Mining protein flexibility with robotics-inspired methods

Juan Cortés

Winter School *Algorithms in Structural Bioinformatics*
Mondonville (France), 25-29 November 2013
Course Contents

- Basic notions
- Path planning algorithms

- Path planning in Structural Biology: Overview
- Exploring energy landscapes

- Computing loop and domain motions
- Computing ligand access/exit pathways
Course Contents

- Basic notions
- Path planning algorithms
- Path planning in Structural Biology: Overview
- Exploring energy landscapes
- Computing loop and domain motions
- Computing ligand access/exit pathways
Course Contents

- Basic notions
- Path planning algorithms

Path planning in Structural Biology: Overview
- Exploring energy landscapes

- Computing loop and domain motions
- Computing ligand access/exit pathway
Course Contents

- Basic notions
- Path planning algorithms
- Path planning in Structural Biology: Overview
- Exploring energy landscapes
- Computing loop and domain motions
- Computing ligand access/exit pathways
Course Contents

- Basic notions
- Path planning algorithms
- Path planning in Structural Biology: Overview
- Exploring energy landscapes
- Computing loop and domain motions
- Computing ligand access/exit pathways
Course Contents

• Basic notions
• Path planning algorithms

• Path planning in Structural Biology: Overview
• Exploring energy landscapes

• Computing loop and domain motions
• Computing ligand access/exit pathways
Course Contents

• Basic notions
• Path planning algorithms

• Path planning in Structural Biology: Overview
• Exploring energy landscapes

• Computing loop and domain motions
• Computing ligand access/exit pathways
Mining protein flexibility with robotics-inspired methods

Basic notions
The Origins: Robot Programming

The **Path Planning** Problem

Solution
Geometric Formulation

2R Mechanism $\Leftrightarrow$ Configuration space

$2R$ Mechanism

obstacles

$\text{CS}_{\text{free}}$

$\text{CS}_{\text{obst}}$
A robot **configuration** is a specification of the positions of all robot points relative to a fixed coordinate system.

Usually a configuration is expressed as a **vector** of position/angle parameters.
Configuration (Pose) of a Rigid Body

Six parameters: $q = \{x_o, y_o, z_o, \gamma, \beta, \alpha\}$

Homogeneous matrix transformation:

$$
W_{TO} = \begin{pmatrix}
\cos \beta \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha & x_o \\
\cos \beta \sin \alpha & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & y_o \\
-\sin \beta & \sin \gamma \cos \beta & \cos \gamma \cos \beta & z_o \\
0 & 0 & 0 & 1
\end{pmatrix}
$$
**Configuration of an Articulated Mechanism**

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

Homogeneous matrix transformation:

\[
i^{-1}T_i = \begin{pmatrix}
    \cos \theta_i & -\sin \theta_i & 0 & a_{i-1} \\
    \sin \theta_i \cos \alpha_{i-1} & \cos \theta_i \cos \alpha_{i-1} & -\sin \alpha_{i-1} & -d_i \sin \alpha_{i-1} \\
    \sin \theta_i \sin \alpha_{i-1} & \cos \theta_i \sin \alpha_{i-1} & \cos \alpha_{i-1} & d_i \cos \alpha_{i-1} \\
    0 & 0 & 0 & 1
\end{pmatrix}
\]

\[0T_n = 0T_1T_2 \ldots n^{-1}T_n\]
Configuration of an 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 \( a_{i-1} \neq 0 \)
Configuration of an Articulated Mechanism

*Open* kinematic chain

*Closed* kinematic chain

Loop-closure constraints: $f(q) = 0$, $f$ non-linear system of equations.

The configuration space is not a parameterizable manifold.
A **metric** or **distance** function $d$ in $C$ is a map:

$$d: (q,q') \in C^2 \rightarrow d(q,q') \geq 0 \text{ (positive real)}$$

such that:

- $d(q,q') = 0$ if and only if $q_1 = q_2$
- $d(q,q') = d(q',q)$
- $d(q,q') \leq d(q,q'') + d(q'',q')$

**Exemple:**

$q = (x,y,\theta)$, $q' = (x',y',\theta')$ with $\theta, \theta' \in [0,2\pi)$

$$\alpha = \min\{|\theta-\theta'|, 2\pi-|\theta-\theta'|\}$$

$$d(q,q') = \sqrt{(x-x')^2 + (y-y')^2 + (\alpha \rho)^2}$$

where $\rho$ is the maximal distance between the reference point and a robot point
A **path** in $C$ is a piece of continuous curve connecting two configurations $q$ and $q'$:

$$\tau : s \in [0, 1] \rightarrow \tau(s) \in C$$

$$s \rightarrow s' \implies d(\tau(s), \tau(s')) \rightarrow 0$$

A **trajectory** is a path parameterized by time:

$$\tau : t \in [0, T] \rightarrow \tau(t) \in C$$
Mining protein flexibility with robotics-inspired methods

Sampling-based path planning algorithms

Juan Cortés

Winter School *Algorithms in Structural Bioinformatics*
Mondonville (France), 25-29 November 2013
Contents

- Introduction
- Principle of main algorithms: PRM, RRT
- Main ingredients:
  - Sampling method
  - Connection strategy
  - Steering methods (kinematically feasible motions)
  - Collision detection (geometrically feasible motions)
- Some PRM variants
- Some RRT variants
Sampling-Based Path Planning: Motivation

- **Complete planners**
  - High computational complexity
  - Limited to a few DOFs

- **Heuristic planners**
  - Local minima problems
  - Too unreliable

**Real applications**:  
- Many DOFs, complex geometry  
- Constrained search spaces  
  (many local minima)
Sampling-Based Path Planning: Weaker Completeness

- Trade a **limited amount of completeness** against a major **gain in computing efficiency**

**Probabilistic completeness**:
- If there is a solution path, the probability that the planner will find is a (fast growing) function that goes to 1 as running time increases
- Unable to determine if a problem is unsolvable
Sampling-Based Path Planning: Main Strategies

- **Roadmap methods** (*PRM and its variants*)
  - Pre-compute roadmap capturing the Cspace connectivity
  - Re-use roadmap for answering queries

- **Diffusion methods** (*Ariane's Clew Algorithm, RRT,...*)
  - Solve a specific query by expanding roadmaps rooted at the query configurations
PRM Principle

- Roadmap construction
- Connect queries
- Graph search
- Path smoothing

[Kavraki, Švestka, Latombe and Overmars 96]
RRT Principle

Diffusion method:
Iteratively expand search tree(s) rooted at the initial (and goal configurations)

[LaValle 98][LaValle and Kuffner 01]
Basic Ingredients

- **Sampling** method
- **Connection** strategy
- **Steering methods** (kinematically feasible motions)
- **Collision detection** (geometrically feasible motions)
Sampling Strategy

- Uniform sampling
  - basic strategy
  - pseudo-random or quasi-random
Sampling Strategy

- Uniform sampling
- Multi-stage sampling
  - expand roadmap in “difficult” regions
    using an heuristic weight function $w(q)$
Sampling Strategy

- Uniform sampling
- Multi-stage sampling
- Obstacle-sensitive sampling
  - place more samples near boundary of obstacles: OBPRM, Gaussian sampling
Sampling Strategy

- Uniform sampling
- Multi-stage sampling
- Obstacle-sensitive sampling
- Free-space dilatation
  - allow some penetration to widen the narrow passages

$\Rightarrow$ penetration distances are difficult to compute
Sampling Strategy

- Uniform sampling
- Multi-stage sampling
- Obstacle-sensitive sampling
- Free-space dilatation
- Medial axis sampling
  - try to retract samples inside the free-space along a random direction
Sampling Strategy

- Uniform sampling
- Multi-stage sampling
- Obstacle-sensitive sampling
- Free-space dilatation
- Medial axis sampling
- Visibility-based sampling
  - makes use of reachability notion to compute small roadmaps and also control termination of learning stage
Connection Strategy

- Try all connections
- Connection to $k$-nearest neighbors
- Connect with nodes at distance $< l$
- Acyclic connections
- ...

