial && P_i != final state)
                reject P_i
        else
        {
                define a new path as P_i;
                break;
        }
        i ++;
```

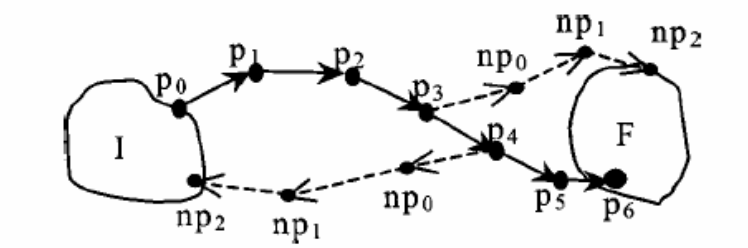

Figure 12.4: The shooting algorithm for sampling paths.

This sampling strategy will capture paths between the boundaries of the initial and final regions. If we are to calculate the MFPT between the initial and final regions, we must also simulate the time a particle can spend within the initial region.

As an example: Langevin dynamics equation of motion is F_{ext} - mγ dx/dy + R = 0, where R is a Gaussian random force. Sampling nodes from computed paths (path shooting) are showed in following Figure 12.5.

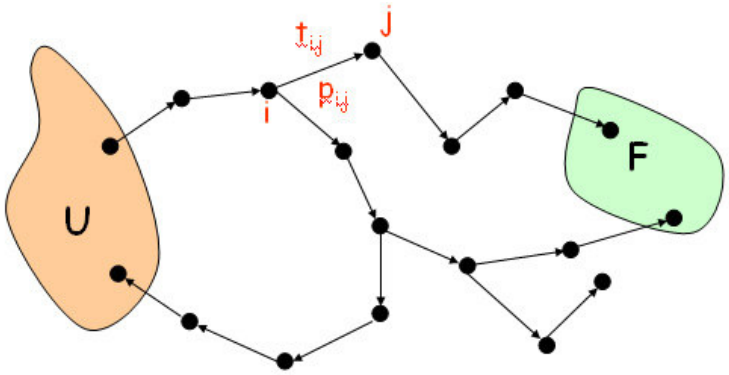

Figure 12.5: Clustering of nodes to guarantee that all nodes can reach the final state.

12.5.2 Method and theory --- Node Merging

Here we describe how to generate the MSM of conformational states, including the probability and time to traverse from node to node in the MSM. Each point in the paths accepted while sampling paths is represented by a node in the MSM, $node_i$, for some unique index *i*.

The MSM is designed to embody the possible pathways that the molecule may take while traversing the conformation space. Different paths generated by our simulation methods may pass through very similar conformations, but since the conformation space is continuous, these points will never be exactly the same. However, we wish to capture the fact that these paths reach essentially the same point. We can do this by clustering nearby points in conformation space according to some metric. We define some cutoff value that represents how close two points need to be in order for us to consider them to be the same point. Then, we combine points that are within this distance from one another according to some clustering algorithm. We may choose different cutoffs for the different regions of

conformation space, the initial region, the final region, and the transition region. To combine two points, we remove all the incoming and outgoing edges from one of the points and connect them to the other point. If there are now multiple edges between two nodes, we combine them into a single edge with the following values. (as following Figure 12.6 showed.)

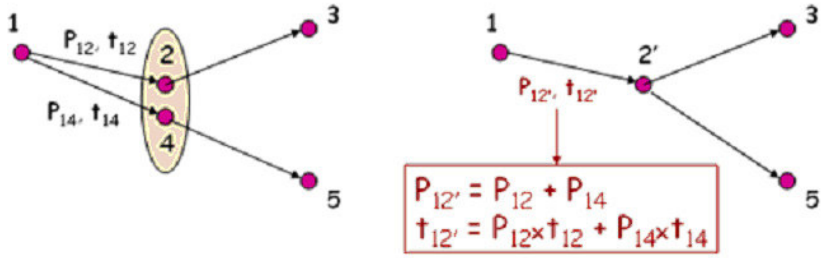

Figure 12.6: Node Merging

Notes: Approximately uniform distribution of nodes over the reachable subset of conformational space.

