

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



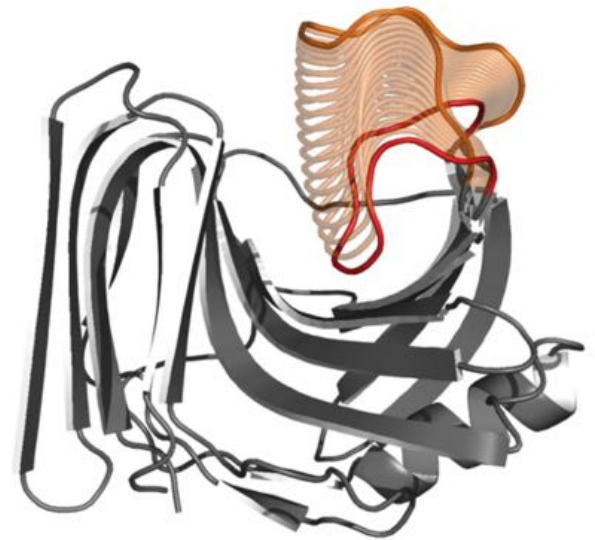
ECHORD++ DualArmWorker Experiment, LAAS-CNRS, Tecnalía, Airbus, Nov. 2017

Course Contents

- **Loop sampling**
- Path sampling on energy landscapes

NOT COVERED

- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design

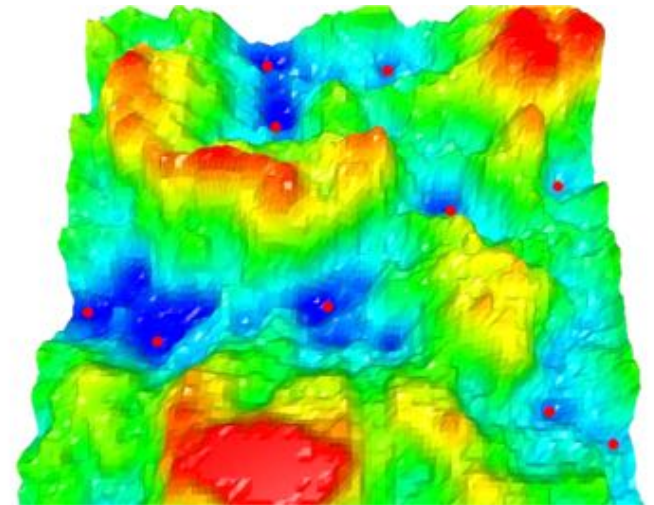


Course Contents

- Loop sampling
- Path sampling on energy landscapes

NOT COVERED

- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design

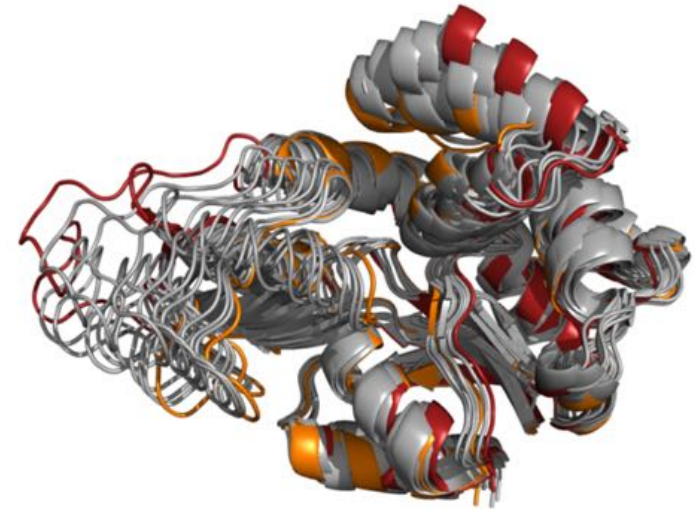


Course Contents

- 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

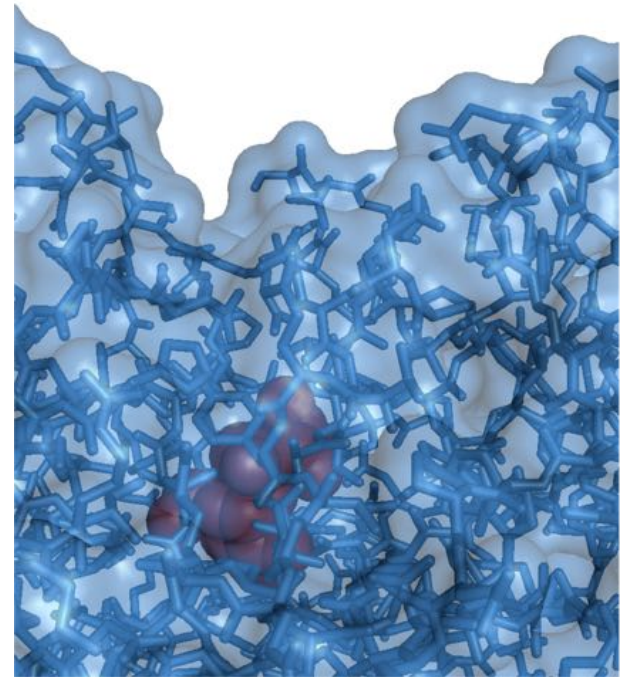
NOT COVERED

- Conformational transitions in proteins
- Protein-ligand access/exit pathways

[Cortés *et al.*, *PCCP* 2010]

[Devaurs *et al.*, *NAR* 2013]

Web server : moma.laas.fr



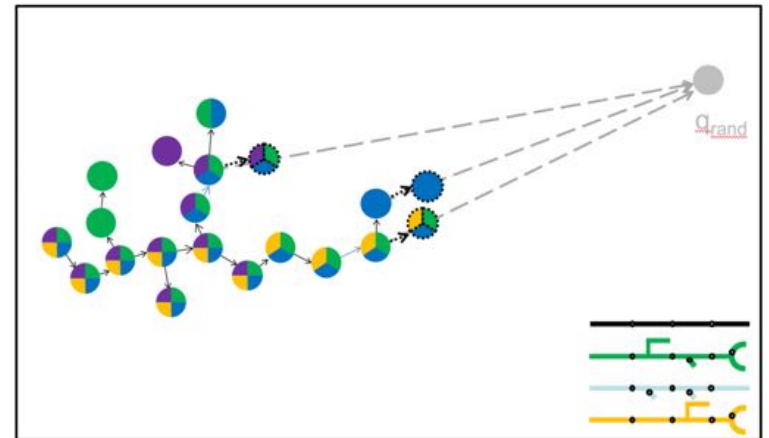
Course Contents

- 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]