- **Aim**: limit (expensive) connection attempts
Steering Method

• Compute a feasible path connecting two given samples

• Requirements:
  • **Deterministic**: eliminates the need to store local paths
  • **Fast**: required for efficient roadmap construction and quasi-instantaneous queries
Steering Method

Examples

- Linear
- Manhattan
- Reeds-Sheep
- Combined methods (Lin/RS)
- Closed chains
- Complex motions (e.g. human walk...)

Juan Cortés

Mining protein flexibility with robotics-inspired methods

AlgoSB School, November 2013
Steering Method

- **Controllability** issues vs. CSpace **topology**

- **Small time controllable (STC) systems**
  all point in a neighborhood of $p$ can be attained in a small time $t$
  
  Any collision-free path can be approximated by a **finite**
  sequence of feasible paths

Example: forward/backward car
  vs. forward-only car
• **Controllability** issues vs. CSpace **topology**

• **Small time controllable (STC)** systems

• STC steering methods

• All STC steering methods induce the same topology

Steering Method

\[ \text{Euclidean} = \text{Manhattan} \]
Collision Checking

- Sampling based planners need for efficient algorithms (90% of computing time)

- Collision vs. Distance
- Static vs. Dynamic
Most methods use **Hierarchical bounding volumes**

- spheres
- ABB
- OBB
- kd-trees
Dynamic Collision Checking

- Fixed resolution
  - Incremental (sequential)
- Dichotimic

Missed collision!
Dynamic Collision Checking

- **Adaptive resolution**
  - More expensive distance computation
  - Less steps (far from obstacles)
  - Guarantees collision avoidance
Visibility-based PRM

Visible sets:

Euclidean

Manhattan like

[Siméon, Laumond and Nissoux 00]
Visibility-based PRM

Two types of nodes:

- guards
- connectors

[Siméon, Laumond and Nissoux 00]
Visibility-based PRM

Computation?

No explicit knowledge of the reachable sets

[Siméon, Laumond and Nissoux 00]
Visibility-based PRM

Algorithm:

- Generate guards and connectors randomly
- Stop after $\#trv$ failures

[Siméon, Laumond and Nissoux 00]
Visibility-based PRM

Property:

The (estimated) percentage of non-covered free-space is \( #try^{-1} \)

[Siméon, Laumond and Nissoux 00]
Visibility-based PRM

[Siméon, Laumond and Nissoux 00]

Advantages:

- connectivity of complex spaces captured in small roadmap

- control of the algorithm wrt. estimated coverage ($#\text{try}^{-1}$)

Puzzle-like example
Visibility-based PRM

Advantages:

• connectivity of **complex spaces** captured in **small roadmap**

• **control of the algorithm** wrt. estimated coverage ($#try^{-1}$)

[Siméon, Laumond and Nissoux 00]
Path Deformation Roadmaps (PDR)

Motivation:
- **Cycles** are important for providing alternative and shorter solution paths
- Keep **compactness** to avoid extra cost of testing (redundant) connections

Principle: Add “useful” cycles to a tree obtained by the Visibility-PRM algorithm

[Jaillet and Siméon 06]
Algorithm : Construct_RRT

\textbf{input} : the search-space } \mathcal{C};
\text{ the root } q_{\text{init}} \text{ and the goal } q_{\text{goal}};

\textbf{output} : the tree } \tau;

\textbf{begin}
\quad \tau \leftarrow \text{InitTree}(q_{\text{init}});
\quad \textbf{while not StopCondition}(\tau, q_{\text{goal}}) \text{ do}
\quad \quad q_{\text{rand}} \leftarrow \text{SampleConf}(\mathcal{C});
\quad \quad q_{\text{near}} \leftarrow \text{BestNeighbor}(\tau, q_{\text{rand}});
\quad \quad q_{\text{new}} \leftarrow \text{Expand}(q_{\text{near}}, q_{\text{rand}});
\quad \quad \text{if not TooSimilar}(q_{\text{near}}, q_{\text{new}}) \text{ then }
\quad \quad \quad \text{AddNewNode}(\tau, q_{\text{new}});
\quad \quad \quad \text{AddNewEdge}(\tau, q_{\text{near}}, q_{\text{new}});
\quad \quad \textbf{end}
\textbf{end}
RRT: Implicit Voronoi bias

Nice property!
**RRT Variants**

**Expansion mode**
- **Single-step**
  “RRT-EXTEND”
- **Iterate while feasible**
  “RRT-CONNECT”

**Exploration strategy**
- **Mono-directional** :
  Explore from $q_{\text{init}}$
- **Bi-directional**
  Explore from $q_{\text{init}}$ and $q_{\text{goal}}$
- **Multiple trees**
  Explore from $q_{\text{init}}$, $q_{\text{goal}}$ and samples

➤ The best strategy depends on the problem
RRT: Sampling Parameters

Holonomous system, without dynamics

- Sample in C-space (degrees of freedom)

Differential constraints, kinodynamics, ...

- Sample control variables

State transition equation:
\[ \dot{x} = f(x,u) \]
\[ x \in C \quad \Leftarrow \text{configurations} \]
\[ u \in U \quad \Leftarrow \text{inputs} \]

Euler integration:
\[ x_{\text{new}} \approx x + f(x,u)\Delta t \]
RRT: Sampling Parameters

Example: *Forward-only car-like robot*

\[ x \in [0,s] \quad \Leftarrow \text{Maximum speed} \]
\[ u \in [0,\theta] \quad \Leftarrow \text{Bounded turning angle} \]

\[ \dot{x}_1 = u_1 \cos(u_2) \]
\[ \dot{x}_2 = u_1 \sin(u_2) \]

\[ x_1(t + \Delta t) = x_1(t) + u_1 \cos(u_2) \cdot \Delta t \]
\[ x_2(t + \Delta t) = x_2(t) + u_1 \sin(u_2) \cdot \Delta t \]
RRT’s Drawback: Metric Sensibility

Standard metric: Euclidean distance in CS
- Does not consider motion constraints (e.g. obstacles).
- Difficult to adapt to complex mechanisms.

Exhausted nodes

[Cheng and LaValle, 03]
### Basic RRT Improvements

[Cortés, Jaillet and Siméon, 07]

#### Reducing exhausted nodes effect

- [Cheng and LaValle, 03]
- [Kim et al., 03]
- [Urmson and Simmons, 03]
  - Remove nodes after $l$ consecutive expansion failures ($l$-RRT).
  - Select $q_{\text{near}}$ among the $k$ nearest neighbors ($k$-RRT).

#### Simplified task-adapted metric

- [Plaku and Kavraki, 06]
  - Consider a subset of $m$ parameters that are more significant for with respect to the task ($m$-RRT).

<table>
<thead>
<tr>
<th></th>
<th>ExA</th>
<th>ExB</th>
<th>ExC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>43.6</td>
<td>562.2</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>187.7</td>
<td>1650.2</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>79</td>
<td>136</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>11186</td>
<td>86303</td>
<td></td>
</tr>
<tr>
<td><strong>l-RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>7.2</td>
<td>310.9</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>14.9</td>
<td>881.2</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>263</td>
<td>4534</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>3093</td>
<td>61448</td>
<td></td>
</tr>
<tr>
<td><strong>k-RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>6.1</td>
<td>169.3</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>6.9</td>
<td>911.7</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>202</td>
<td>270</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>1119</td>
<td>10259</td>
<td></td>
</tr>
<tr>
<td><strong>lk-RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>6.1</td>
<td>62.2</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>6.0</td>
<td>237.6</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>227</td>
<td>733</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>1242</td>
<td>8250</td>
<td></td>
</tr>
<tr>
<td><strong>m-RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>4.0</td>
<td>17.2</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>7.4</td>
<td>141.3</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>104</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>3436</td>
<td>16202</td>
<td></td>
</tr>
<tr>
<td><strong>lk-Rm-RRT</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$avrT$</td>
<td>1.7</td>
<td>5.1</td>
<td></td>
</tr>
<tr>
<td>$SN$</td>
<td>1.4</td>
<td>6.1</td>
<td></td>
</tr>
<tr>
<td>$Nn$</td>
<td>234</td>
<td>457</td>
<td></td>
</tr>
<tr>
<td>$Ns$</td>
<td>1059</td>
<td>3727</td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