12.5.3 Method and theory --- Node Pruning

We propose two different methods for removing the nodes that were not marked. In the first, we simply delete those nodes, thus ensuring that all nodes in the MSM can reach a node in the final state. If there are not many such nodes, this should not bias the results very much. However, if there are many unmarked nodes, deleting these nodes could distort the results. Alternatively, nodes that cannot reach the final state are merged into the closest nodes until all nodes can reach the final state ~Fig. 3!. This nearest neighbor provides the best guess to the future dynamics of the unmarked node with respect to reaching the final state. In following Figure 12.7, the red nodes will be deleted and only those nodes from which a path exists to the folded state are retained in the final roadmap.

Figure 12.7: Node Pruning

12.5.4 Application --- Computation of MFPT

One can define the mean first passage time (MFPT) of any node as the average time taken to get from that node to any node in the final state. The MFPT can be defined conditionally based on the first transition made from the node.

$$\begin{aligned}&\mathrm{MFPT}(\mathrm{node}_i)\\&=\sum_{\mathrm{transition}(i,j)} P(\mathrm{transition}(i,j))\\&\quad\times\mathrm{MFPT}(\mathrm{node}_i|\mathrm{transition}(i,j)),\end{aligned}$$

In addition, we can define
MFPT(node$_i$) = 0, node$_i$ ∈ F
MFPT(node$_i$) = ∞, node$_i$ ∈ U

However, to validate the new methods for calculating kinetic properties, it is important to test the methods on systems in which the direct kinetics simulations can be performed. In this case, one can calculate the mean first passage time ~in terms of number of Monte Carlo steps for MC simulations and simulated time for Langevin simulations! directly from many independent simulations, even if these simulations are each shorter than the mean folding time.

If one assumes first order kinetics, the probability that a particle has reached the final state at some time t is given by

$$P_f(t) = 1 - e^{-rt}$$

where t is the time when a protein folds, r is the folding rate, $P_f(t)$ is the probability of having reached a final state by time t. By running many independent simulations shorter than 1/r, one can estimate the cumulative distribution $P_f(t)$, and hence fit the value for the rate, r. The MFPT is the average time when a particle will first reach the final state, given that it is in an initial state at t=0,

$$MFPT = \int_0^\infty \left(P_f(t)\right) t\,dt = 1/r$$

One could also find the MFPT by directly calculating the average time when each simulation first reached a final state. However, if some simulations are stopped before the final state is reached because of simulation time constraints, the MFPT calculated will be too low. By first fitting the rate to $P_f(t)$ data (which can be calculated accurately even if some simulations do not finish), one gets a much more accurate MFPT value. For simple systems (such as the two-dimensional energy landscape presented below), one can simply directly simulate kinetics on long timescales.

12.5.5 Application ---Computation Test

In addition to the model energy landscape, we applied our methods above to a small protein, the 12-residue tryptophan zipper beta hairpin, TZ2. TZ2 has previously been simulated on Folding@Home. The goal here is to use these trajectories from FoldingHome to build a MSM to further study the folding of TZ2. This should be a much more challenging test of our methods than the simple two-dimensional example above.

Using FoldingHome, TZ2 folding has been simulated using the OPLSaa all atom parameter set22 and the generalized Born/surface area implicit solvent model23 at a temperature of 296 K. Trajectories were started from an extended conformation and ranged in length from 10 to 450 nanoseconds.

To generate the MSM, we chose a tenth of this data set at random, resulting in 1750 independent trajectories. Of these trajectories, 14 reached the final folded state. Frames from the nonfolding trajectories were selected every 10 ns and frames from the folding trajectories were selected every 250 ps. This was done so that there would be more representative conformations in the transition and final states, while still allowing the number of nodes to stay manageable. As discussed in the MSM generation section, because the edges contain the time taken to traverse them, multiresolution data can be accommodated. This selection of data resulted in a total of approximately 22 400 nodes.

Over reasonable ranges of cutoffs for the initial (>2 Å) and transitional (1–2.5 Å) regions, we can estimate the MFPT as between 2–9 microseconds. This estimate agrees well with experimental results of 1.8±0.01 μs from fluorescence and 2.47±0.05 μs from IR[8] and with analysis of this simulation data fitting the rate directly of 8 μs for the full data set and 4.5 μs for the random tenth sample used in the MSM analysis.

