Predicting Protein Folding Paths

Probabilistic Roadmap Planning (PRM):

Aims

- Find good quality folding paths (into given native structure)
  - no structure prediction!
- Predict formation orders (of secondary structure)
Motion planning

- Probabilistic roadmap planning
  - Sampling of configuration space $Q$
  - Connect nearest configurations by (simple) *local planner*
  - Apply graph algorithms to “roadmap”: Find shortest path
Motion planning

- Probabilistic roadmap planning
  - Sampling of configuration space $Q$
  - Connect nearest configurations by (simple) \textit{local planner}
  - Apply graph algorithms to “roadmap”: Find shortest path
Motion planning

- Motion planning

Probabilistic roadmap planning
- Sampling of configuration space $Q$
- Connect nearest configurations by (simple) \textit{local planner}
- Apply graph algorithms to “roadmap”: Find shortest path
Motion planning

- Probabilistic roadmap planning
  - Sampling of configuration space $Q$
  - Connect nearest configurations by (simple) *local planner*
  - Apply graph algorithms to “roadmap”: Find shortest path
Motion planning

Probabilistic roadmap planning

- Sampling of configuration space $Q$
- Connect nearest configurations by (simple) local planner
- Apply graph algorithms to “roadmap”: Find shortest path
More on PRM for motion planning

tree-like robots (*articulated robots*)

configuration = vector of angles

configuration space

\[ Q = \{ q \mid q \in S^n \} \]

- \( S \) — set of angles
- \( n \) — number of angles = degrees of freedom (dof)
More on PRM for motion planning

- tree-like robots (articulated robots)

- configuration = vector of angles

- configuration space

\[ Q = \{ q \mid q \in S^n \} \]

- \( S \) — set of angles
- \( n \) — number of angles = degrees of freedom (dof)
Proteins are Robots (aren’t they?)

Obvious similarity ;-)
Proteins are Robots (aren’t they?)

Obvious similarity ;-) 

Our model

- Protein \( \rightarrow \) vector of phi and psi angles (treelike robot with 2n dof)
- possible models range from only backbone up to full atom
Proteins are Robots (aren’t they?)

Obvious similarity ;-)
Proteins are Robots (aren’t they?)

Obvious similarity ;-)
Differences to usual PRM

- no external obstacles, but
  - self-avoidingness
  - torsion angles
- quality of paths
  - low energy intermediate states
  - kinetically prefered paths
  - highly probable paths
Energy Function

- The method can use any potential.

Our coarse potential

- Each sidechain is represented by only one “atom” (zero degrees of freedom).

\[
U_{tot} = \sum \left\{ K_d \left[ ((d_i - d_0)^2 + d_c^2)^{1/2} - d_c \right] + E_{hp} \right. \]

- **First term** favors known secondary structure through main chain hydrogen bonds and disulphide bonds.

- **Second term** hydrophobic effect.

- Van der Waals interaction modeled by step function.

- All-atom potential: EEF1

[Lazaridis, Karplus. Proteins, 1999.]

PRM method for Proteins

Sampling  Connecting  Extracting
Sampling — Node Generation

Sampling

Connecting

Extracting
Node Generation

No uniform sampling
- configuration space too large
- \( \Rightarrow \) need biased sampling strategy

Gaussian sampling
- centered around native conformation
- with different STDs \( 5^\circ, 10^\circ, \ldots, 160^\circ \)
- ensure representants for different numbers of native contacts

Selection by energy

\[
P(\text{accept } q) = \begin{cases} 
1 & \text{if } E(q) < E_{\text{min}} \\
\frac{E_{\text{max}} - E(q)}{E_{\text{max}} - E_{\text{min}}} & \text{if } E_{\text{min}} \leq E(q) \leq E_{\text{max}} \\
0 & \text{if } E(q) > E_{\text{max}}
\end{cases}
\]
More on Node Generation

- Visualization of Sampling Strategy

- Distribution
  - Psi and Phi angles
  - RMSD vs. Energy
Node Connection

Sampling

Connecting

Extracting
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate N intermediary nodes by local planner

Assign weights to edges

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

Weight = \( \sum_{i=0}^{N} -\log(P_i) \)
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate N intermediary nodes by local planner
- assign weights to edges

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

Weight = \[\sum_{i=0}^{N} -\log(P_i)\]
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate \( N \) intermediary nodes by local planner

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

Weight = \( \sum_{i=0}^{N} -\log(P_i) \)
Connecting Nodes by Local Planner

- Connect configurations in close distance
- Generate $N$ intermediary nodes by local planner

Assign weights to edges

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

Weight:

$$\text{Weight} = \sum_{i=0}^{N} -\log(P_i)$$
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate $N$ intermediary nodes by local planner

### Assign Weights to Edges

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

$$\text{Weight} = \sum_{i=0}^{N} -\log(P_i)$$
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate N intermediary nodes by local planner

Assign weights to edges

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

Weight

\[ \text{Weight} = \sum_{i=0}^{N} -\log(P_i) \]
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate N intermediary nodes by local planner

assign weights to edges

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

\[ \text{Weight} = \sum_{i=0}^{N} -\log(P_i) \]
Connecting Nodes by Local Planner

- connect configurations in close distance
- generate N intermediary nodes by local planner

assign weights to edges

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

Weight = \( \sum_{i=0}^{N} -\log(P_i) \)
Extracting Paths

Sampling

Connecting

Extracting
Extracting Paths

- **Shortest Path**
  - extract one shortest path
  - from some starting conformation, one path at a time

- **Single Source Shortest Paths (SSSP)**
  - extract shortest paths from all starting conformation
  - compute paths simultaneously
  - generate tree of shortest paths (SSSP tree)
Big Picture