How to reduce the influence of the sampling domain on the RRT performance?
Dynamic Domain RRT (DD-RRT)

Main Motivations:
- Control the balance between exploration and refinement
- Incorporate the obstacles into Voronoi bias

[Yershova, Jaillet, Siméon, and LaValle 05]
Dynamic Domain RRT (DD-RRT)

[Vershova, Jaillet, Siméon, and LaValle 05]

Voronoi Diagram in $\mathbb{R}^2$
Dynamic Domain RRT (DD-RRT)

[Vershova, Jaillet, Siméon, and LaValle 05]

Voronoi Diagram in $\mathbb{R}^2$

Non-frontier nodes
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Voronoi Diagram in $\mathbb{R}^2$

Frontier nodes
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Refinement vs. Expansion

- Where will the random sample fall?
- How to control the behavior of RRT?
Dynamic Domain RRT (DD-RRT)

Determining the Boundary

Expansion dominates

Balanced refinement / expansion

The tradeoff depends on the C-space bounds
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Bias Toward some Directions

Vertical Bias

Horizontal Bias

The C-space bounds can lead to uncontrolled bias toward some directions
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Sampling domain for the basic RRT
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Sampling domain for the *basic* RRT
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Visibility-based clipping of the sampling domain

Nice idea, but how can this be done in practice?
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Dynamic-Domain RRT Bias

Tradeoff between nearest neighbor calls and collision detection calls
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Tuning of the Sampling Domain
Dynamic Domain RRT (DD-RRT)

[Yershova, Jaillet, Siméon, and LaValle 05]

Adaptation to Surrounding Obstacles
Dynamic Domain RRT (DD-RRT)

Adaptive DD-RRT Construction

```java
BUILD_ADAPTIVE_DD_RRT(q_init)
1
2    T.init(q_init);
3    for k = 1 to K do
4        repeat
5            q_rand ← RANDOM_CONFIG();
6            q_near ← NEAR_NEIGH(q_rand, T, d_near);
7            until (d_near < q_near.radius)
8            if q_new ← CONNECT(T, q_rand, q_near)
9                q_new.radius = ∞;
10               if q_near.radius ≠ ∞;
11                  q_near.radius = (1 + α) × q_near.radius;
12                  T.add_vertex(q_new);
13                  T.add_edge(q_near, q_new);
14            else
15                if q_near.radius = ∞
16                    q_near.radius = R;
17                else
18                  q_near.radius = (1 − α) × q_near.radius;
19        Return T;
```

- Automatic Balance between Exploration and Refinement
- Local Adaptation Function of the Free Space Shape
Advantages:
- General planning scheme for a large class of problems
- Practical efficiency
- Conceptual robustness and simple implementation

Drawbacks:
- Weak completeness guarantee
- Difficulty for evaluating/controlling the performance
- Provide sub-optimal solutions:
  - Only some variants (e.g. PRM*, RRT*, T-RRT) are able to provide good-quality solutions, but are more computationally expensive
  - Post-processing is required to (locally) optimize solutions
Sampling-Based Path Planning: Textbooks

- Steven M. LaValle
  *Planning Algorithms*
  Cambridge University Press, 2006

  *Principles of Robot Motion: Theory, Algorithms, and Implementations*
Mining protein flexibility with robotics-inspired methods

Path-planning-based methods in structural bioinformatics

Juan Cortés
Winter School *Algorithms in Structural Bioinformatics*
Mondonville (France), 25-29 November 2013

Robotic Algorithms for Molecular Modeling
December 2009
Different Approaches

• Apply path planning algorithms to explore the molecular force field

• Explore with geometric constraints and then refine using energy functions

• Combine geometric and energetic approaches
Robots-Inspired Methods in Structural Bioinformatics: Some References

Groups
- J.-C. Latombe (Stanford, USA)
- L. Kavraki (Rice Univ., USA)
- N. Amato (TAMU, USA)
- A. Shehu (GMU, USA)
- G. Chirikjian (JHU, USA)
- B. Donald (Duke, USA)
- D. Halperin (TAU, Israel)
- O. Brock (TU Berlin, Germany)
- L. Ros, J.M. Porta, V. Ruiz (IRI, Spain)
- J. Cortés, T. Siméon (LAAS, France)

Surveys

Books
Exploring Energy Landscapes: T-RRT

[Jaillet et al., T-RO, 2010] [Jaillet et al., J Comput Chem, 2011]
Basic Path Planning vs. Cost-based Path Planning

- Binary configuration space
- Obtain a feasible path
- (Optionally) minimize length

- Continuous cost space
- Obtain a good-quality path
- Minimize: cost variation, cost integral, maximum cost, …
Transition-based RRT (T-RRT): The Basic Principle

Favor exploration of low-cost (i.e. low-energy) regions.
Combining ideas of robot motion planning and statistical physics:

• Efficiency of RRTs with implicit Voronoi-biased exploration

\[
\text{TransitionTest}(E(q_{\text{near}}), E(q_{\text{new}}), d) \]

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

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

**Algorithm** : Transition-based RRT

**input** : the conformational space \( C \)
- the energy function \( E : C \rightarrow \mathbb{R} \)
- the initial conformation \( q_{\text{init}} \)
- the target conformation \( q_{\text{goal}} \) (optional)
- the extension step-size \( \delta \)

**output**: the tree \( \mathcal{T} \)
- \( \mathcal{T} \leftarrow \text{initTree}(q_{\text{init}}) \)

**while not** \( \text{stopCondition}(\mathcal{T}, q_{\text{goal}}) \)** do

- \( q_{\text{rand}} \leftarrow \text{sampleRandomConformation}(C) \)
- \( q_{\text{near}} \leftarrow \text{findNearestNeighbor}(\mathcal{T}, q_{\text{rand}}) \)
- **if** \( \text{refinementControl}(\mathcal{T}, q_{\text{near}}, q_{\text{rand}}) \)** then
  - \( q_{\text{new}} \leftarrow \text{extend}(q_{\text{near}}, q_{\text{rand}}, \delta) \)
  - **if** \( q_{\text{new}} \neq \text{null} \) and
  - \( \text{transitionTest}(\mathcal{T}, E(q_{\text{near}}), E(q_{\text{new}})) \)** then
    - \( \text{addNewNode}(\mathcal{T}, q_{\text{new}}) \)
    - \( \text{addNewEdge}(\mathcal{T}, q_{\text{near}}, q_{\text{new}}) \)
Transition-based RRT (T-RRT) : The Transition Test

• Based on the **Metropolis criterion** :

\[ P_{ij} = \begin{cases} 
\exp\left(-\frac{E_j - E_i}{K \cdot T}\right) & \text{if } E_j - E_i > 0 \\
1 & \text{otherwise}
\end{cases} \]

\[ K = \text{Boltzmann constant} \quad \text{OR} \quad K = \frac{E_{init} + E_{goal}}{2} \quad \text{(normalization coefficient)} \]

\( T : \text{“Temperature” parameter} \rightarrow \text{control of transition test difficulty} \)
Transition-based RRT (T-RRT) : Temperature Self-Tuning

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

Algorithm : transitionTest ($T$, $E_i$, $E_j$)

input : the energy threshold $E_{max}$  
the current temperature $T$  
the temperature increase rate $T_{rate}$

output: true if the transition is accepted, false if not

if $E_j > E_{max}$ then return False
if $E_j \leq E_i$ then return True
if $\exp\left(-\frac{(E_j - E_i)}{(K \cdot T)}\right) > 0.5$ then
    $T \leftarrow \frac{T}{2^{\left(E_j - E_i\right)/(0.1 \cdot \text{energyRange}(T))}}$
    return True
else
    $T \leftarrow T \cdot 2^{T_{rate}}$
    return False

Determines greediness
Transition-based RRT (T-RRT) : Temperature Self-Tuning

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

- An undesired side effect...
- Desired behavior

“Useless” refinement of explored regions

Global/homogeneous coverage
Refinement slows down the expansion by decreasing the temperature.