12.6 Conformational Analysis of Protein Loops

This idea is proposed by J.Cortés, T.Siméon, M.Renaud-Siméon and V.Tran in 2004. The efficient filtering of unfeasible conformations would considerably benefit the exploration of the conformational space when searching for minimum energy structures or during molecular simulation. The most important conditions for filtering are the maintenance of molecular chain integrity and the avoidance of steric clashes (clash-free subset of the conformational space of a loop). These conditions can be seen as geometric constraints on a molecular model. In this article, we discuss how techniques issued from recent research in robotics can be applied to this filtering by building a tree-shaped roadmap.

The conformational analysis of a whole macromolecule is a difficult problem. From a methodological point of view, two stages are usually necessary: the first corresponds to the identification of rigid segments (i.e., secondary structural elements) capable of participating in the molecular framework; the second is devoted to the remaining segments, so-called loops, assumed to be much more flexible. However, available

techniques to predict low-energy conformations of long loops are limited and much less efficient because of the loop flexibility.

12.6.1 Problem Formulation

The problem is formulated from a robotic point of view. First, the geometric model of the molecule is described. The constraints that must be satisfied during the exploration of the conformational space are then defined – **Kinematics-Inspired model**. The kinematic model of a polypeptide segment is composed of a set of chains: the main-chain (ϕ-ψ angles on the backbone) and torsional angles χ_i on the side-chains, which are built upon it. The conformation of the segment is then specified by an array containing the conformation parameters of the backbone and of all the side-chains. Following Figure 12.8 shows the molecular chain model.

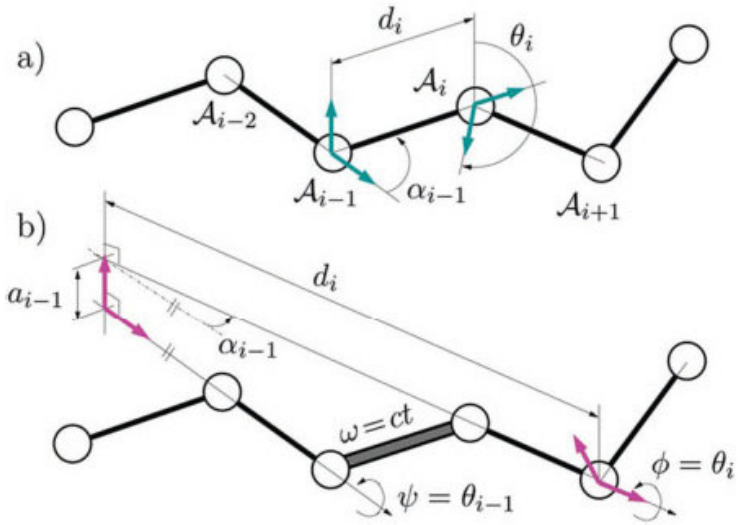

Figure 12.8: Molecular Chain model. Frames asscociated (a) with atoms and (b) defining the articulated mechanism.

12.6.2 Amylosucrase (AS)

Amylosucrase (AS) is a glucansucrase that catalyzes the synthesis of an amylose-like polymer from sucrose. This enzyme is classified in family 13 of glucoside-hydrolases (GH), which mainly contains starch-converting enzymes. Remarkably, this enzyme is the only polymerase acting on sucrose substrate reported in this family, all the other glucansucrases being gathered in GH family 70. Which structural features are involved in AS specificity is an important fundamental question.

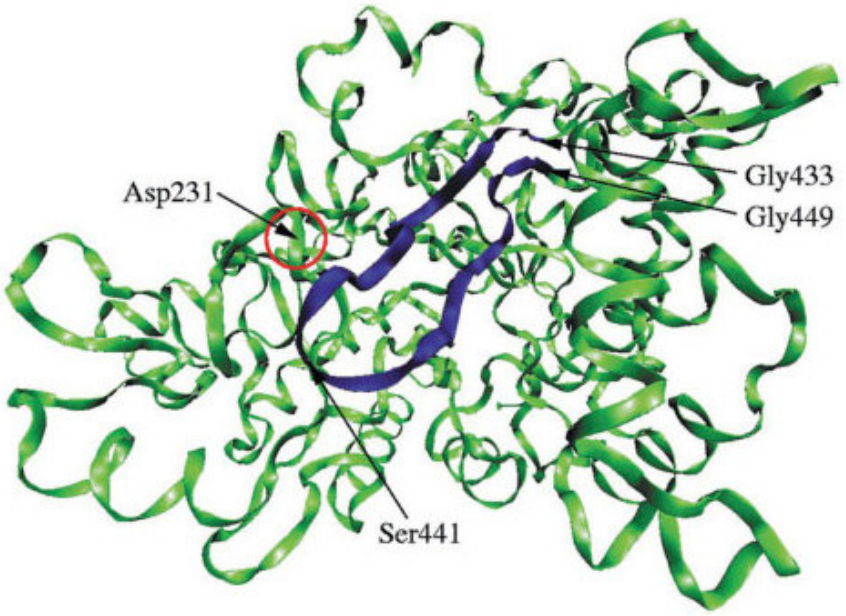

Figure 12.9: Structure of AS

The comparison of the various structures obtained suggests that motion of the 17-residue fragment of domain B' starting at residue Gly433 and ending at residue Gly449, consecutive to oligosaccharide binding, could facilitate sucrose translocation from SB2 to the active site. In the following part, this fragment will be called loop 7. This loop could play a pivotal role responsible for the structural change and the polymerase activity.

