Robotics-inspired methods to sample conformations and transition paths of flexible biomolecules

Juan Cortés

Winter School *Algorithms in Structural Bioinformatics*

2019
Connections between robotics and molecular modeling

ECHORD++ DualArmWorker Experiment, LAAS-CNRS, Tecnalia, Airbus, Nov. 2017
Course Contents

- Loop sampling
- Path sampling on energy landscapes

**NOT COVERED**
- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design
Course Contents

• Loop sampling
• Path sampling on energy landscapes

NOT COVERED
• Conformational transitions in proteins
• Protein-ligand access/exit pathways
• Towards molecular motion design
Course Contents

- Loop sampling
- Path sampling on energy landscapes

**NOT COVERED**
- Conformational transitions in proteins
  [Al-Bluwi et al., *BMC Struct Biol*, 2013]
Course Contents

- Loop sampling
- Path sampling on energy landscapes

**NOT COVERED**
- Conformational transitions in proteins
- Protein-ligand access/exit pathways

[Cortés et al., *PCCP* 2010]
[Devaurs et al., *NAR* 2013]

Web server: moma.laas.fr
Course Contents

• Loop sampling
• Path sampling on energy landscapes

**NOT COVERED**
• Conformational transitions in proteins
• Protein-ligand access/exit pathways
• Towards molecular motion design
  
  [Molloy et al., *Int. J. Robotics Research*, 2018]
Course Contents

- Loop sampling
- Path sampling on energy landscapes

NOT COVERED
- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design
Internal coordinate models of proteins and RNA

Nucleotide

3 amino-acid fragment (*tripeptide*)

6 bond torsion (dihedral) angles
Modeling molecular chains as articulated mechanism

Denavit-Hartenberg parameters: \( \{a_i, \alpha_i, d_i, \theta_i\} \), only \( d_i \) or \( \theta_i \) is variable

Homogeneous matrix transformation:

\[
\begin{align*}
  i-1T_i &= \begin{pmatrix}
  \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\
  \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\
  \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\
  0 & 0 & 0 & 1
  \end{pmatrix} \\
  0T_n &= 0T_1^1 T_2 \ldots \ T_{n-1}^1 T_n
\end{align*}
\]
Modeling molecular chains as articulated mechanism

Denavit-Hartenberg parameters for a molecular chain

a) Consecutive bond torsions: link length \( a_{i-1} = 0 \)
b) Non-consecutive bond torsions: link length \( a_{i-1} \neq 0 \)
Modeling molecular chains as articulated mechanism

Inverse Kinematics Problem

Given the relative pose $T_{rel}$ of the end-effector with respect to the base-frame find the values of the joint variables $\theta_i$

$$T_{rel} = ^0T_1(\theta_1) \cdot ^1T_2(\theta_2) \cdot ^2T_3(\theta_3) \cdot ^3T_4(\theta_4) \cdot ^4T_5(\theta_5) \cdot ^5T_6(\theta_6)$$

→ Efficient closed-form solvers [Renaud, 2000]
Monte Carlo move classes

[Denarie et al., Molecules 2018]

Local (fixed-end) backbone perturbation methods

• One particle moves
  • Perturb one particle
  • IK for two tripeptides
  • Similar to ConRot [Dodd et al., 1993]

• Flexible fragment moves
  • Perturb \( n \) consecutive particles
  • IK for \( n+1 \) tripeptides
  • Similar to CCL [Canutescu et al., 2003]

• Rigid-body block moves (“hinge”)
  • Perturb \( n \) particles as a rigid-body
  • IK for two tripeptides
  • Similar to CRRUBAR [Betancourt, 2005]
Sampling long loops

Challenging problem

How to sample submanifolds?