- Frontier nodes: **Expansion**
- Non frontier nodes: **Refinement**

Control the Expansion / Refinement ratio
Transition-based RRT (T-RRT): Refinement Control

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

Axiom:

\[
\text{Voronoi Volume(\text{non frontier nodes})} \ll \text{Voronoi Volume(\text{frontier nodes})}
\]

\[
d(q_{\text{near}}, q_{\text{rand}}) > \delta \rightarrow \text{expansion}
\]

\[
d(q_{\text{near}}, q_{\text{rand}}) \leq \delta \rightarrow \text{refinement}
\]

- If \( \text{ratio(refinement)} > \text{ratio}_{\text{max}} \): reject new node
- \( \delta = d \)
- Experimental values:
  - (expansion step)
    \[
    \text{ratio}_{\text{max}} = 50%
    \]
Transition-based RRT (T-RRT) : Experiments

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

- Experiments on 2D Cost Maps
Transition-based RRT (T-RRT) : Experiments

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

- Experiments on 2D Cost Maps
Transition-based RRT (T-RRT) : Experiments

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

- Experiments on 2D Cost Maps

**Property**

\[
W(P) = \int_{s_+}^{s} \frac{\partial E^+}{\partial s} ds + \varepsilon \int_{s} ds
\]

: An Interesting

T-RRT  Optimal path : Minimum of \(W(P)\)
Transition-based RRT (T-RRT) : Experiments

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

- Alanine dipeptide

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>$C_7^{ax}$</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>
Transition-based RRT (T-RRT) : Experiments

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

- **Alanine dipeptide**

**T-RRT settings:**
- Variables = dihedral angles
- Max. angular variation = 5°

**Computational performance:**
(Non-optimized implementation)
- finding all minima : 15 min.
- transition paths : 1-5 min.
Transition-based RRT (T-RRT) : Experiments

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

- **Alanine dipeptide** : Example of one run
Transition-based RRT (T-RRT) : Experiments

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

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

<table>
<thead>
<tr>
<th>I (%)</th>
<th>II (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_L \rightarrow C_7^{ax}$</td>
<td>62</td>
</tr>
<tr>
<td>$C_7^{ax} \rightarrow \alpha_L$</td>
<td>60</td>
</tr>
</tbody>
</table>
Transition-based RRT (T-RRT) : Experiments

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

- **Alanine dipeptide** : Transition $\alpha_R \leftrightarrow P_{II}$

<table>
<thead>
<tr>
<th></th>
<th>I (%)</th>
<th>II (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_R \rightarrow P_{II}$</td>
<td>34</td>
<td>66</td>
</tr>
<tr>
<td>$P_{II} \rightarrow \alpha_R$</td>
<td>64</td>
<td>36</td>
</tr>
</tbody>
</table>
Transition-based RRT (T-RRT) : Experiments

Alanine dipeptide : Transition $\alpha_R \leftrightarrow C_7^{ax}$

<table>
<thead>
<tr>
<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>
</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>
</tr>
</tbody>
</table>
Multiple Tree Variant: Multi-T-RRT

Algorithm: Multi-T-RRT

input: the conformational space $\mathcal{C}$
the energy function $E : \mathcal{C} \to \mathbb{R}$
the initial conformations $q_{init}^k$, $k = 1..n$
the extension step-size $\delta$

output: the tree $\mathcal{T}$

for $k = 1..n$ do
    $\mathcal{T}_k \leftarrow$ initTree($q_{init}^k$)

while not stopCondition($\{\mathcal{T}_k | k = 1..n\}$) do
    $\mathcal{T}' \leftarrow$ chooseNextTreeToExpand()
    $q_{rand} \leftarrow$ sampleRandomConfiguration($\mathcal{C}$)
    $q'_{near} \leftarrow$ findNearestNeighbor($\mathcal{T}'$, $q_{rand}$)
    if refinementControl($\mathcal{T}'$, $q'_{near}$, $q_{rand}$) then
        $q_{new} \leftarrow$ extend($q'_{near}$, $q_{rand}$, $\delta$)
        if $q_{new} \neq$ null and
            transitionTest($\mathcal{T}'$, $E(q'_{near})$, $E(q_{new})$) then
                addNewNode($\mathcal{T}'$, $q_{new}$)
                addNewEdge($\mathcal{T}'$, $q'_{near}$, $q_{new}$)
                ($\mathcal{T}''$, $q''_{near}$) $\leftarrow$ findNearestTree($q_{new}$)
                if distance($q_{new}$, $q''_{near}$) $\leq$ $\delta$ then
                    $\mathcal{T} \leftarrow$ merge($\mathcal{T}'$, $q_{new}$, $\mathcal{T}''$, $q''_{near}$)
                    $n \leftarrow n - 1$
Multi-T-RRT: Experiments

- Experiments on 2D Cost Maps
Multi-T-RRT: Preliminary Results on Alanine Dipeptide

- **Energy landscape** projection on \((\Phi, \Psi)\) coordinates

- AMBER parm96 force-field with implicit solvent model (GB)

- Coordinates and relative energies of minima:

<table>
<thead>
<tr>
<th></th>
<th>(C_5)</th>
<th>(P_{II})</th>
<th>(\alpha_R)</th>
<th>(\alpha_P)</th>
<th>(C_{7}^{\alpha x})</th>
<th>(\alpha_L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\phi) (°)</td>
<td>-145</td>
<td>-65</td>
<td>-62</td>
<td>-143</td>
<td>45</td>
<td>43</td>
</tr>
<tr>
<td>(\psi) (°)</td>
<td>160</td>
<td>148</td>
<td>-49</td>
<td>-70</td>
<td>-116</td>
<td>61</td>
</tr>
<tr>
<td>E (kcal/mol)</td>
<td>0</td>
<td>0.3</td>
<td>1</td>
<td>1.6</td>
<td>3.3</td>
<td>3.9</td>
</tr>
</tbody>
</table>
Multi-T-RRT: Preliminary Results on Alanine Dipeptide

Construction of a multiple trees rooted at the minima
Multi-T-RRT: Preliminary Results on Alanine Dipeptide

100 paths connecting all minima

- $T_{rate} = 0.1$
- Total CPU time $\equiv 1$ min
  ($< 1$ sec. for each solution)

- $T_{rate} = 0.01$
- Total CPU time $\equiv 8$ min
  (~ 5 sec. for each solution)
A Two-Stage Approach: Geometry + Energy

LAAS-CNRS, [Cortés et al., 2005]

Filtering
- Exploration with geometric constraints
  - Path planning algorithms
- Energy-based exploration
  - Classical molecular modeling methods
  - New methods?

Aim: Global and continuous exploration of the conformational space
Protein Loop Mobility

Xylanase
from *Thermobacillus xylanilyticus*

Flexible loop: 69 d.o.f.

[Cortés et al., Bioinformatics, 2005]
Protein Loop Mobility

[Cortés et al., Bioinformatics, 2005]

Exploration with RRT
- 5000 nodes
- 15 min.

→ Prediction for directed mutagenesis

Mutated Xylanase
(deletion 111, 121)

Xylanase
(from Thermobacillus xylanilyticus)
Protein Loop Mobility

[Paes et al., CSBJ, 2012]

Understanding the enzymatic mechanism
THOUSANDS of DOFs !!!