Sampling

Connecting

Extracting
## Studied Proteins

Overview of studied proteins, roadmap size, and construction times

<table>
<thead>
<tr>
<th>pdb</th>
<th>Description</th>
<th>Length</th>
<th>SS</th>
<th># Nodes</th>
<th>Time (h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1gb1</td>
<td>Protein G domain B1</td>
<td>56</td>
<td>$1\alpha + 4\beta$</td>
<td>8,000</td>
<td>6.400</td>
</tr>
<tr>
<td>2crt</td>
<td>Cardiotoxin III</td>
<td>60</td>
<td>$5\beta$</td>
<td>8,000</td>
<td>6.430</td>
</tr>
<tr>
<td>1bdd</td>
<td>Staphylococcus protein A</td>
<td>60</td>
<td>$3\alpha$</td>
<td>10,000</td>
<td>10.400</td>
</tr>
<tr>
<td>1shg</td>
<td>SH3 domain $\alpha$-spectrin</td>
<td>62</td>
<td>$5\beta$</td>
<td>10,000</td>
<td>8.344</td>
</tr>
<tr>
<td>2ptl</td>
<td>Protein L, B1 domain</td>
<td>62</td>
<td>$1\alpha + 4\beta$</td>
<td>4,000</td>
<td>3.104</td>
</tr>
<tr>
<td>1coa</td>
<td>CI2</td>
<td>64</td>
<td>$1\alpha + 4\beta$</td>
<td>10,000</td>
<td>9.984</td>
</tr>
<tr>
<td>1srl</td>
<td>SH3 domain src</td>
<td>64</td>
<td>$5\beta$</td>
<td>8,000</td>
<td>5.990</td>
</tr>
<tr>
<td>1nyf</td>
<td>SH3 domain fyn</td>
<td>67</td>
<td>$5\beta$</td>
<td>10,000</td>
<td>8.418</td>
</tr>
<tr>
<td>2ait</td>
<td>Tendamistat</td>
<td>74</td>
<td>$7\beta$</td>
<td>10,000</td>
<td>13.327</td>
</tr>
<tr>
<td>1ubq</td>
<td>Ubiquitin</td>
<td>76</td>
<td>$1\alpha + 5\beta$</td>
<td>8,000</td>
<td>10.381</td>
</tr>
<tr>
<td>1pks</td>
<td>SH3 domain PI3 kinase</td>
<td>79</td>
<td>$1\alpha + 5\beta$</td>
<td>10,000</td>
<td>14.446</td>
</tr>
<tr>
<td>1pba</td>
<td>Procarboxypeptidase A2</td>
<td>81</td>
<td>$3\alpha + 3\beta$</td>
<td>8,000</td>
<td>10.845</td>
</tr>
</tbody>
</table>
Formation orders

- formation order of secondary structure for verifying method
- formation orders can be determined experimentally [Li, Woodward. Protein Science, 1999.]
  - Pulse labeling
  - Out-exchange
- prediction of formation orders
  - single paths
  - averaging over multiple paths (SSSP-tree)
Timed Contact Maps

(I: $\alpha$)

(V: $\beta_{\text{turn}_1}$)

(II: $\beta_{1/2}$)

(III: $\beta_{3/4}$)

(IV: $\beta_{1/4}$)

(VI: $\beta_{2/4}$)

(V): $\beta_{\text{turn}_1}$

(II): $\beta_{1/2}$

(III): $\beta_{3/4}$

(IV): $\beta_{1/4}$

(VI): $\beta_{2/4}$
no (reported) contradictions between prediction and validation

different kind of information from experiment and prediction
The Proteins G and L

- Studied in more detail
- good test case
- structurally similar: $1\alpha + 4\beta$

- fold differently
  - Protein G: $\beta$-turn 2 forms first
  - Protein L: $\beta$-turn 1 forms first
## Comparison of Analysis Techniques

### $\beta$-Turn Formation

<table>
<thead>
<tr>
<th>Name</th>
<th>Contacts considered</th>
<th>Energy function</th>
<th>Secondary structure formation order</th>
<th>Analyze first $x%$ contacts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Protein G</td>
<td>All</td>
<td>Our</td>
<td>$\alpha$, turn 2, turn 1</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>turn 2, $\alpha$, turn 1</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha$, turn 1, turn 2</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All-atom</td>
<td>$\alpha$, turn 2, turn 1</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>turn 2, $\alpha$, turn 1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Hydrophobic</td>
<td>Our</td>
<td>$\alpha$, turn 2, turn 1</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha$, turn 1, turn 2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All-atom</td>
<td>$\alpha$, turn 2, turn 1</td>
<td>76</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha$, turn 1, turn 2</td>
<td>24</td>
</tr>
<tr>
<td>Protein L</td>
<td>All</td>
<td>Our</td>
<td>$\alpha$, turn 1, turn 2</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>turn 1, $\alpha$, turn 2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Hydrophobic</td>
<td>Our</td>
<td>$\alpha$, turn 1, turn 2</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\alpha$, turn 2, turn 1</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>All-atom</td>
<td>Our</td>
<td>$\alpha$, turn 1, turn 2</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>Hydrophobic</td>
<td></td>
<td>turn 1, $\alpha$, turn 2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All-atom</td>
<td>$\alpha$, turn 1, turn 2</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>turn 1, $\alpha$, turn 2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Hydrophobic</td>
<td></td>
<td>$\alpha$, turn 2, turn 1</td>
<td>31</td>
</tr>
</tbody>
</table>
Conclusion

- PRM can be applied to “realistic” protein models
- Introduced method makes verifiable prediction
- Coarse potential is sufficient
- Predictions in good accordance to experimental data