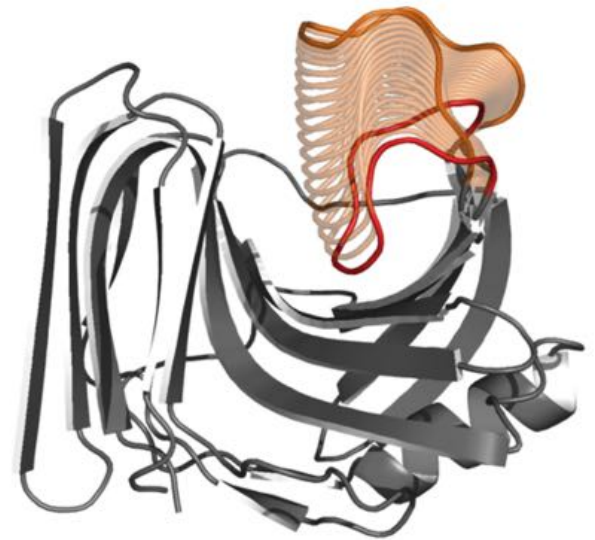


Course Contents

- Loop sampling
- Path sampling on energy landscapes

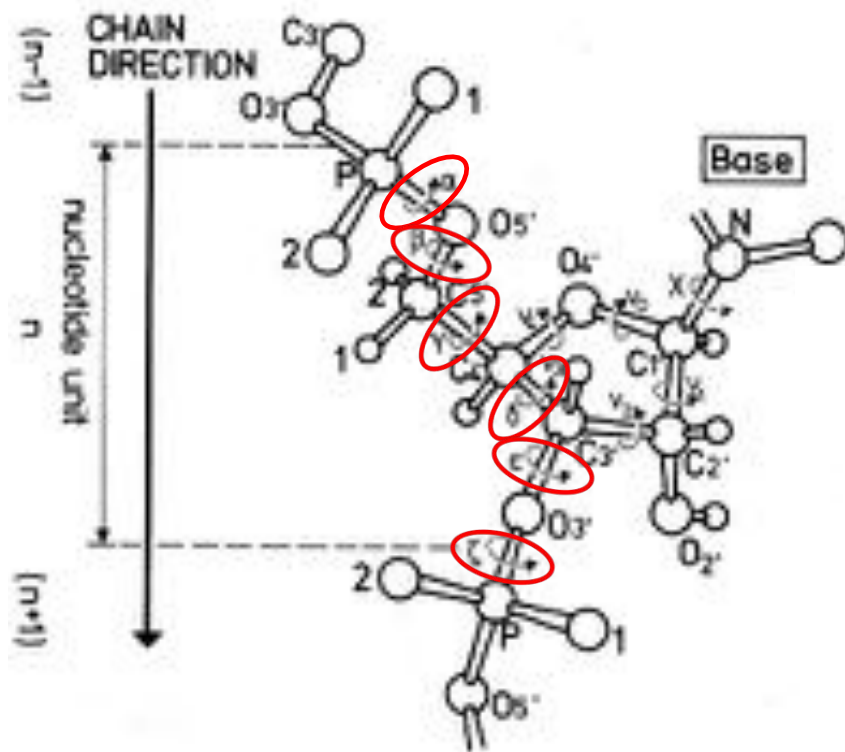
NOT COVERED

- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design

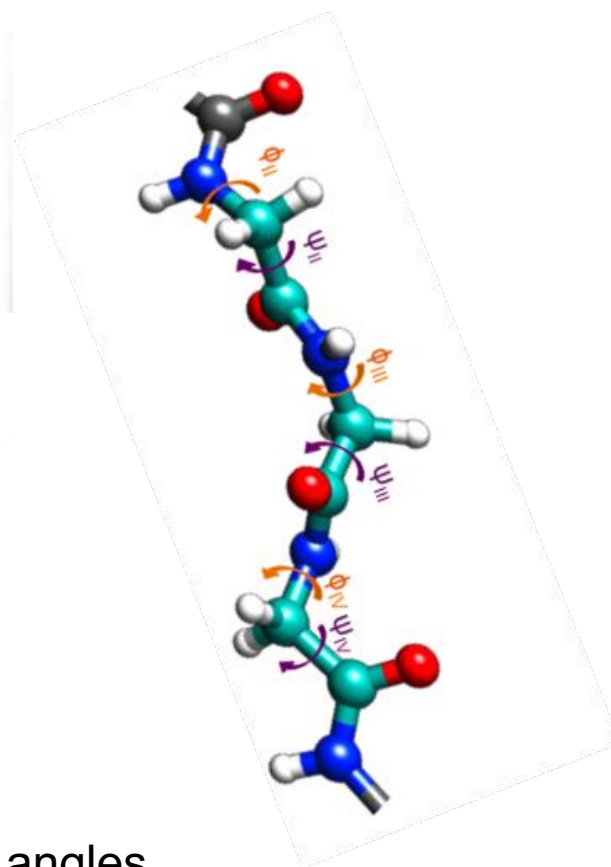


Internal coordinate models of proteins and RNA

Nucleotide

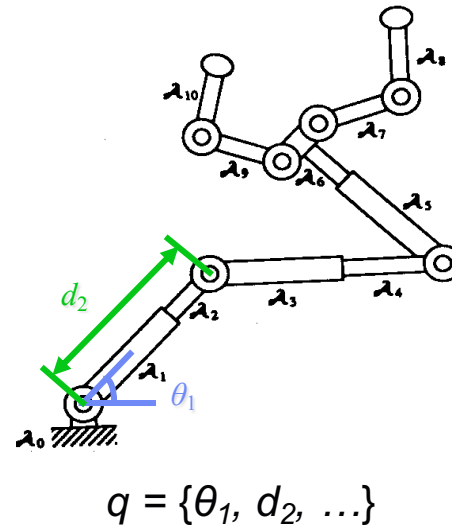
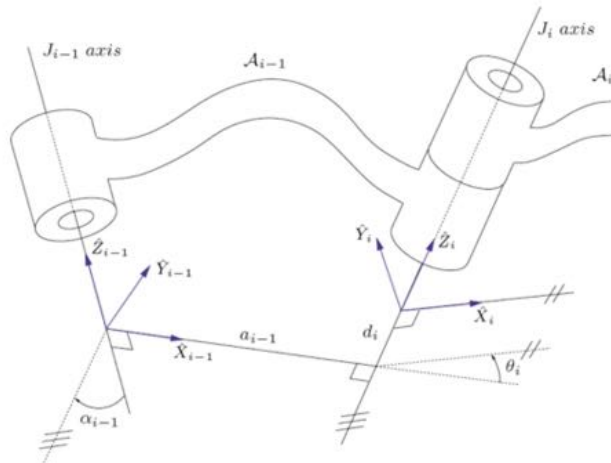


3 amino-acid fragment (*tripeptide*)



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 θ_i is variable

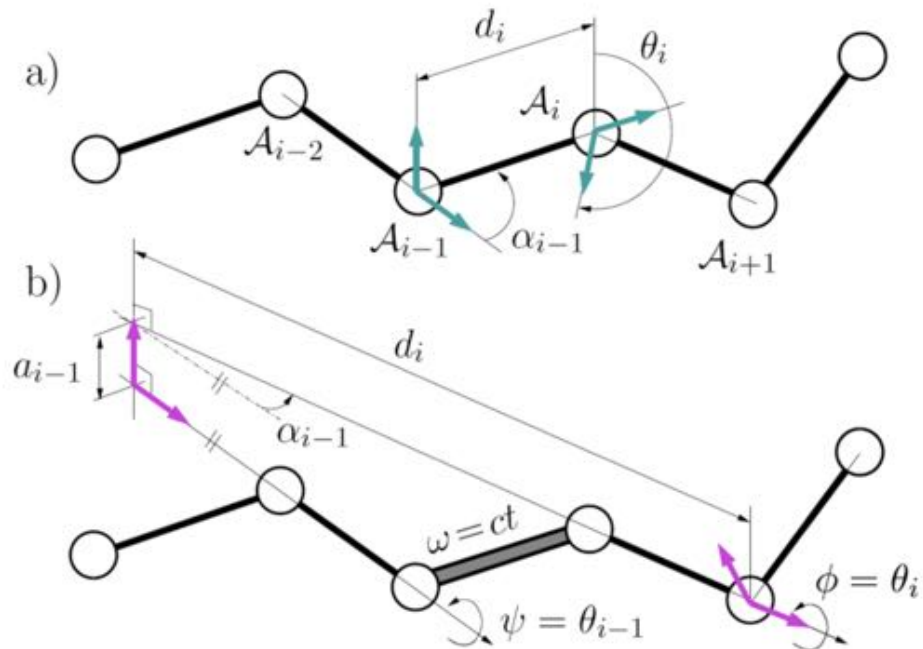
Homogeneous matrix transformation :

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

$${}^0T_n = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n$$

Modeling molecular chains as articulated mechanism

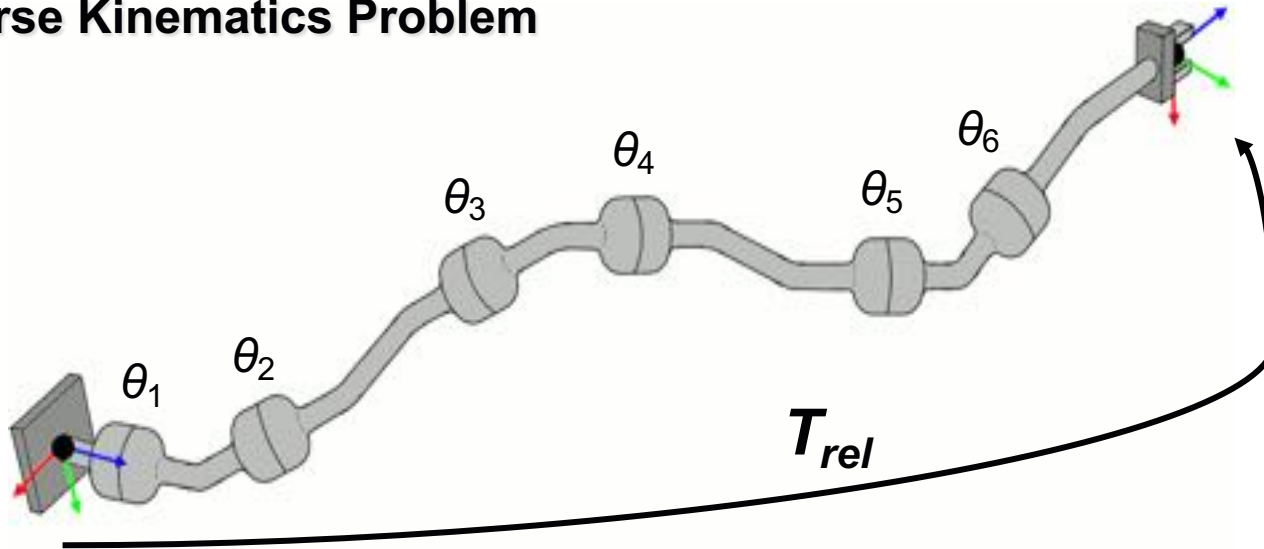
Denavit-Hartenberg parameters for a **molecular chain**