→ Too complex … even for sampling-based algorithms
Multi-Scale Multi-Physics Approach

[RRT]

For exploring the conformational space efficiently

To bias the search of the RRT towards relevant parts of the conformational space

To reduce the dimensionality of the search space without loosing all-atom details

[Al-Bluwi et al. BMC Struct Biol, 2013]
Mechanistic Model: Decomposition into Tripeptides

Cut the protein chain into fragments of 3 amino acid residues

[Al-Bluwi et al. BMC Struct Biol, 2013]
Mechanistic Model: Decomposition into Tripeptides

Cut the protein chain into fragments of 3 amino acid residues

Three pairs of $\Phi$, $\Psi$ angles

$6R$ mechanism

[Al-Bluwi et al. BMC Struct Biol, 2013]
Mechanistic Model: Decomposition into Tripeptides

[Al-Bluwi et al. BMC Struct Biol, 2013]

Conformational sampling using the tripeptide-based model

1. Sample/perturb the pose of the base-frame of tripeptides (oriented particles)
   ➔ Coarse-grained model

2. Solve IK for tripeptides
   Semi-analytical solver (adapted from [Renaud, 2000])
   ➔ All-atom model
Elastic Network Normal Model Analysis

[Al-Bluwi et al. BMC Struct Biol, 2013]

Tripeptide-based Elastic Network Model

- **Topological** force field
  - Springs between particles

\[ V(x) = \frac{1}{2} k (x - x_0)^2 \]
Elastic Network Normal Model Analysis

[Al-Bluwi et al. BMC Struct Biol, 2013]

Tripeptide-based Elastic Network Model

• Diagonalization of the Hessian matrix:
  • Eigenvectors → Vibration directions
  • Eigenvalues → Vibration frequencies

• Low-frequency modes ➞ collective motions

\[
H = \begin{pmatrix}
\frac{\partial^2 V}{\partial r_i \partial r_i} & \frac{\partial^2 V}{\partial r_i \partial r_2} & \cdots & \frac{\partial^2 V}{\partial r_i \partial r_{3N}} \\
\frac{\partial^2 V}{\partial r_1 \partial r_2} & \frac{\partial^2 V}{\partial r_2 \partial r_2} & \cdots & \frac{\partial^2 V}{\partial r_2 \partial r_{3N}} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 V}{\partial r_{3N} \partial r_1} & \frac{\partial^2 V}{\partial r_{3N} \partial r_2} & \cdots & \frac{\partial^2 V}{\partial r_{3N} \partial r_{3N}}
\end{pmatrix}
\]
Elastic Network Normal Model Analysis

[Al-Bluwi et al. BMC Struct Biol, 2013]

Tripeptide-based Elastic Network Model

• Advantages:
  • Reduces the size of the hessian matrix by a factor of 3 (compared to Cα-ENM)
  • Preserves the full flexibility of the protein (compared to rigid-block approaches)
Overall Algorithm

Algorithm 1: COMPUTE_PATHWAY

```
input : Initial conformation \( q_{init} \), final conformation \( q_{goal} \) and minimum distance to target \( d_{target} \)
output : The transition pathway \( p \)
begin
    \( q_{root} \leftarrow q_{init} \);
    while \( \text{RMSD}(q_{root}, q_{goal}) > d_{target} \) do
        \( m \leftarrow \text{COMPUTE.NORMALMODES}(q_{root}); \)
        \( t \leftarrow \text{BUILD.RRT}(m, q_{root}, q_{goal}); \)
        \( q_{close} \leftarrow \text{CLOSEST.TOTARGET}(t, q_{goal}); \)
        \( q_{root} \leftarrow \text{MINIMIZE}(q_{close}); \)
        \( p \leftarrow \text{CONCATENATE}(p, q_{root}); \)
    end
```

[Al-Bluwi et al. BMC Struct Biol, 2013]
The RRT algorithm

[Al-Bluwi et al. BMC Struct Biol, 2013]

Algorithm: BUILD_RRT

input: Initial conformation $q_{root}$, final conformation $q_{goal}$
output: The tree $t$

begin

$t \leftarrow \text{INITTREE}(q_{root})$;

while not STOPCONDITION($t$, $q_{goal}$) do

$q_{rand} \leftarrow \text{SAMPLE}(t)$;
$q_{near} \leftarrow \text{BESTNEIGHBOR}(t$, $q_{rand})$;
$q_{new} \leftarrow \text{EXPANDTREE}(q_{near}, q_{rand})$;
if ISVALID($q_{new}$) then

$\text{ADDNEWNODE}(t$, $q_{new})$;
$\text{ADDDNEWEDGE}(t$, $q_{near}, q_{new})$;

end
Sampling Conformations of the Coarse-Grained Model

[Al-Bluwi et al. BMC Struct Biol, 2013]
Nearest Neighbor Search

[Al-Bluwi et al. BMC Struct Biol, 2013]

- Weighted distance metric to accelerate search process:

\[d(q_i, q_{rand}) = \frac{\text{RMSD}(q_i, q_{rand}) \text{RMSD}(q_i, q_{goal})}{\text{RMSD}(q_{init}, q_{goal})}\]

- A simple brute-force algorithm is used.
- More efficient NN search algorithms could decrease computing time.
Generating a New Conformation

[Al-Bluwi et al. BMC Struct Biol, 2013]

- All particle positions in $q_{\text{near}}$ are linearly interpolated towards $q_{\text{rand}}$ with a predefined distance $k$.

- The all-atom model corresponding to $q_{\text{new}}$ is generated using an iterative process.
Results: Validating the Tripeptide-based ENM

[Al-Bluwi et al. BMC Struct Biol, 2013]

• Topological features are preserved
Results: Validating the Tripeptide-based ENM

[Al-Bluwi et al. BMC Struct Biol, 2013]

- Comparison of overlap values [Marques 95, Tama 01]

<table>
<thead>
<tr>
<th>Protein</th>
<th>Cα-ENM</th>
<th>P-ENM_{best}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td>Che Y</td>
<td>0.32</td>
<td>0.34</td>
</tr>
<tr>
<td>LAO</td>
<td>0.84</td>
<td>0.40</td>
</tr>
<tr>
<td>Triglyceride</td>
<td>0.30</td>
<td>0.17</td>
</tr>
<tr>
<td>Thymidulate Synthase</td>
<td>0.56</td>
<td>0.40</td>
</tr>
<tr>
<td>Maltodextrine</td>
<td>0.86</td>
<td>0.77</td>
</tr>
<tr>
<td>Enolase</td>
<td>0.33</td>
<td>0.30</td>
</tr>
<tr>
<td>Diphtheria Toxin</td>
<td>0.58</td>
<td>0.37</td>
</tr>
</tbody>
</table>
Results: Finding the Optimal Cut-Off Distance

[Al-Bluwi et al. BMC Struct Biol, 2013]
## Results: Modeling Conformational Transitions

[Al-Bluwi et al. BMC Struct Biol, 2013]

<table>
<thead>
<tr>
<th>Protein</th>
<th>Residues</th>
<th>PDB ID&lt;sub&gt;init&lt;/sub&gt;</th>
<th>PDB ID&lt;sub&gt;goal&lt;/sub&gt;</th>
<th>ParRMSD&lt;sub&gt;init&lt;/sub&gt;</th>
<th>Cα-RMSD&lt;sub&gt;init&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADK</td>
<td>214</td>
<td>4ake</td>
<td>1ake</td>
<td>6.52</td>
<td>6.51</td>
</tr>
<tr>
<td>LAO</td>
<td>238</td>
<td>2lao</td>
<td>1laf</td>
<td>3.77</td>
<td>3.73</td>
</tr>
<tr>
<td>DAP</td>
<td>320</td>
<td>1dap</td>
<td>3dap</td>
<td>3.81</td>
<td>3.78</td>
</tr>
<tr>
<td>NS3</td>
<td>436</td>
<td>3kqk</td>
<td>3kql</td>
<td>2.75</td>
<td>2.75</td>
</tr>
<tr>
<td>DDT</td>
<td>535</td>
<td>1ddt</td>
<td>1mdt</td>
<td>10.93</td>
<td>10.96</td>
</tr>
<tr>
<td>GroEL</td>
<td>547</td>
<td>1aon</td>
<td>1oel</td>
<td>10.38</td>
<td>10.49</td>
</tr>
<tr>
<td>ATP</td>
<td>573</td>
<td>1m8p</td>
<td>1i2d</td>
<td>3.79</td>
<td>3.78</td>
</tr>
<tr>
<td>BKA</td>
<td>691</td>
<td>1cb6</td>
<td>1bka</td>
<td>4.73</td>
<td>4.75</td>
</tr>
<tr>
<td>UKL</td>
<td>876</td>
<td>1ukl</td>
<td>1qgk</td>
<td>6.16</td>
<td>6.17</td>
</tr>
<tr>
<td>HKC</td>
<td>917</td>
<td>1hkc</td>
<td>1hkb</td>
<td>2.98</td>
<td>3.00</td>
</tr>
</tbody>
</table>
## Results: Modeling Conformational Transitions

[Al-Bluwi et al. BMC Struct Biol, 2013]