Several approaches:

- **Optimization-based**: Coordinate Cyclic Descent (CCD) [Welman, 1993]
  - Application to proteins: [Canutescu et al, *Protein Sci.*, 2003]

- **Semi-analytical**: Random Loop Generator (RLG) [Cortés et al, 2002]
Random Loop Generator (RLG)

Main Chain (Backbone)

→ Independent sub-chain
  • RLG algorithm + CollCheck

→ Dependent sub-chain
  • General 6R IK + CollCheck
    [Renaud, 2000]

Side Chains

• Random Sampling + CollCheck

Sampling long loops

[Cortés et al., J Comput Chem, 2004]
Machine-learning approach to protein loop sampling

Idea: Sample from a database of tripeptide configurations (instead of uniform random sampling)

Data: Extracted from ~ 80,000 protein domain structures

Database ~ 6 million tripeptide configurations from coils
Machine-learning approach to protein loop sampling

[Barozet et al., in preparation]

Incremental sampling method

At each iteration:

1. Choose random position $s$

2. Iteratively build a feasible loop configuration by sampling tripeptides $T_1 \ldots T_{s-1}, T_{s+1} \ldots T_k$

3. Solve for $T_s$ via semi-analytical IK

$s = 5$

Anchors
Machine-learning approach to protein loop sampling

Tripeptide selection with Reinforcement Learning

Idea: Incorporate prior knowledge from previous attempts (~ online RL)

How: Feature-based organization in tree data structure (e.g. octree)

- Cell score = rate of success
- Iterative construction (dependency)
  - Scoring function captures downstream results
    
    \[ score^k_c = score^k_c \times \prod_{m=k+1}^{K} score^m \]

[Barozet et al., in preparation]
Early results:
Comparison of learning method against naïve tripeptide selection

- 4 seconds per computation
- Learning outperforms naïve
- Gain improves with time

[Barozet et al., in preparation]
Exhaustive sampling of small cyclic peptides

Joint work with J. Chomilier and D. Stratmann (UPMC, Paris)

Cyclic-pentapeptide

Conformational space (considering only $\Phi$, $\Psi$) → 4-dimension manifold in a 10-dimension space

Can be sampled exhaustively!
Exhaustive sampling of small cyclic peptides

[Jusot et al., JCIM 2018]

Sampling of $\Phi_1, \Psi_1, \Phi_5, \Psi_5 (+ \text{ all } \omega)$
Exhaustive sampling of small cyclic peptides

Sampling of $\Phi_1$, $\Psi_1$, $\Phi_5$, $\Psi_5$ ($+$ all $\omega$)

Inverse kinematics:
- 0 solution
- 1 to 16 solutions

Check collisions

[Jusot et al., JCIM 2018]
Exhaustive sampling of small cyclic peptides

Sampling of $\Phi_1$, $\Psi_1$, $\Phi_5$, $\Psi_5$ (+ all $\omega$)

Inverse kinematics:
- 0 solution
- 1 to 16 solutions

Check collisions

Side chains addition (SCWRL)

[Jusot et al., JCIM 2018]
Exhaustive sampling of small cyclic peptides

Sampling of $\Phi_1$, $\Psi_1$, $\Phi_5$, $\Psi_5$ (+ all $\omega$)

Inverse kinematics:
- 0 solution
- 1 to 16 solutions

Check collisions

Side chains addition (SCWRL)

Relaxation (Amber)

[Jusot et al., JCIM 2018]
Exhaustive sampling of small cyclic peptides

Sampling of $\Phi_1, \Psi_1, \Phi_5, \Psi_5 ( + \text{ all } \omega )$

Inverse kinematics:
- 0 solution
- 1 to 16 solutions

Check collisions

Side chains addition (SCWRL)

Relaxation (Amber)

Add node to adjacency graph

[Jusot et al., JCIM 2018]
Exhaustive sampling of small cyclic peptides

Peptide c(RGDkV)  
Exhaustive search : 839061 conformations founds

- ARG
- GLY
- ASP
- d-LYS
- VAL

REMD

Energy min
Frequencies min

Jusot et al., JCIM 2018
Course Contents

- Loop sampling
- Path sampling on energy landscapes

NOT COVERED
- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design
Algorithm : Construct_RRT

input : the search-space $C$;
        the root $q_{init}$ and the goal $q_{goal}$;
output : the tree $\tau$;
begin
    $\tau \leftarrow \text{InitTree}(q_{init})$;
    while not StopCondition($\tau$, $q_{goal}$) do
        $q_{rand} \leftarrow \text{SampleConf}(C)$;
        $q_{near} \leftarrow \text{BestNeighbor}(\tau, q_{rand})$;
        $q_{new} \leftarrow \text{Expand}(q_{near}, q_{rand})$;
        if not TooSimilar($q_{near}, q_{new}$) then
            AddNewNode($\tau$, $q_{new}$);
            AddNewEdge($\tau$, $q_{near}, q_{new}$);
    end
Types of Path Planning Problems