- 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 θ_i

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

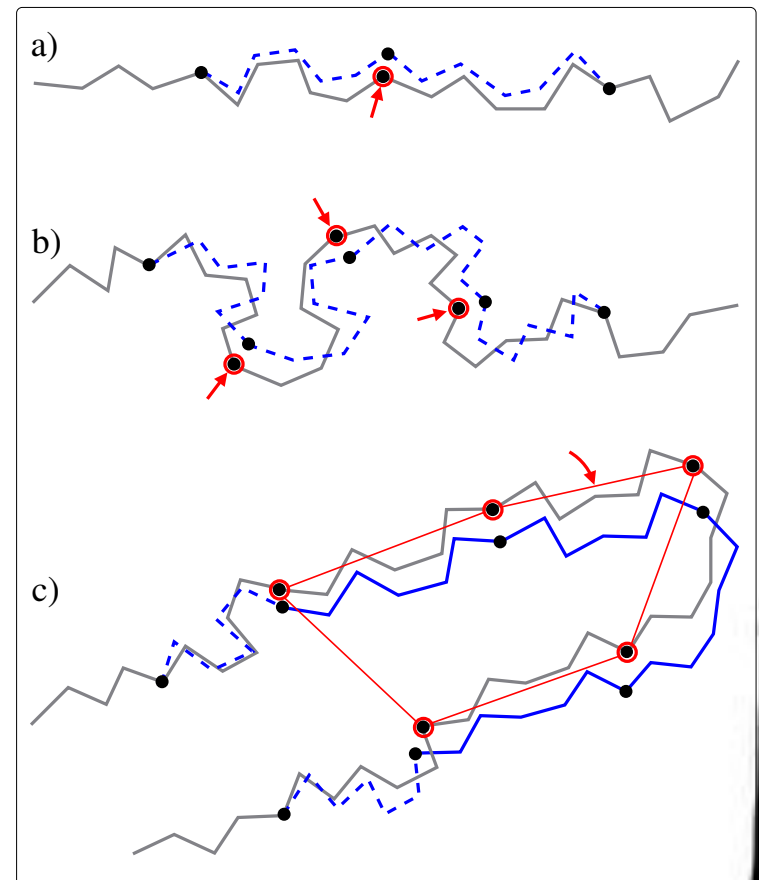
→ **Efficient closed-form solvers** [Renaud, 2000]

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]

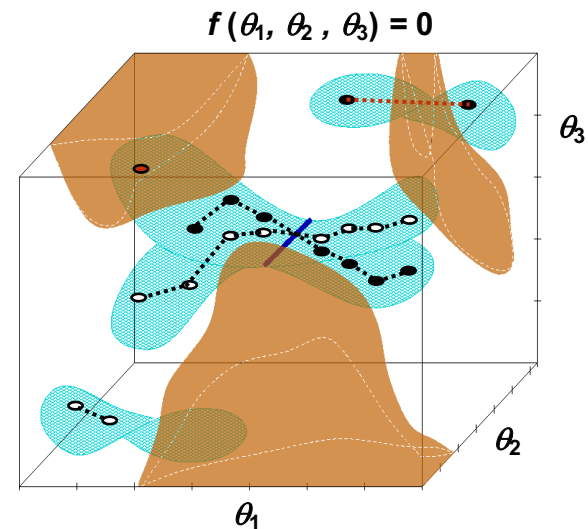


Sampling long loops

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]
 - Application to proteins: [Cortés et al, *J. Comput. Chem.*, 2004]

Sampling long loops

[Cortés et al., *J Comput Chem*, 2004]

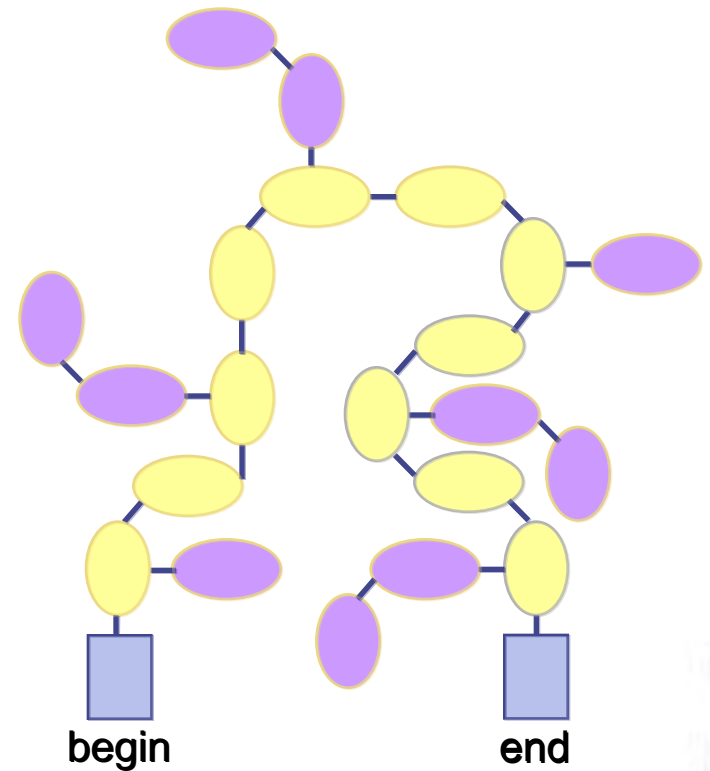
Random Loop Generator (RLG)

Main Chain (Backbone)

- Independent sub-chain
 - RLG algorithm + CollCheck
- Dependent sub-chain
 - General 6R IK + CollCheck
[Renaud, 2000]

Side Chains

- Random Sampling + CollCheck



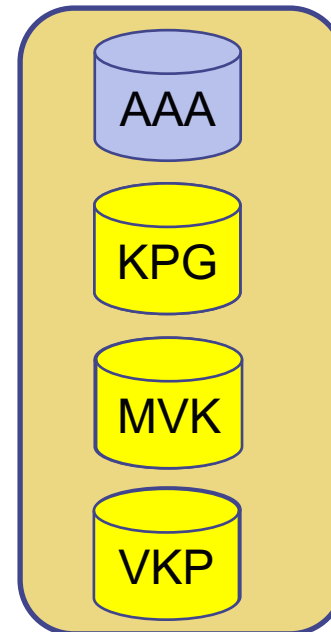
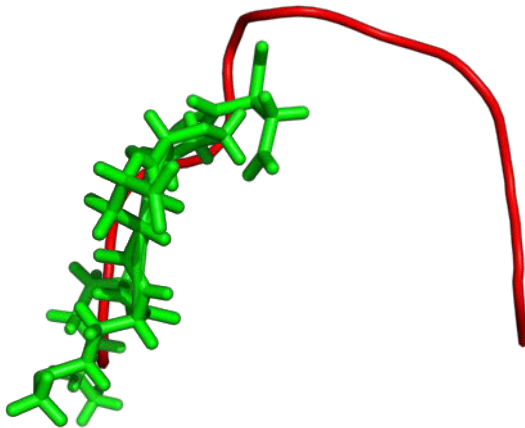
Machine-learning approach to protein loop sampling

[Barozet *et al.*, in preparation]

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

Data: Extracted from ~ 80,000 protein domain structures

M V K P G T F D P E M K



Database
~ 6 million
tripeptide
configurations
from coils

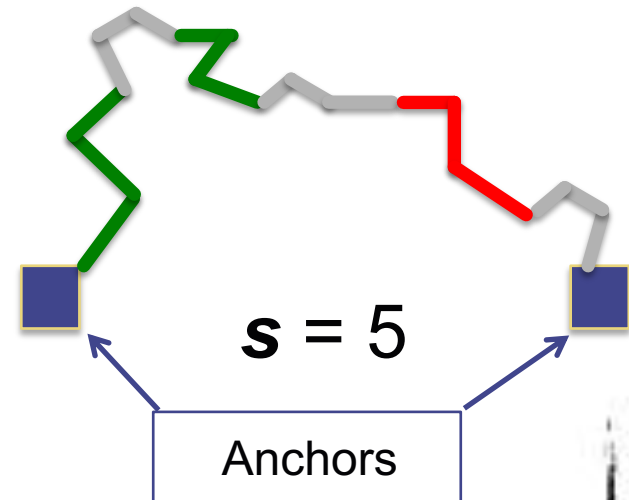
Machine-learning approach to protein loop sampling

[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 $T_1 \dots T_{s-1}, T_{s+1} \dots T_k$
3. Solve for T_s via semi-analytical IK



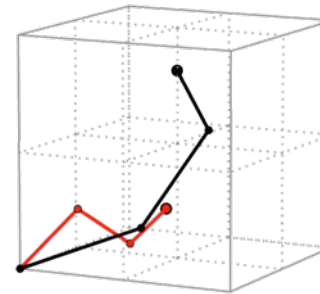
Machine-learning approach to protein loop sampling

[Barozet *et al.*, in preparation]