<table>
<thead>
<tr>
<th>Protein</th>
<th>$C_\alpha$-RMSD$_{init}$</th>
<th>$C_\alpha$-RMSD$_{end}$</th>
<th>Iterations</th>
<th>Time$_{RRT}$</th>
<th>Time$_{total}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADK</td>
<td>6.51</td>
<td>1.56</td>
<td>31</td>
<td>1.82</td>
<td>2.00</td>
</tr>
<tr>
<td>LAO</td>
<td>3.73</td>
<td>1.32</td>
<td>20</td>
<td>1.52</td>
<td>1.65</td>
</tr>
<tr>
<td>DAP</td>
<td>3.78</td>
<td>1.31</td>
<td>16</td>
<td>1.78</td>
<td>1.92</td>
</tr>
<tr>
<td>NS3</td>
<td>2.75</td>
<td>1.29</td>
<td>14</td>
<td>2.82</td>
<td>3.00</td>
</tr>
<tr>
<td>DDT</td>
<td>10.96</td>
<td>2.88</td>
<td>272</td>
<td>81.54</td>
<td>86.4</td>
</tr>
<tr>
<td>GroEL</td>
<td>10.49</td>
<td>2.79</td>
<td>142</td>
<td>40.21</td>
<td>42.17</td>
</tr>
<tr>
<td>ATP</td>
<td>3.78</td>
<td>1.45</td>
<td>30</td>
<td>13.46</td>
<td>14.16</td>
</tr>
<tr>
<td>BKA</td>
<td>4.75</td>
<td>1.96</td>
<td>74</td>
<td>29.56</td>
<td>31.09</td>
</tr>
<tr>
<td>UKL</td>
<td>6.17</td>
<td>2.02</td>
<td>80</td>
<td>80.61</td>
<td>82.62</td>
</tr>
<tr>
<td>HKC</td>
<td>3.00</td>
<td>1.64</td>
<td>38</td>
<td>37.91</td>
<td>39.63</td>
</tr>
</tbody>
</table>
Results: Modeling Conformational Transitions

[Al-Bluwi et al. BMC Struct Biol, 2013]

init - goal

end - goal

GroEL

DDT
Results: Modeling Conformational Transitions

[Al-Bluwi et al. BMC Struct Biol, 2013]
Results: Scalability of the Method

[Al-Bluwi et al. BMC Struct Biol, 2013]
## Results: Computational Cost Decomposition

[Al-Bluwi et al. BMC Struct Biol, 2013]

<table>
<thead>
<tr>
<th>Protein</th>
<th>NN</th>
<th>CC</th>
<th>IK</th>
<th>RS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADK</td>
<td>57.2%</td>
<td>14.1%</td>
<td>15.0%</td>
<td>6.3%</td>
</tr>
<tr>
<td>LAO</td>
<td>51.3%</td>
<td>20.9%</td>
<td>17.0%</td>
<td>5.4%</td>
</tr>
<tr>
<td>DAP</td>
<td>50.5%</td>
<td>20.6%</td>
<td>11.0%</td>
<td>12.3%</td>
</tr>
<tr>
<td>NS3</td>
<td>67.9%</td>
<td>13.4%</td>
<td>6.6%</td>
<td>8.9%</td>
</tr>
<tr>
<td>DDT</td>
<td>64.3%</td>
<td>17.1%</td>
<td>6.9%</td>
<td>9.0%</td>
</tr>
<tr>
<td>GroEL</td>
<td>60.4%</td>
<td>17.6%</td>
<td>8.9%</td>
<td>9.8%</td>
</tr>
<tr>
<td>ATP</td>
<td>57.3%</td>
<td>20.9%</td>
<td>6.8%</td>
<td>11.9%</td>
</tr>
<tr>
<td>BKA</td>
<td>55.1%</td>
<td>16.8%</td>
<td>6.1%</td>
<td>19.3%</td>
</tr>
<tr>
<td>UKL</td>
<td>62.9%</td>
<td>15.5%</td>
<td>4.1%</td>
<td>15.5%</td>
</tr>
<tr>
<td>HKC</td>
<td>68.9%</td>
<td>5.8%</td>
<td>3.3%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Average</td>
<td>59.58%</td>
<td>16.27%</td>
<td>8.57%</td>
<td>11.66%</td>
</tr>
</tbody>
</table>

**NN**: Nearest Neighbor Search  
**CC**: Collision Checking  
**IK**: Inverse Kinematics  
**RS**: Random Sampling
Results: A Closer Look to ADK

The structure of ADK is divided into three main domains: **LID**, **CORE** and **NMPbind**.

The LID and NMPbind domains undergo clear conformational changes, whereas the CORE domain remains almost unchanged [Maragakis 05, Muller 96].
Results: Residue Displacements in ADK

• Residues around 20-60 and 130-160, approximately correspond to the NMPbind and LID domains.
• Two-step process: NMPbind moves less clearly than LID at the beginning and then moves faster towards the end [Maragakis 05]
Results: Intermediate conformations of ADK

[Al-Bluwi et al. BMC Struct Biol, 2013]

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>RMSD</th>
<th>Iteration</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1DVR (A)</td>
<td>1.48</td>
<td>2</td>
<td>9%</td>
</tr>
<tr>
<td>2RH5 (A)</td>
<td>1.80</td>
<td>1</td>
<td>4%</td>
</tr>
<tr>
<td>2RH5 (B)</td>
<td>1.91</td>
<td>3</td>
<td>15%</td>
</tr>
<tr>
<td>1E4Y (A)</td>
<td>2.20</td>
<td>27</td>
<td>94%</td>
</tr>
</tbody>
</table>

- Comparison to several available structures for ADK
- Four structures were found to be very close to intermediate conformations generated by our method.
- Results are consistent with previous results [Feng 2009]
**Biological interest**
Analyzing the ligand access/exit to the protein active site is important for **understanding the molecular interaction**.

**Difficulty**
Computing such pathways with energy-based molecular modeling methods is too **computationally expensive**.
Geometric formulation
Assembly/disassembly problem for articulated objects.

Path planning algorithms
can be used to compute pathways.

Challenging problem
• High number of DOF.
• Very constrained space.
Mechanistic Problem Formulation

Compute a disassembly path for two objects with articulated parts

[Cortés et al., IEEE-TRO, 2008]
Solution Method: RRT-like Path Planning Algorithms

Some reasons for using RRT
- **Sampling-based** method: the problem is high-dimensional.
- **Single-query** method: the initial conformation (configuration) is given.
- **Incremental-search** method: the search space is a “pipe-like” manifold.

[Cortés et al., IEEE-TRO, 2008]
Solution Method: RRT-like Path Planning Algorithms

[Cortés et al., IEEE-TRO, 2008]

Difficulties

- **Different nature of DOFs**: ligand motion vs. side-chain motions
- **Are all DOFs necessary** for the disassembly?
Two sets of parameters

- **Active parameters**: $q^{\text{act}}$
  - are directly handled by the planner
- **Passive parameters**: $q^{\text{pas}}$
  - only treated when required for expansion
Decoupling Part Motions: The ML-RRT Algorithm

[ Cortés et al., IEEE-TRO, 2008 ]

Two sets of parameters
- **Active parameters**: $q^{\text{act}}$
  - are directly handled by the planner
- **Passive parameters**: $q^{\text{pas}}$
  - only treated when required for expansion

Protein-ligand disassembly
- $q^{\text{act}} = q^{\text{lig}}$
  - ligand position/orientation + flexibility
- $q^{\text{pas}} = q^{\text{prot}}$
  - protein side-chains flexibility
Decoupling Part Motions: The ML-RRT Algorithm