Figure 12.9 shows the crystallographic structure of AS and the location of the residues we mention in the following paragraphs. The model for our tests was created from the PDB file containing this structure (PDB ID: 1G5A), considering loop 7 as an articulated mechanism and the rest of the atoms as static elements. Atoms were modeled with 70% of their vdW radii. Images on the left in Figure 12.9 represent the articulated vdW model of the loop and a portion of its environment. Under our modeling assumptions, the results of the geometric exploration showed that only slight conformational variations of the loop are possible if the backbone integrity is maintained and steric clashes are avoided.

12.6.3 RLG (Random Loop Generator) Algorithm

This algorithm produces random configurations of articulated mechanisms containing closed chains. The configuration parameters of a closed kinematic chain are separated into two arrays: we call the independent variables of the closure equations the active variables q^a and the dependent variables the passive variables q^p. The RLG algorithm performs a particular random sampling for q^a that notably increases the probability of obtaining solutions for q^p.

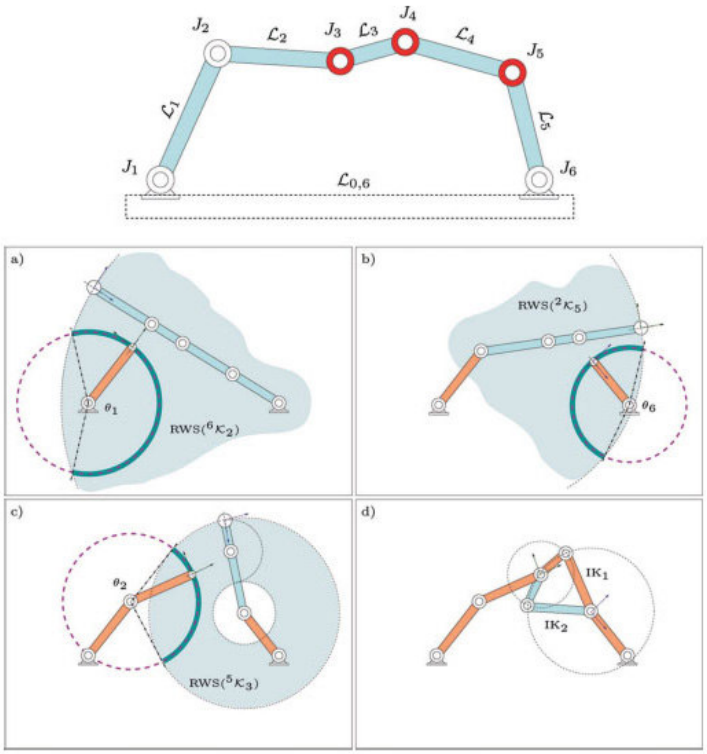

Figure 12.10: Steps of the RLG algorithm performed on a 6R planar linkage

We next explain the main elements of our approach and how it can be applied to polypeptide backbone segments. Explanations are illustrated on a simple mechanism, the 6R planar linkage in Figure 12.10. The L_1 are the rigid bodies and the J_i the revolute joints connecting them.

Algorithm 2: RANDOMBACKBONECONF.