Tripeptide selection with Reinforcement Learning

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

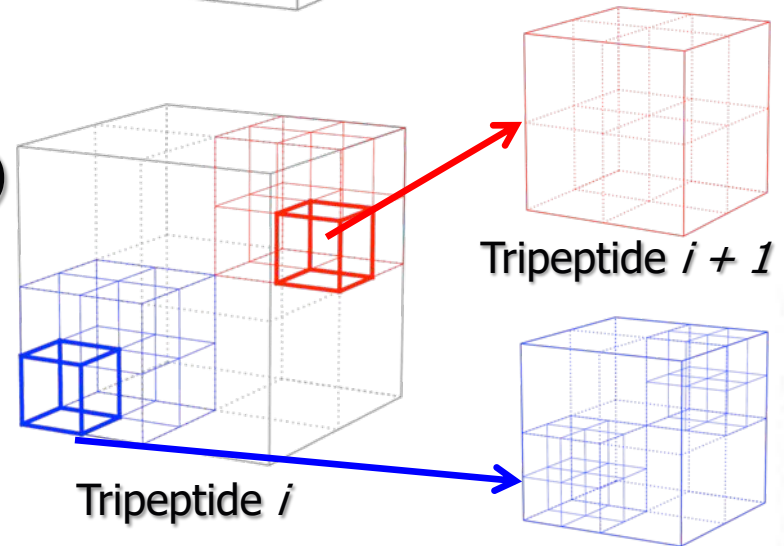
How: Feature-based organization in tree data structure (e.g. octree)



- N1-C3 position
- Length
- Orientation
- ...

- Cell score = rate of success
- Iterative construction (dependency)
 - Scoring function captures downstream results

$$score_c^k = score_c^k \times \prod_{m=k+1}^K score^m$$

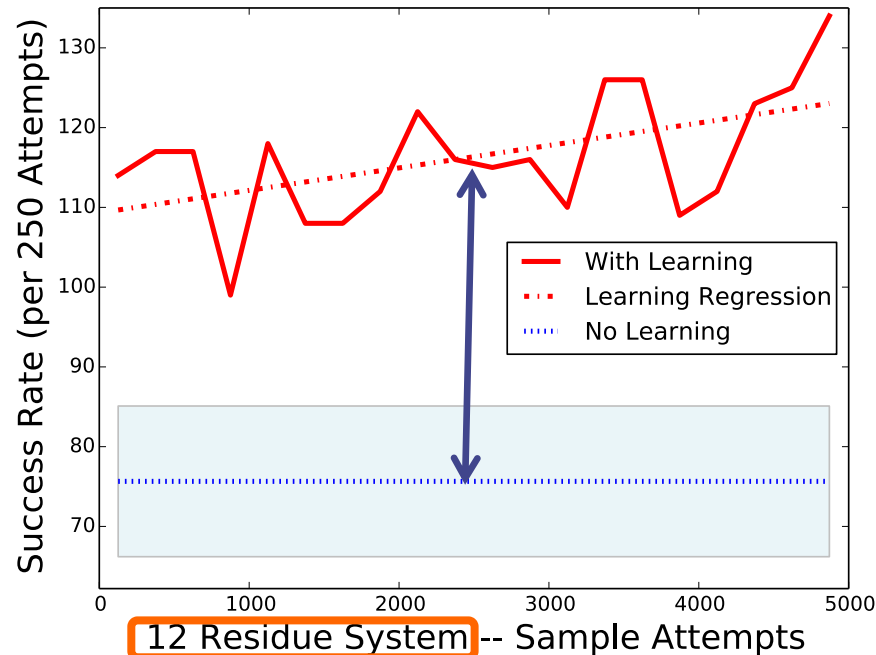


Machine-learning approach to protein loop sampling

[Barozet *et al.*, in preparation]

Early results:

Comparison of learning method against naïve tripeptide selection



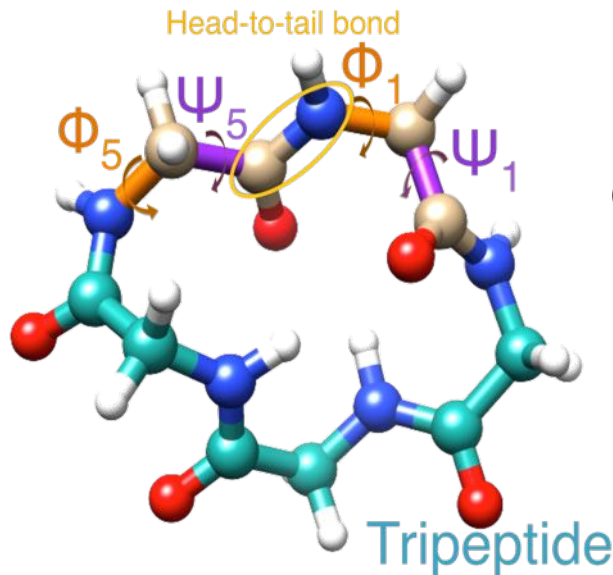
- 4 seconds per computation
- Learning outperforms naïve
- Gain improves with time

Exhaustive sampling of small cyclic peptides

[Jusot *et al.*, JCIM 2018]

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

Cyclic-pentapeptide

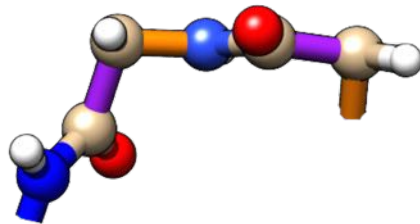


Conformational space (considering only Φ , Ψ)
→ 4-dimension manifold in a 10-dimension space
Can be sampled exhaustively !

Exhaustive sampling of small cyclic peptides

[Jusot *et al.*, JCIM 2018]

Sampling of Φ_1 , Ψ_1 , Φ_5 , Ψ_5 (+ all ω)



Exhaustive sampling of small cyclic peptides

[Jusot *et al.*, JCIM 2018]

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

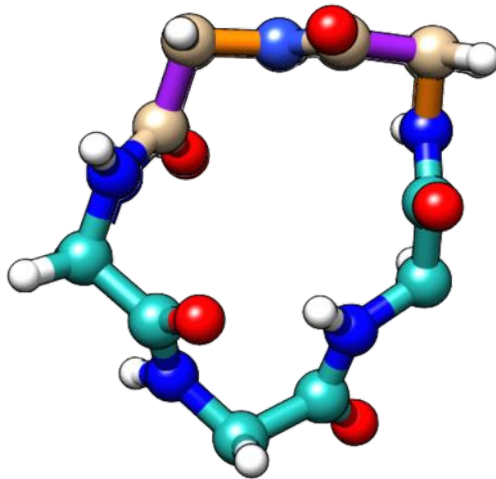
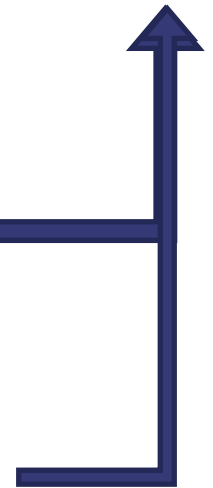


Inverse kinematics:

- 0 solution
- 1 to 16 solutions



Check collisions



Exhaustive sampling of small cyclic peptides

[Jusot *et al.*, JCIM 2018]

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



Inverse kinematics:

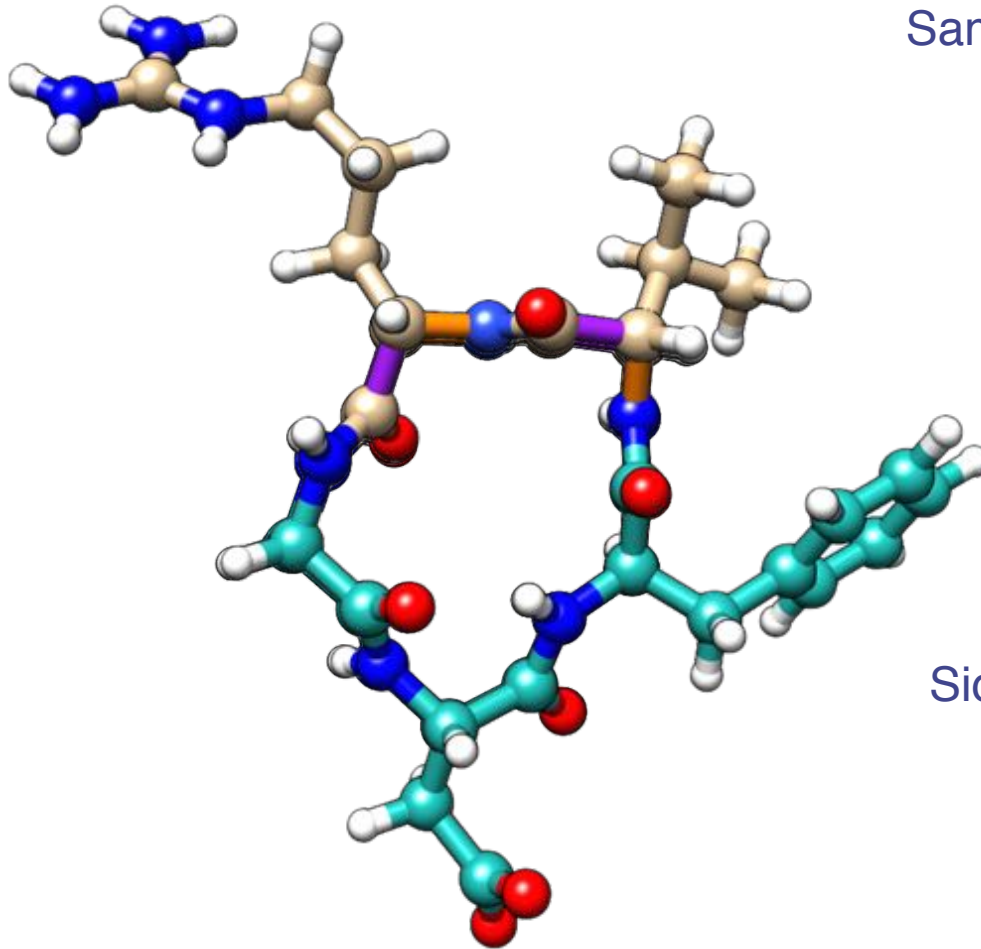
- 0 solution
- 1 to 16 solutions



Check collisions



Side chains addition(SCWRL)