**Main idea:**
Treat active/passive parameters alternately

---

Active parameters expansion

---

**Algorithm :** Construct\_ML-\_RRT

\[
\begin{align*}
\text{input} & : \text{the search-space } C; \\
& \quad \text{the root } q_{\text{init}} \text{ and the goal } q_{\text{goal}}; \\
& \quad \text{the partition } \{L_{\text{act}}, L_{\text{pas}}\}; \\
\text{output} & : \text{the tree } \tau; \\
\text{begin} \\
\tau & \leftarrow \text{InitTree}(q_{\text{init}}); \\
\text{while not StopCondition}(\tau, q_{\text{goal}}) \text{ do} \\
& \quad q_{\text{act}} \leftarrow \text{SampleConf}(C, L_{\text{act}}); \\
& \quad q_{\text{near}} \leftarrow \text{BestNeighbor}(\tau, q_{\text{act}}, L_{\text{act}}); \\
& \quad (q_{\text{new}}, L'_{\text{pas}}) \leftarrow \text{Expand}(q_{\text{near}}, q_{\text{act}}); \\
& \quad \text{if not TooSimilar}(q_{\text{near}}, q_{\text{new}}) \text{ then} \\
& \quad \quad \text{AddNewNode}(\tau, q_{\text{new}}); \\
& \quad \quad \text{AddNewEdge}(\tau, q_{\text{near}}, q_{\text{new}}); \\
& \quad \quad q_{\text{near}} \leftarrow q_{\text{new}}; \\
& \quad \text{while } L'_{\text{pas}} \neq \emptyset \text{ do} \\
& \quad \quad q_{\text{pass}} \leftarrow \text{PerturbConf}(C, q_{\text{near}}, L'_{\text{pas}}); \\
& \quad \quad (q_{\text{new}}, L_{\text{col}}') \leftarrow \text{Expand}(q_{\text{near}}, q_{\text{pass}}); \\
& \quad \quad \text{if not TooSimilar}(q_{\text{near}}, q_{\text{new}}) \text{ then} \\
& \quad \quad \quad \text{AddNewNode}(\tau, q_{\text{new}}); \\
& \quad \quad \quad \text{AddNewEdge}(\tau, q_{\text{near}}, q_{\text{new}}); \\
& \quad \quad \quad q_{\text{near}} \leftarrow q_{\text{new}}; \\
& \quad \quad L_{\text{pas}} \leftarrow L_{\text{pas}} \setminus L_{\text{col}}; \\
& \text{end}
\end{align*}
\]
Decoupling Part Motions: The ML-RRT Algorithm

Main idea:
Treat active/passive parameters alternately

Passive parameters expansion (1)

Algorithm : Construct_ML-RRT

input : the search-space \( C \);
         the root \( q_{\text{init}} \) and the goal \( q_{\text{goal}} \);
         the partition \( \{ L_{\text{act}}, L_{\text{pas}} \} \);
output : the tree \( \tau \);
begin
  \( \tau \leftarrow \text{InitTree}(q_{\text{init}}) \);
  while not \text{StopCondition}(\tau, q_{\text{goal}}) \) do
    \( q_{\text{act}} \leftarrow \text{SampleConf}(C, L_{\text{act}}) \);
    \( q_{\text{near}} \leftarrow \text{BestNeighbor}(\tau, q_{\text{act}}^{\text{act}}, L_{\text{act}}) \);
    \( (q_{\text{new}}, L_{\text{pas}}^{\text{col}}) \leftarrow \text{Expand}(q_{\text{near}}, q_{\text{act}}^{\text{act}}) \);
    if not \text{TooSimilar}(q_{\text{near}}, q_{\text{new}}) then
      \( \text{AddNewNode}(\tau, q_{\text{new}}) \);
      \( \text{AddNewEdge}(\tau, q_{\text{near}}, q_{\text{new}}) \);
      \( q_{\text{near}} \leftarrow q_{\text{new}} \);
    \end{while}
    \( L_{\text{pas}}^{\text{col}} \leftarrow L_{\text{pas}}^{\text{col}} \setminus L_{\text{pas}}^{\text{col}} \);
  \end{while}
end
Decoupling Part Motions: The ML-RRT Algorithm

Main idea:
Treat active/passive parameters alternately

Passive parameters expansion (2)
## Experimental Performance Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>T nodes</th>
<th>N nodes</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRT</td>
<td></td>
<td>5047</td>
<td>751.7</td>
</tr>
<tr>
<td>ML-RRT</td>
<td></td>
<td>856</td>
<td>8.0</td>
</tr>
<tr>
<td>ML-RRT</td>
<td></td>
<td></td>
<td>13.5</td>
</tr>
<tr>
<td>RRT</td>
<td></td>
<td></td>
<td>→∞</td>
</tr>
<tr>
<td>ML-RRT</td>
<td></td>
<td></td>
<td>→∞</td>
</tr>
</tbody>
</table>

### Diagram:

- Two diagrams showing the performance analysis of RRT and ML-RRT methods.
Experimental Performance Analysis

RRT projection on the position parameters of the mobile object

RRT
\[ N_{\text{nodes}} = 1000 \]

ML-RRT
\[ N_{\text{nodes}} \approx 100 \]
A Difficult Molecular Disassembly Example

• Amylosucrase: narrow deep active site
• Protein size: 628 residues
• 1137 (potential) DOF
  • Ligand : 11 DOF
  • Protein side-chains: 1126 DOF
• 9 side-chains move in the solution path
  • 1 known to be biologically important

• ML-RRT computing $T < 2$ min.
Extension to Flexible Backbone

ML-RRT with passiveness levels

- Ligand ➔ Active
- Side-chains ➔ Passive Level 1
- Flexible backbone (loops) ➔ Passive Level 2
- Rigid backbone ➔ Static

[Cortés et al., PCCP 2010]
Extension to Flexible Backbone

[Cutés et al., PCCP 2010]

Partially flexible protein model
Extension to Flexible Backbone

ML-RRT exploration process

[extension to flexible backbone with ML-RRT exploration process]

Case 1: \( P_{\text{pop}}(sc_1, G, e_{L_{11}}, e_{L_{12}}) \)
Case 2: \( P_{\text{pop}}(sc_2, l_{11}, G, e_{L_{11}}, e_{L_{12}}) \)
Case 3: \( P_{\text{pop}}(sc_3, l_{12}, G, G_{i1}, e_{L_{11}}, e_{L_{12}}) \)
Test Systems

- **Amylosucrase**: Multiple loop motions
- **Lactose Permease (LacY)**: Domain motions
- **β2-Adrenergic Receptor**: TM helices and loops motions
Amylosucrase: Multiple loop motions

- Protein model: Four flexible loops (3, 4, 7 and 8)
- Ligand: disaccharide Allyl-α-D-Glcp-(1→4)-a-DGlcNac
- Flexibility: > 1500 DOF
  - Loops 156, Side-chain 1474, Ligand 22
- CPU Time: ~1 h on a single processor
Amylosucrase: Multiple loop motions

Ligand exit pathway analysis

- Catalytic residues: Glu328, Asp393 [Albenne02, Champion09]
- Substrate stabilization: His187, Phe250, His392, Arg284 [Albenne02, 04]
- Salt-bridge: Asp144, Arg509 [Albenne02]
- Other important interactions: Arg446, Arg226, Asp444

Consistent with experimental results
Lactose Permease: Domain motions

- Protein model: Two main domains connected by a linker
- Substrate: TDG
- Flexibility: ~800 DOF
  - Domains 12, Linker 75, Side-chain 678, Ligand 10
- CPU Time: <1 h on a single processor
Lactose Permease: Domain motions

### Outward-open model analysis

- Rotation between two domains: $20^\circ = 1/3$ values of DEER [Smirnova07]
- Residue-pair distance variation: $\sim 1/3$ values of DEER [Smirnova07]
- Ile40-Asn245 distance: $15 \text{ Å} = \text{minimal distance for activity determined by cross-linking}$ [Zhou08]

<table>
<thead>
<tr>
<th>Residue pair</th>
<th>Inward-open</th>
<th>Outward-open (experimental)</th>
<th>Outward-open (simulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>73-401</td>
<td>41 Å</td>
<td>27 Å</td>
<td>36.9(±1.0) Å</td>
</tr>
<tr>
<td>73-340</td>
<td>36 Å</td>
<td>21 Å</td>
<td>31.0(±1.3) Å</td>
</tr>
<tr>
<td>136-340</td>
<td>34 Å</td>
<td>17 Å</td>
<td>28.7(±1.4) Å</td>
</tr>
<tr>
<td>137-340</td>
<td>32 Å</td>
<td>16 Å</td>
<td>26.7(±1.4) Å</td>
</tr>
<tr>
<td>136-401</td>
<td>40 Å</td>
<td>24 Å</td>
<td>35.6(±1.3) Å</td>
</tr>
<tr>
<td>137-401</td>
<td>38 Å</td>
<td>22 Å</td>
<td>33.5(±1.4) Å</td>
</tr>
<tr>
<td>105-310</td>
<td>34 Å</td>
<td>41 Å</td>
<td>38.0(±1.6) Å</td>
</tr>
<tr>
<td>164-310</td>
<td>27 Å</td>
<td>43 Å</td>
<td>32.8(±1.4) Å</td>
</tr>
<tr>
<td>164-375</td>
<td>33 Å</td>
<td>49 Å</td>
<td>35.8(±1.2) Å</td>
</tr>
</tbody>
</table>
Lactose Permease: Domain motions

Ligand exit pathway analysis
- Hydrophobic interactions: Trp151, Tyr26, Phe27, Phe49, Phe261
- H-bonding interactions: Glu269, Lys358, Met23, His322, Asp237, ...