---

input : the *backbone*
output : the conformation q_b
begin
  $q^a \leftarrow$ SAMPLE_q^a(*backbone*);
  if $q^p \leftarrow$ COMPUTE_q^p (*backbone*, q^a) **then**
    $q_b \leftarrow$ COMPOUNDCONF(*backbone*, q^a, q^p);
  else return *Failure;*
end

12.6.4 Roadmap Construction

In particular, algorithms based on the PRM (probabilistic roadmap) approach have mostly been developed. The general PRM principle is to construct a graph (roadmap) that captures the topology of the feasible subset of robot configurations, ζ_{feas}. The nodes of this graph are randomly sampled configurations satisfying intrinsic conditions in this

subset (e.g., collision avoidance). The edges are short feasible paths (local paths) linking "nearby" nodes.

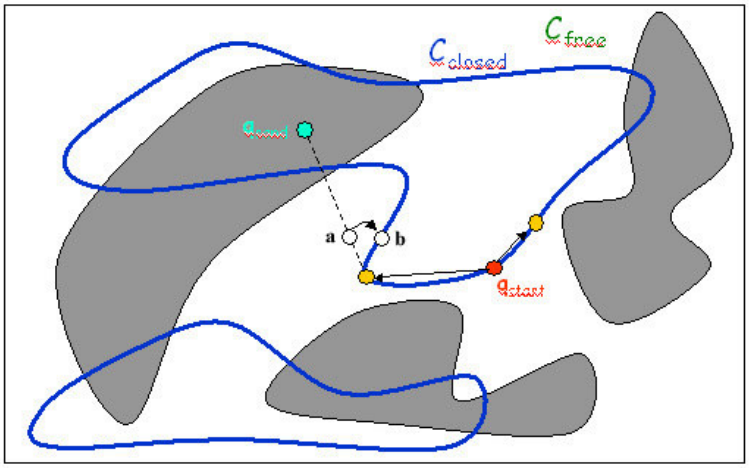

Figure 12.11

Figure 12.11 illustrates the exploration in a simple 2D example. The dardker regions in the Figure 12.11 correspond to conformations with steric clashes, C_{free} being the rest of the space. In general, conformations satisfying closure (in C_{closed}) are grouped into different disjoint continuous manifolds. The starting point q_{start} can be a randomly sampled feasible conformation (e.g., generated by the technique explained above) or a known conformation. For executing an expansion step of the RRT, a random conformation q_{rand} is first sampled in ζ_{feas}. q_{rand} need not satisfy either closure or clash avoidance constraints. In Figure 12.11, the chain between a and b can be decomposed by RLG in Figure 12.12.

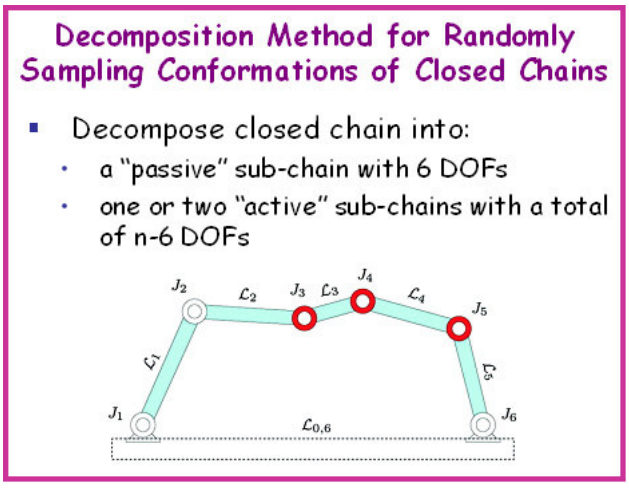

Figure 12.12

After that loop in RLG, we stops when one can't get closer to q_{rand} or a clash is detected. Then we get the updated exploration of ζ_{feas} as Figure 12.13.