Exhaustive sampling of small cyclic peptides

[Jusot *et al.*, JCI 2018]

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



Inverse kinematics:

- 0 solution
- 1 to 16 solutions



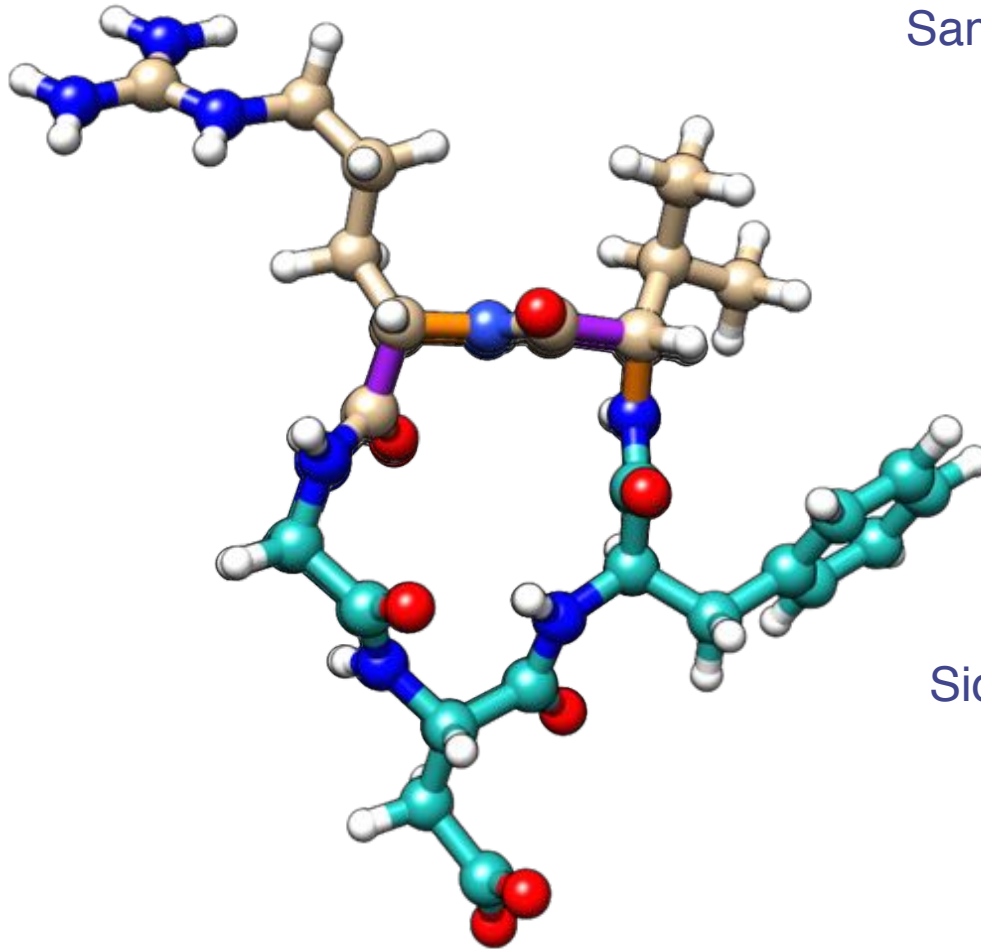
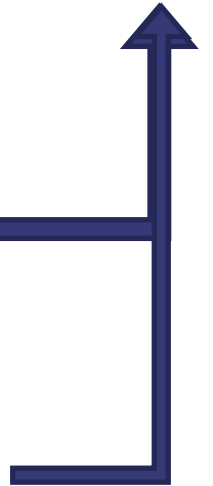
Check collisions



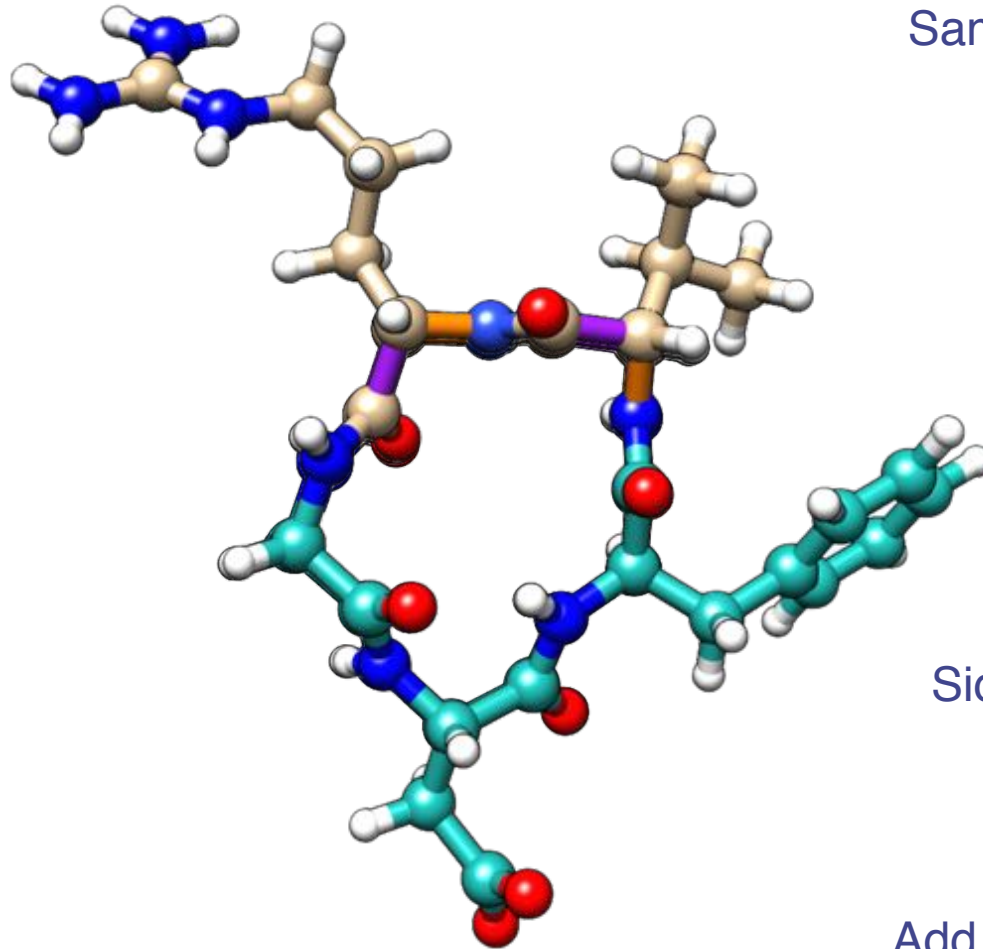
Side chains addition(SCWRL)



Relaxation (Amber)



Exhaustive sampling of small cyclic peptides



[Jusot *et al.*, JCI 2018]

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



Inverse kinematics:

- 0 solution
- 1 to 16 solutions



Check collisions



Side chains addition(SCWRL)



Relaxation (Amber)