→ Consistent with SMD simulations [Jensen07]
β2-Adrenergic Receptor: Helices and loops motions

- Protein model: 7 mobile helices connected by flexible loops
- Ligand: Carazolol
- Flexibility: > 700 DOF
  - Helices 42, Loops 159, Side-chains 490, Ligand 12
- CPU Time: \(~ 2 \text{ min}\) on a single processor
Ligand exit pathway analysis

- **Two types of exit pathway**: (similar to RAMD [Wang09])
  - Left-hand: 31/60
  - Right-hand: 29/60
Ligand exit pathway analysis

- **Two types of exit pathway**: (similar to RAMD [Wang09])
  - Left-hand: 31/60
  - Right-hand: 29/60
- **Larger** conformational changes of ECL2 loop in **right-hand** paths
β2-Adrenergic Receptor: Helices and loops motions

Ligand exit pathway analysis

- Salt-bridge: Asp192, Lys305
- Ligand entry and stabilization: Phe193
- Polar interaction, stabilization: Asn312

*Consistent with RAMD simulations* [Wang09]
Molecular Disassembly: An Interesting Application

[Guieysse et al., ChemBioChem, 2008]
[Lafaquiére et al., ChemBioChem, 2009]

Enzymatic Enantioselectivity Prediction

- **Goal**
  - Separate and Intermediate for drug synthesis

- **Enzymatic reaction**

\[
\text{OR} + \text{CH}_3(\text{CH}_2)_3\text{OH} \rightarrow \text{Enzyme} \rightarrow \text{R}\text{' enantiomer} + \text{ROH} + \text{S enantiomer}
\]
Enzymatic Enantioselectivity Prediction

$E_{\text{int}}(R) \approx E_{\text{int}}(S)$

[Bulkholderia cepacia Lipase]

[Guieysse et al., Tetrahedron: Asymmetry, 2003]

[Guieysse et al., ChemBioChem, 2008]
[Lafaceire et al., ChemBioChem, 2009]
Enzymatic Enantioselectivity Prediction

Geometrically feasible pathway

[Ligand: (R)-ph(Br)Et]

[Ligand: (S)-ph(Br)Et]

[Guieysse et al., ChemBioChem, 2008]
[AlgoSB School, November 2013]
Enzymatic Enantioselectivity Prediction

Energy profile along paths

Pseudo-molecular dynamics

Two-stage approach

Computing time = 3 days

Computing time = 30 minutes
(including energy minimization)

→ Correlation of results :
similar paths and energy profiles
Enzymatic Enantioselectivity Prediction

[Guieysse et al., ChemBioChem, 2008]
[Lafaquière et al., ChemBioChem, 2009]

Correlation of results

\[ \text{Computing time ratio} \sim \text{enantioselectivity} \]

\[
\begin{array}{|c|c|c|}
\hline
\text{Ratio R/S} & \text{Br} & \text{Cl} & \text{F} \\
\hline
\text{Experimental enantioselectivity} & 63 & 17 & \sim 1 \\
\hline
\text{Computing time (sec.)} & 29 & 18 & 18, 17 \\
\hline
\end{array}
\]
## Enzymatic Enantioselectivity Prediction

[Guieysse et al., ChemBioChem, 2008]
[Lafraguère et al., ChemBioChem, 2009]

### Correlation of results

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Experimentally determined $E$ value (vi R/vi S)</th>
<th>Average computing time [s] (±s.d.)</th>
<th>In silico CPU (S)/CPU (R) ratio</th>
<th>Number of computing failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>57 (±7)$^a$</td>
<td>1.7 (±0.2)</td>
<td>24.0 (±1.3)</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>4.3 (±0.5)$^a$</td>
<td>3.0 (±0.7)</td>
<td>7.6 (±1.0)</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>59 (±2)$^a$</td>
<td>1.7 (±0.4)</td>
<td>104.9 (±5.5)</td>
<td>61</td>
</tr>
<tr>
<td>4</td>
<td>50 (±3)$^a$</td>
<td>1.0 (±0.4)</td>
<td>39.0 (±3.4)</td>
<td>40</td>
</tr>
</tbody>
</table>

→ Computing time ratio ~ enantioselectivity

Juan Cortés
Mining protein flexibility with robotics-inspired methods
AlgoSB School, November 2013
Tools for Protein Design

Identification of Mutagenesis Targets

[Guieysse et al., ChemBioChem, 2008]
[Lafaquière et al., ChemBioChem, 2009]
Identification of Mutagenesis Targets

Table 3. Specific hydrolytic activity and Enantioselectivity values determined for the wild-type BCL and its six double-mutants towards each enantiomer of the racemic substrate (R,S)-2-chloro ethyl 2-bromophenylacetate (1). (Error range for the double-mutants is within 10 to 20%).

<table>
<thead>
<tr>
<th>Variants</th>
<th>$v_{ir}^{[a]}$ [mU/mg]</th>
<th>$v_{is}^{[a]}$ [mU/mg]</th>
<th>$v_{ir}^{[b]}$ [mU/mg]</th>
<th>Enantiopreference</th>
<th>$E$-value$^{[b]}$</th>
<th>Conversion [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wild-type</td>
<td>0.354</td>
<td>0.028</td>
<td>0.38</td>
<td>$R$</td>
<td>13</td>
<td>6.5 (48 h)</td>
</tr>
<tr>
<td>L17M/V266M</td>
<td>2.81</td>
<td>0.017</td>
<td>2.83</td>
<td>$R$</td>
<td>166</td>
<td>9 (19 h)</td>
</tr>
<tr>
<td>L17S/L287A</td>
<td>1.83</td>
<td>0.081</td>
<td>1.91</td>
<td>$R$</td>
<td>22.5</td>
<td>15.6 (20 h)</td>
</tr>
<tr>
<td>L17S/L287I</td>
<td>5.95</td>
<td>0.033</td>
<td>5.98</td>
<td>$R$</td>
<td>178</td>
<td>15.5 (20 h)</td>
</tr>
<tr>
<td>L17S/L287W</td>
<td>0.55</td>
<td>0.01</td>
<td>0.56</td>
<td>$R$</td>
<td>55</td>
<td>6 (20 h)</td>
</tr>
</tbody>
</table>

[a] $v_{ir}$, $v_{is}$ : initial rates; $v_i = v_{ir} + v_{is}$. [b] $E$-value = $v_{ir}/v_{is}$.

→ 15-fold increased activity (hydrolysis)
→ 10-fold enhanced enantioselectivity
Available via a web server: moma.laas.fr

[Devaurs et al., NAR, 2013]
Robotics Algorithms for Molecular Modeling: Conclusions

• Path planning algorithms are a suitable complement to molecular modeling techniques
• New tools based on path planning algorithms can help biologist to understand important intracellular processes
• Many applications can be found in pharmacology and biotechnologies

• The complexity of problems requires developing efficient algorithms
• There are still many open problems to address (e.g. macromolecular docking)
Molecular Motion Algorithms

Juan Cortés
juan.cortes@laas.fr

www.laas.fr