- Basic RRT algorithm
Types of Path Planning Problems

- Feasible path planning
  - Path-quality criterion
    - Path length
    - Path duration
  - Configuration-cost function
    - Clearance
    - Energy
    - Integral of cost
    - Maximal cost
    - ...
Examples of cost-space path planning problems

**In robotics**

- **Admissible tension-range**
- **6-D wrench-set**

**In structural biology**

- Examples of cost-space path planning problems

- In robotics
  - Admissible tension-range
  - 6-D wrench-set

- In structural biology
  - Examples of cost-space path planning problems
Cost-space Path Planning: the T-RRT algorithm

Stochastic transition test

\[ p_{ij} = \begin{cases} \exp\left(-\frac{c_j - c_i}{kT}\right) & \text{if } c_j - c_i > 0 \\ 1 & \text{otherwise} \end{cases} \]

Adaptive parameter

[Jaillet et al., T-RO, 2010] [Jaillet et al., J Comput Chem, 2011]
T-RRT: Temperature Self-Tuning

Algorithm: transitionTest \((T, c_i, c_j)\)

- **input**: the current temperature \(T\) and the increase rate \(T_{rate}\)
- **output**: true if the transition is accepted, false otherwise

\[
\text{if } c_j \leq c_i \text{ then return True}
\]
\[
\text{if } \exp\left(-\frac{(c_j - c_i)}{T}\right) > 0.5 \text{ then}
\]
\[
T \leftarrow T / 2(c_j - c_i) / \text{costRange}(T) \quad \text{return True}
\]
\[
\text{else}
\]
\[
T \leftarrow T \cdot 2^{T_{rate}} \quad \text{return False}
\]

\(T_{rate} = 0.1\) yields good performance in most cases

Determines **greediness**

\[T \uparrow \downarrow\]
Transition-based RRT (T-RRT)

- Very different behavior compared to a basic Monte Carlo method

[Jaillet et al., T-RO, 2010] [Jaillet et al., J Comput Chem, 2011]
Transition-based RRT (T-RRT)

[Jaillet et al., T-RO, 2010] [Jaillet et al., J Comput Chem, 2011]

- An interesting property: tends to find "minimal work" paths
  … but no guarantees

\[ W(P) = \int_{s^*} \frac{\partial C}{\partial s} ds + \varepsilon \int ds \]
Types of Path Planning Problems
Optimal Path Planning: the RRT* algorithm

[Karaman and Frazzoli, RSS, 2010]

- Asymptotic convergence to the global optimum
- Basically conceived to minimize and additive cost along the path (e.g. length)

"Re-wiring" process
the edge created is the one minimizing the cost from $q_{\text{init}}$ to $q_{\text{new}}$ among the neighbors of $q_{\text{new}}$
Types of Path Planning Problems

- Feasible path planning
- Path-quality criterion
  - Path length
  - Path duration
  - ...>
- Configuration-cost function
  - Clearance
  - Energy
  - ...>
- Optimal path planning
- Path-quality criterion based on a configuration-cost function
  - Integral of cost
  - Maximal cost
  - ...>
- Cost-space path planning
Optimal Path Planning in Continuous Cost-Spaces

[Devaurs et al., IEEE TASE, 2016]

• Idea: combine the beneficial concepts underlying RRT* and T-RRT
  - cost-based node creation (configuration cost)
  - quality-based edge management (path quality)
• Two new algorithms
  - Transition-based RRT* (T-RRT*)
  - Anytime T-RRT (AT-RRT)
• Theoretical guarantees
  - probabilistic completeness
  - asymptotic optimality
Optimal Path Planning in Continuous Cost-Spaces

[Devaurs et al., IEEE TASE, 2016]