Add node to adjacency graph

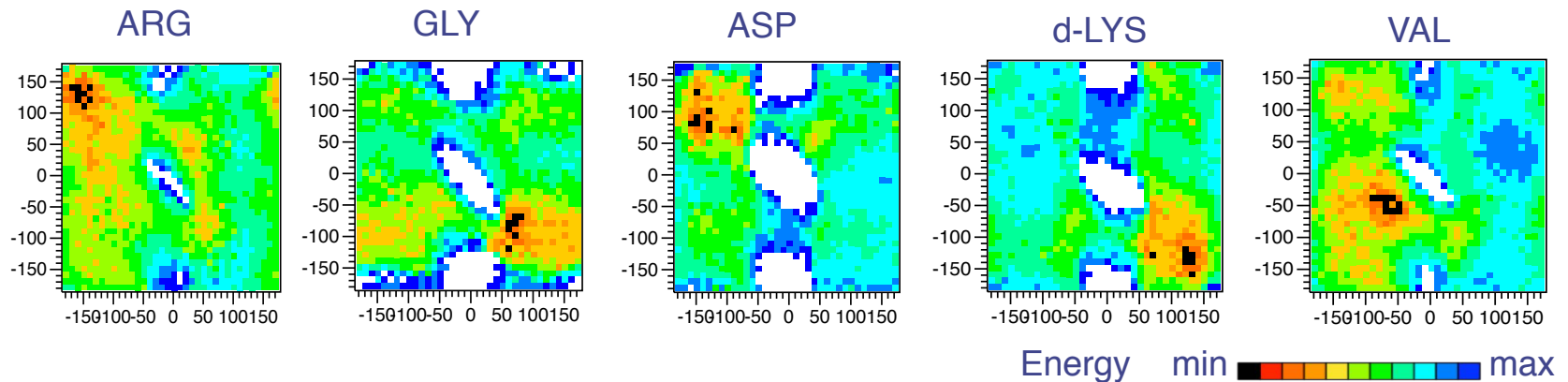


Exhaustive sampling of small cyclic peptides

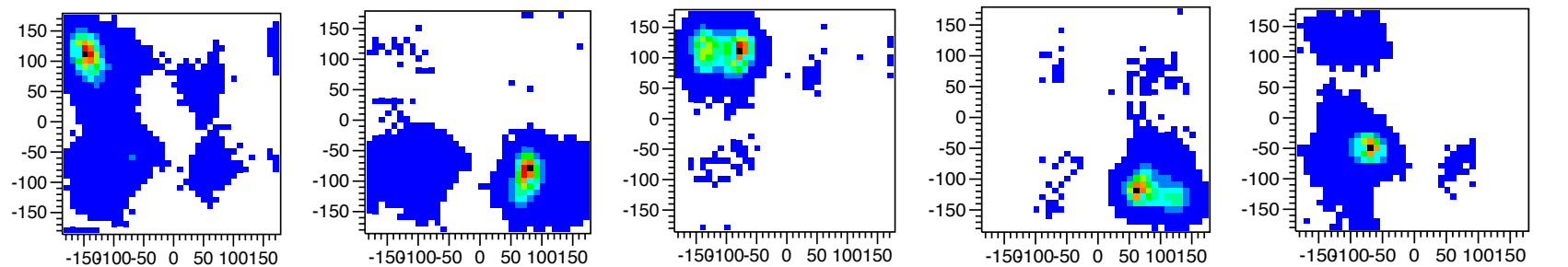
Peptide c(RGDkV)

[Jusot *et al.*, JCIM 2018]

Exhaustive search : 839061 conformations founds



REMD

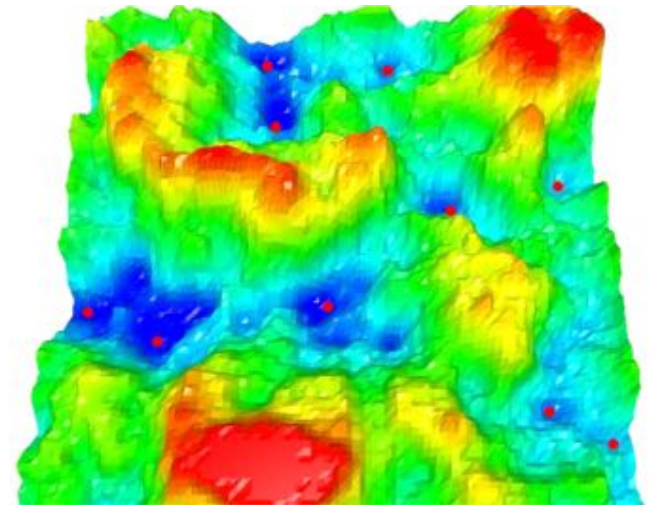


Course Contents

- Loop sampling
- Path sampling on energy landscapes

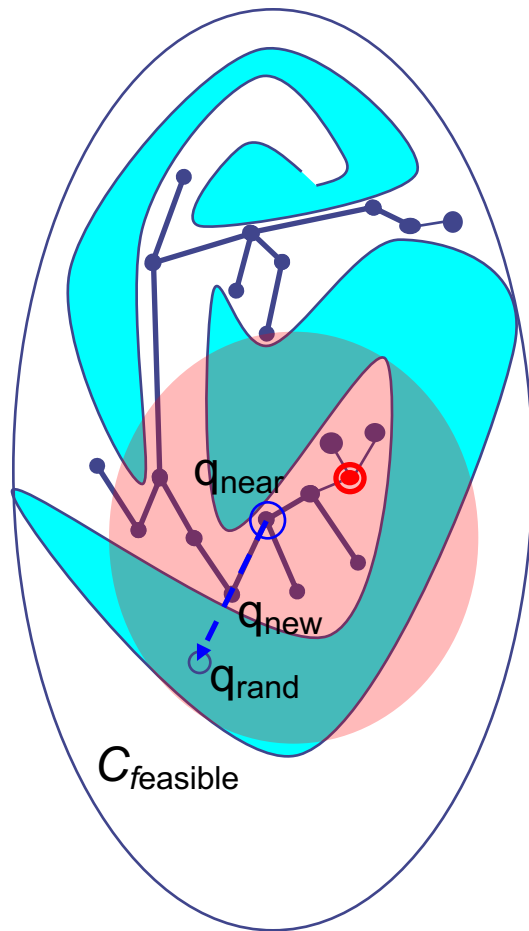
NOT COVERED

- Conformational transitions in proteins
- Protein-ligand access/exit pathways
- Towards molecular motion design



RRT

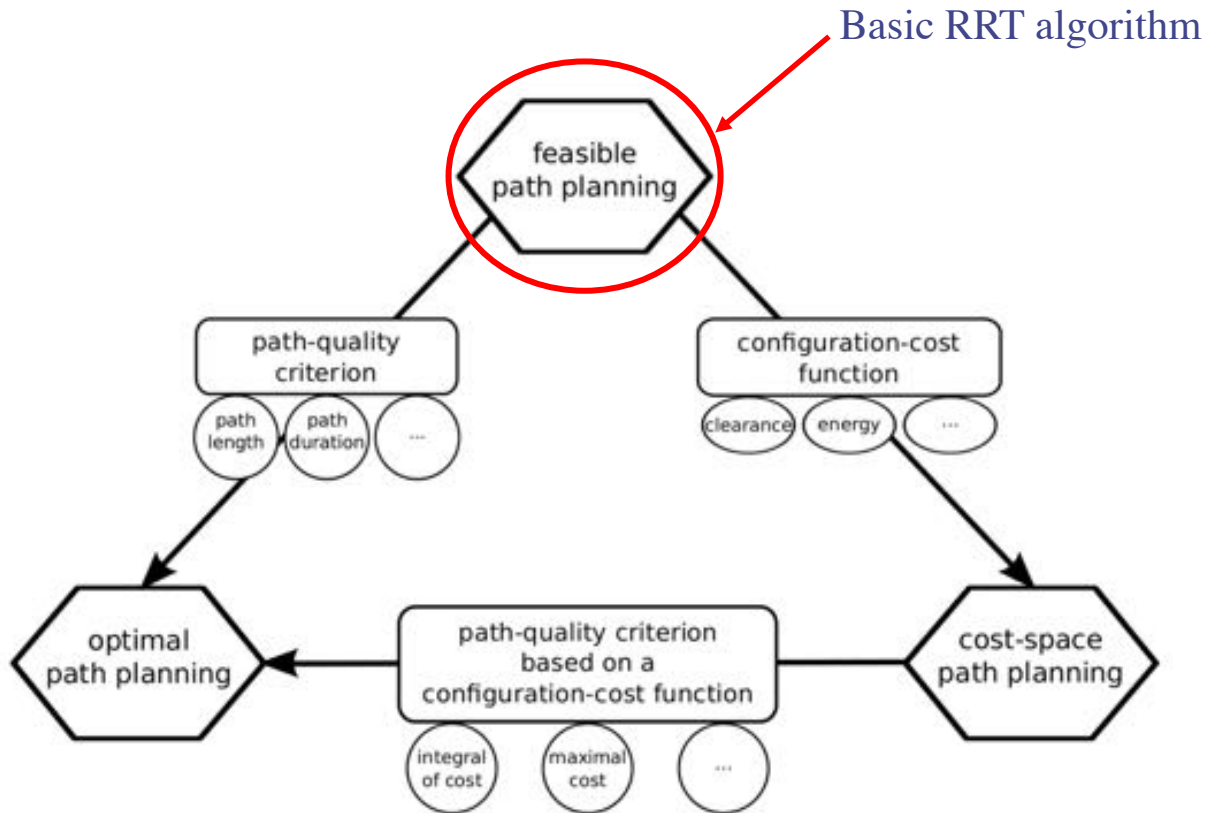
[LaValle 98][LaValle and Kuffner 01]