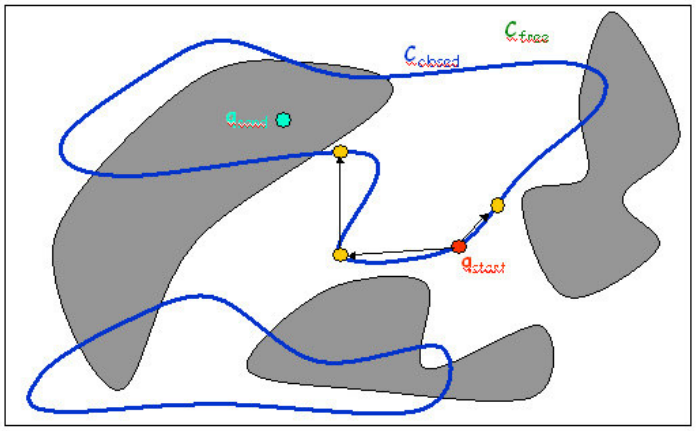

Figure 12.13

12.6.5 Computational Results of AS

Images on the left in Figure 12.14 represent the articulated vdW model of the loop and a portion of its environment. Under our modeling assumptions, the results of the geometric exploration showed that only slight conformational variations of the loop are possible if the backbone integrity is maintained and steric clashes are avoided. The image on the right in Figure 12.14(a) shows the skeleton of the articulated segment and a representation of one of the RRTs computed for this test. Nodes of the RRT are graphically represented by the positions explored by the C_α atom of Ser441, the middle residue of the loop

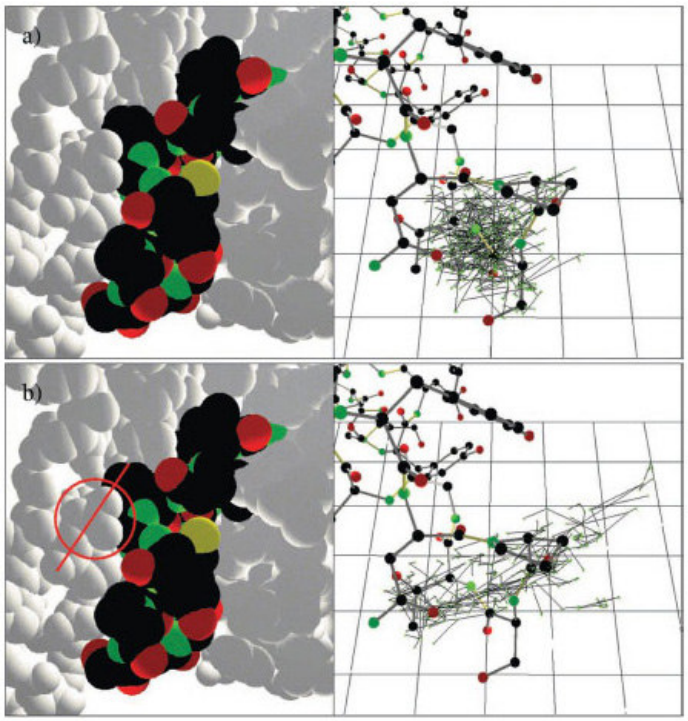

Figure 12.14: Exploration (a)with and (b) without the side-chain of Asp231

Figure 12.14(b) shows the representation of the RRT constructed in one of these tests. The images in Figure 12.10 correspond to four frames of the conformational change encoded in the RRT.

12.7 Conclusion

The using of probabilistic roadmaps provides a promising tool for exploring the conformational space and computing the ensemble properties of different molecular pathways.

As the exploration of usage of probabilistic roadmaps is still recent, further research can be done to improve on the sampling strategies to handle more complex molecular models such as protein to protein binding. Another aspect is to include time information in the roadmaps. Finally, to see the effectiveness of roadmaps, more thorough experimental validation is required to compare computed and measured quantitative properties.

References

[1] A.P. Singh, J.C. Latombe and D.L.Brutlag, "A Motion Planning Approach to Flexible Ligand Binding", *Proc. 7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), 252-261, 1991*

[2] N.M. Amato, K.A Dill and G.Song, "Using Motion Planning to Map Protein Folding Landscapes and Analyze Folding Kinetics of Known Native Structures", *J. Comp. Biology, 10(2):239-255, 2003*

[3] M.S.Apaydin, D.L.Brutlag, C.Guestrin, D.Hsu, J.C Latombe and C.Varma, "Stochastic Roadmap Simulation: An efficient Representation and Algorithm for Analyzing Molecular Motion", *J. Comp. Biology, 10(3-4):257-281, 2003*

[4] N.Singhal, C.D.Snow 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", *J. Chemical Physics, 121(1):415-425, 2004*

[5] J.Cortés, T.Siméon, M.Renaud-Siméon and V.Tran, "Geometric Algorithms for the Conformational Analysis of Long Protein Loops", *J. Comp. Chemistry, 25:956-967, 2004*