---

**Algorithm**: Anytime Transition-based RRT (AT-RRT)

**Input**: the optimal path planning problem \((C, q_{\text{init}}, q_{\text{goal}}, c, c_p)\)

**Output**: the graph \(G\)

1. \(G \leftarrow \text{initGraph}(q_{\text{init}})\)
2. **while not** stoppingCriteria\((G)\) **do**
3. \(q_{\text{rand}} \leftarrow \text{sampleRandomConfiguration}(C)\)
4. \(q_{\text{near}} \leftarrow \text{findNearestNeighbor}(G, q_{\text{rand}})\)
5. \(q_{\text{new}} \leftarrow \text{extend}(q_{\text{near}}, q_{\text{rand}})\)
6. **if** \(q_{\text{new}} \neq \text{null and}
7. \)transitionTest\((G, c(q_{\text{near}}), c(q_{\text{new}}))\) **then**
8. \( \text{addNewNode}(G, q_{\text{new}})\)
9. \( \text{addNewEdge}(G, q_{\text{near}}, q_{\text{new}})\)
10. **if** solutionPathExists\((G, q_{\text{init}}, q_{\text{goal}})\) **then**
11. \( \text{addUsefulCycles}(G, q_{\text{new}}, c_p)\)
12. **return** \(G\)

---

**Algorithm**: addUsefulCycles \((\mathcal{G}, q_{\text{new}}, c_p)\)

**Input**: the dimension \(d\) of the \(C\)-space
the \(\gamma\) constant derived from the volume of \(C_{\text{free}}\)

1. \(n \leftarrow \text{numberOfNodes}(\mathcal{G})\)
2. \(Q_{\text{near}} \leftarrow \text{nodesInBall}(\mathcal{G}, q_{\text{new}}, \gamma (\log(n) / n)^{1/d})\)
3. **foreach** \(q_n \in Q_{\text{near}}\) **do**
4. \(\pi_g \leftarrow \text{pathInGraph}(\mathcal{G}, q_{\text{new}}, q_n)\)
5. \(\pi_s \leftarrow \text{pathInSpace}(q_{\text{new}}, q_n)\)
6. **if** \(c_p(\pi_s) < c_p(\pi_g)\) **and** isCollisionFree\((\pi_s)\) **then**
7. \(\text{addNewEdge}(\mathcal{G}, q_{\text{new}}, q_n)\)
An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

[Jaillet et al., J Comput Chem, 2011]

- **Alanine dipeptide**

![Alanine dipeptide structure](image)

**Energy function:**
- parm96 AMBER force field
- implicit solvent (GB)

**Energy minima:**

<table>
<thead>
<tr>
<th></th>
<th>$p_{II}$</th>
<th>$\alpha_R$</th>
<th>$\alpha_L$</th>
<th>$C_7^{\alpha_x}$</th>
<th>$\alpha_p$</th>
<th>$C_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
<td>-67</td>
<td>-63</td>
<td>47</td>
<td>50</td>
<td>-148</td>
<td>-146</td>
</tr>
<tr>
<td>$\psi$</td>
<td>144</td>
<td>-44</td>
<td>51</td>
<td>-138</td>
<td>-70</td>
<td>162</td>
</tr>
<tr>
<td>$E$</td>
<td>0.3</td>
<td>1.1</td>
<td>4.4</td>
<td>4.2</td>
<td>1.7</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Main transition states:**

<table>
<thead>
<tr>
<th></th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
<th>$s_5$</th>
<th>$s_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
<td>0</td>
<td>3</td>
<td>72</td>
<td>74</td>
<td>-111</td>
<td>-142</td>
</tr>
<tr>
<td>$\psi$</td>
<td>95</td>
<td>-90</td>
<td>137</td>
<td>-8</td>
<td>10</td>
<td>-118</td>
</tr>
<tr>
<td>$E$</td>
<td>7.3</td>
<td>7.7</td>
<td>7.3</td>
<td>7.7</td>
<td>3.4</td>
<td>2.6</td>
</tr>
</tbody>
</table>
An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

[Jaillet et al., J Comput Chem, 2011]

- **Alanine dipeptide**: Example of one run of T-RRT for $C_5 \rightarrow \alpha_L$
  
  CPU time $\approx 1$ sec. (on a single processor)

![Diagram showing structural transitions](image)
An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

[Jaillet et al., J Comput Chem, 2011]

- **Alanine dipeptide**: Transition $\alpha_L \leftrightarrow C_7^{ax}$

---

<table>
<thead>
<tr>
<th></th>
<th>I (%)</th>
<th>II (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_L \rightarrow C_7^{ax}$</td>
<td>62</td>
<td>38</td>
</tr>
<tr>
<td>$C_7^{ax} \rightarrow \alpha_L$</td>
<td>60</td>
<td>40</td>
</tr>
</tbody>
</table>
An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