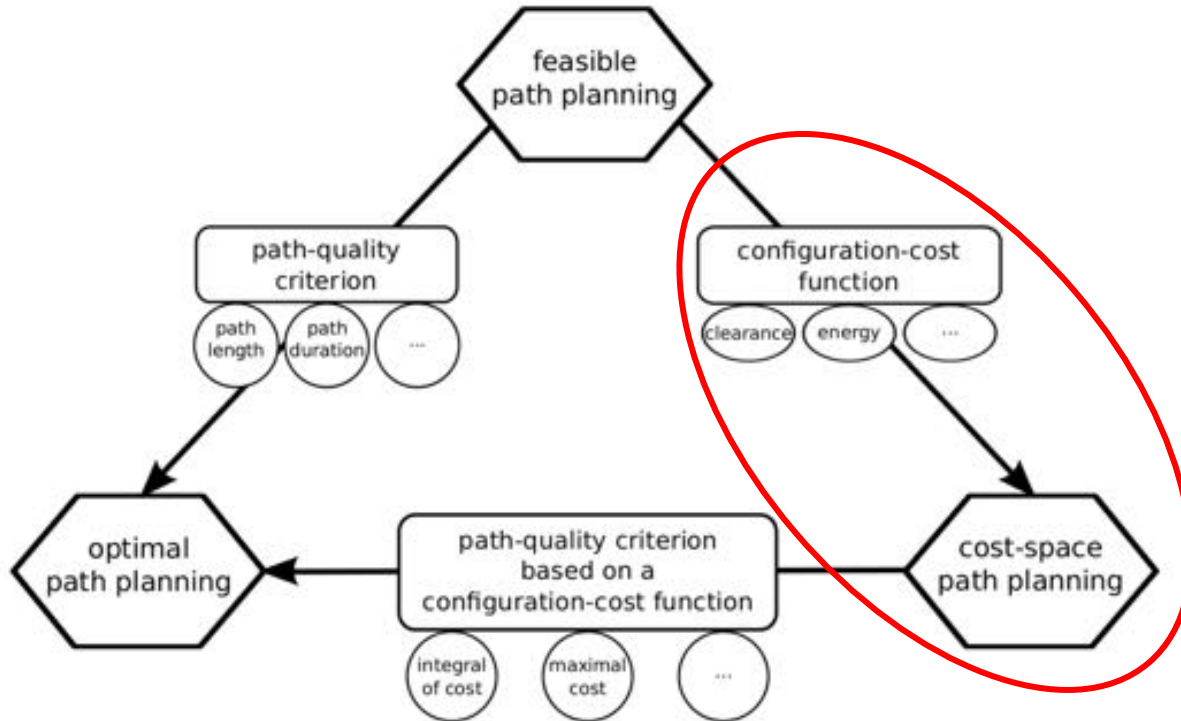
Algorithm : Construct_RRT

```
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

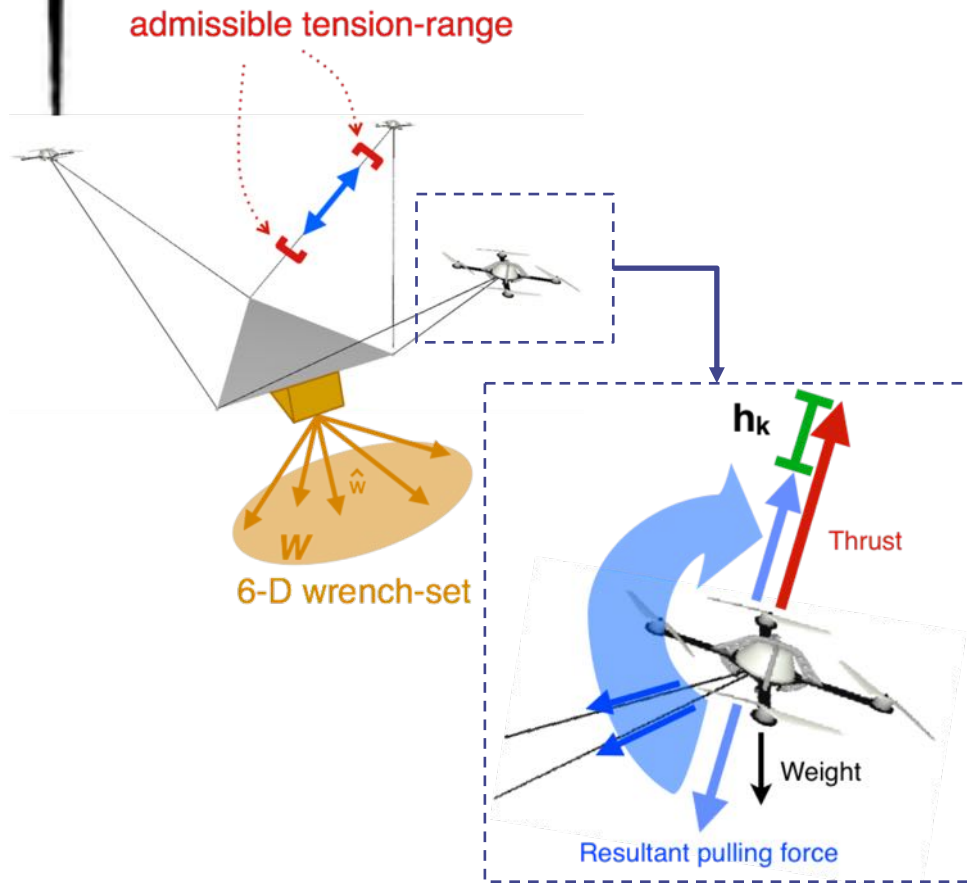


Types of Path Planning Problems

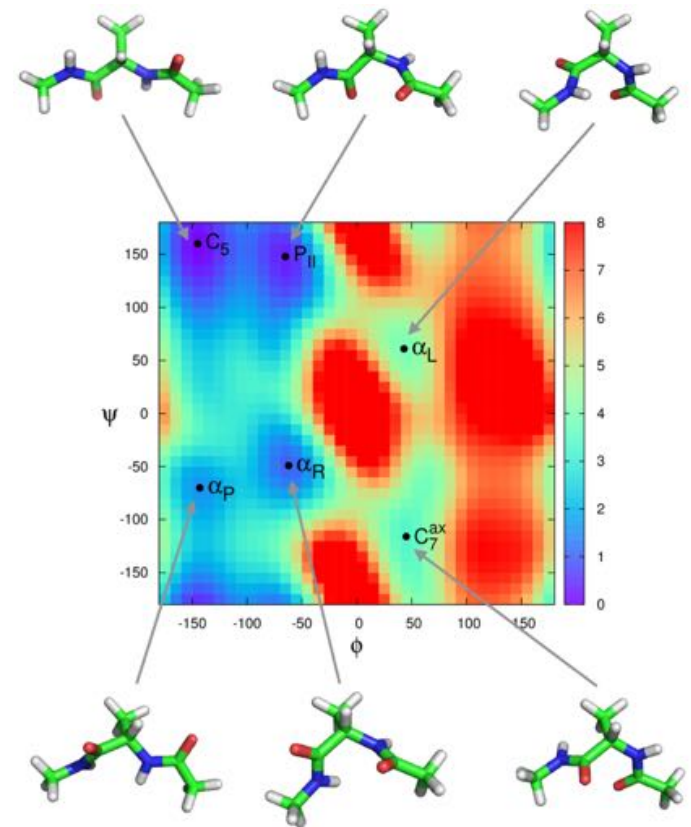


Examples of cost-space path planning problems

In robotics

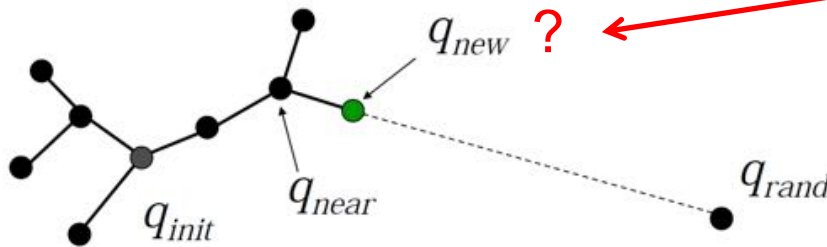


In structural biology



Cost-space Path Planning : the T-RRT algorithm

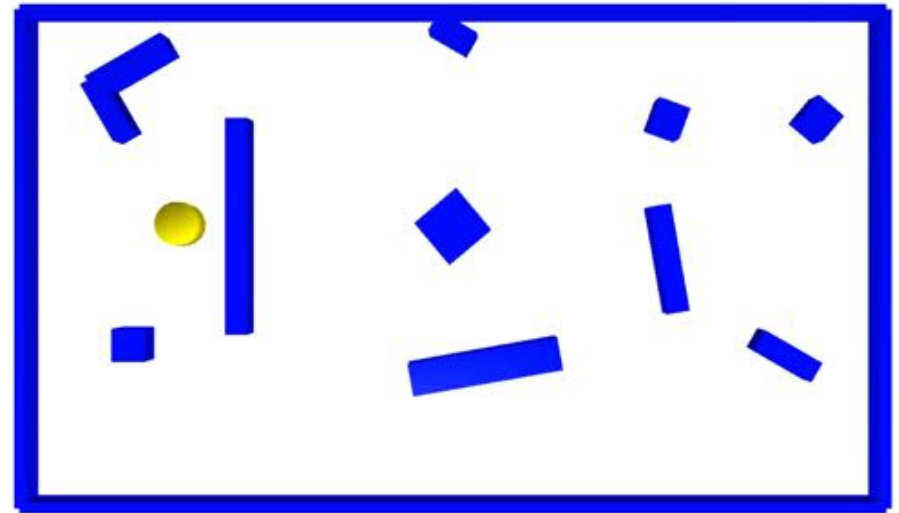
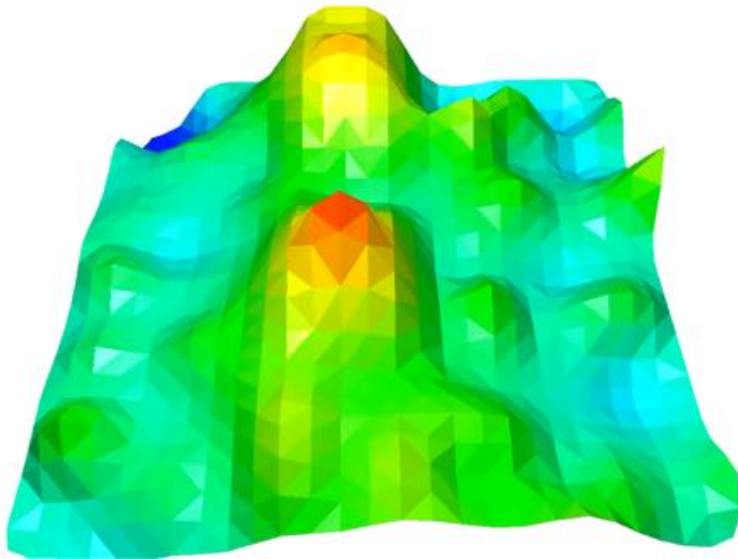
[Jaillet *et al.*, T-RO, 2010] [Jaillet *et al.*, J Comput Chem, 2011]



Stochastic transition test

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

Adaptive parameter



T-RRT : Temperature Self-Tuning

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

Determines greediness