[Jaillet et al., J Comput Chem, 2011]

- **Alanine dipeptide**: Transition $\alpha_R \leftrightarrow P_{II}$
An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

[Jaillet et al., J Comput Chem, 2011]

- **Alanine dipeptide**: Transition $\alpha_R \leftrightarrow C_{7}^{ax}$

<table>
<thead>
<tr>
<th></th>
<th>I (%)</th>
<th>II (%)</th>
<th>III (%)</th>
<th>IV (%)</th>
<th>V (%)</th>
<th>VI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_R \rightarrow C_{7}^{ax}$</td>
<td>21</td>
<td>54</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>$C_{7}^{ax} \rightarrow \alpha_R$</td>
<td>9</td>
<td>17</td>
<td>31</td>
<td>27</td>
<td>11</td>
<td>5</td>
</tr>
</tbody>
</table>
The Ordering-and-Pathfinding Problem

«Off-road» TSP
- Paths (and costs) between pairs of states are unknown \textit{a priori}
- A continuous cost-space has to be explored

Two-level problem:
- Low-level (cost-based path planning): Connect pairs of states
- High-level (ordering / classical TSP): Find the optimal order to visit all states

\textit{Can be interleaved and solved in an anytime manner}
The Ordering-and-Pathfinding Problem: Solved with the Anytime Multi-T-RRT Algorithm

[Devaurs et al., IEEE IROS, 2014]

**T-RRTs construction**
- Build $n$ trees rooted at the given states
- Until all trees are connected

**Useful cycle addition**
- Incremental local improvements
- Guarantees asymptotic convergence to the global optimum
A more interesting peptide: Met-enkephalin

[Devaurs et al., IEEE TNB, 2015]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Conformation</th>
<th>4-sign code</th>
<th>Energy (kcal/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+ + + +</td>
<td>-217.9</td>
</tr>
<tr>
<td>□</td>
<td></td>
<td>+ + +</td>
<td>-215.9</td>
</tr>
<tr>
<td>△</td>
<td></td>
<td>+ + + +</td>
<td>-212.7</td>
</tr>
<tr>
<td>▼</td>
<td></td>
<td>* + + +</td>
<td></td>
</tr>
</tbody>
</table>

sign of \{\psi_2, \psi_3, \psi_4, \phi_3\}

A more interesting peptide: Met-enkephalin

[Devaurs et al., IEEE TNB, 2015]

Finding paths between minima: Multi-T-RRT (100 runs)

- First solution CPU time = ~ 2 min./run
- Cycle addition (refinement) CPU time = 10 min./run
A more interesting peptide: Met-enkephalin

Finding paths between minima: Multi-T-RRT (100 runs)

- First solution CPU time = ~ 2 min./run
- Cycle addition (refinement) CPU time = 10 min./run

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Conformation</th>
<th>4-sign code</th>
<th>Energy (kcal/mol)</th>
<th>Transition To</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Trans Prob</td>
<td>Path Cost (MW)</td>
</tr>
<tr>
<td><img src="image1.png" alt="Symbol" /></td>
<td><img src="image2.png" alt="Conformation" /></td>
<td>++-++</td>
<td>-217.9</td>
<td>-</td>
</tr>
<tr>
<td><img src="image3.png" alt="Symbol" /></td>
<td><img src="image4.png" alt="Conformation" /></td>
<td>---</td>
<td>-216.5</td>
<td>0.64</td>
</tr>
<tr>
<td><img src="image5.png" alt="Symbol" /></td>
<td><img src="image6.png" alt="Conformation" /></td>
<td>+++</td>
<td>-215.9</td>
<td>0.63</td>
</tr>
<tr>
<td><img src="image7.png" alt="Symbol" /></td>
<td><img src="image8.png" alt="Conformation" /></td>
<td>+++++</td>
<td>-212.7</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Parallelization of RRT-based algorithms

Hybrid shared/distributed-memory implementation: For execution on computer clusters

[Estaña et al., Parallel Comput., 2018]

Parallel Multi-TRRT

Study of IDPs/IDRs
Question?

juan.cortes@laas.fr