Algorithm : transitionTest (\mathcal{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

→ **if** $c_j \leq c_i$ **then return** True

→ **if** $\exp(-(c_j - c_i) / T) > 0.5$ **then**

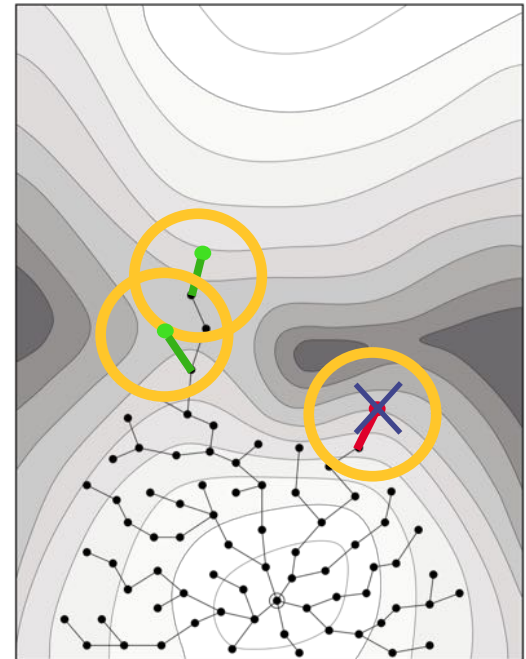
→ $T \leftarrow T / 2^{(c_j - c_i) / \text{costRange}(\mathcal{T})}$; **return** True

else

→ $T \leftarrow T \cdot 2^{T_{rate}}$; **return** False

$T_{rate} = 0.1$ yields good performance in most cases

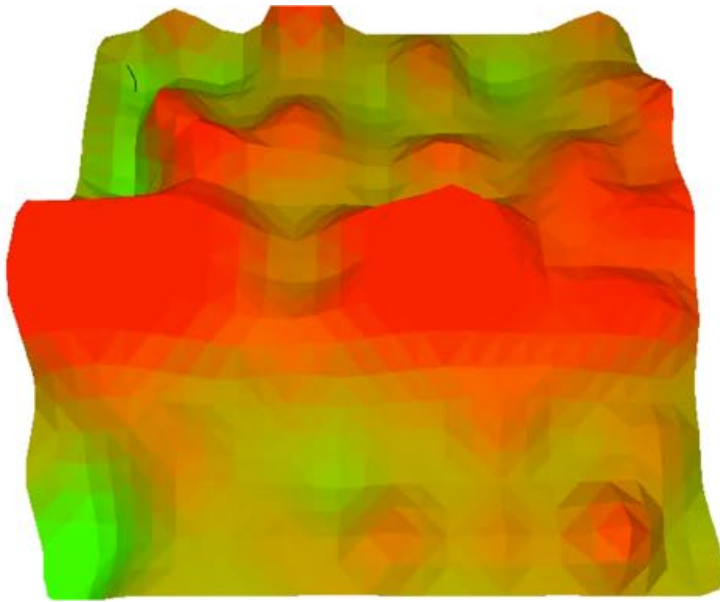
T ↑↓



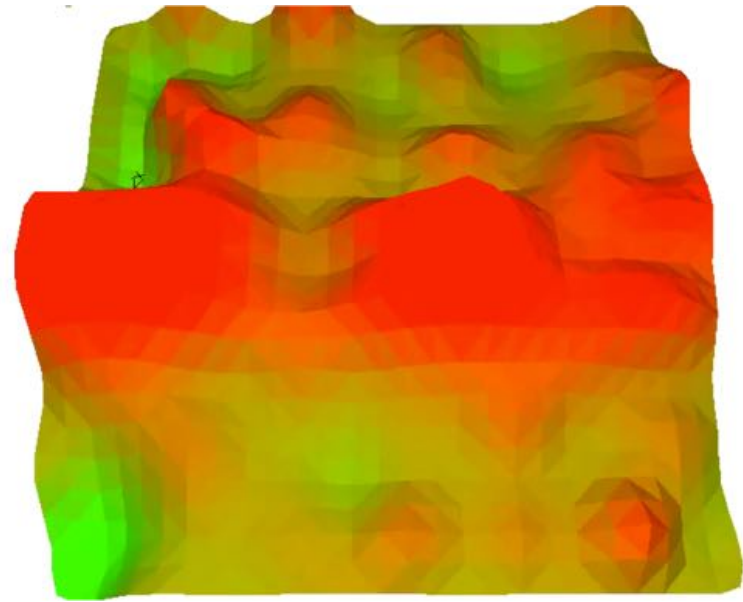
Transition-based RRT (T-RRT)

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

- Very different behavior compared to a basic Monte Carlo method



T-RRT

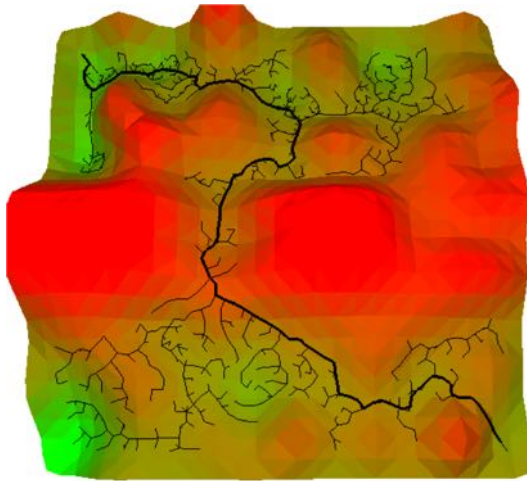


Monte Carlo

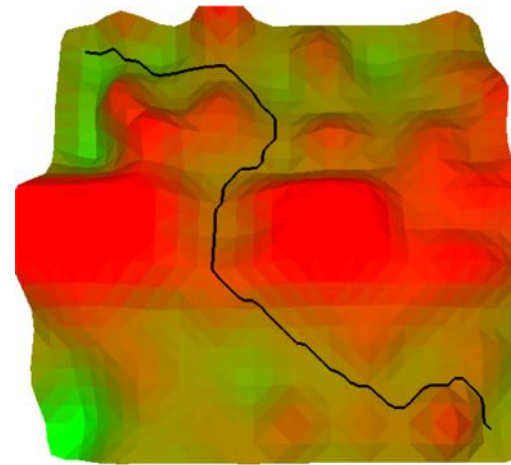
Transition-based RRT (T-RRT)

[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



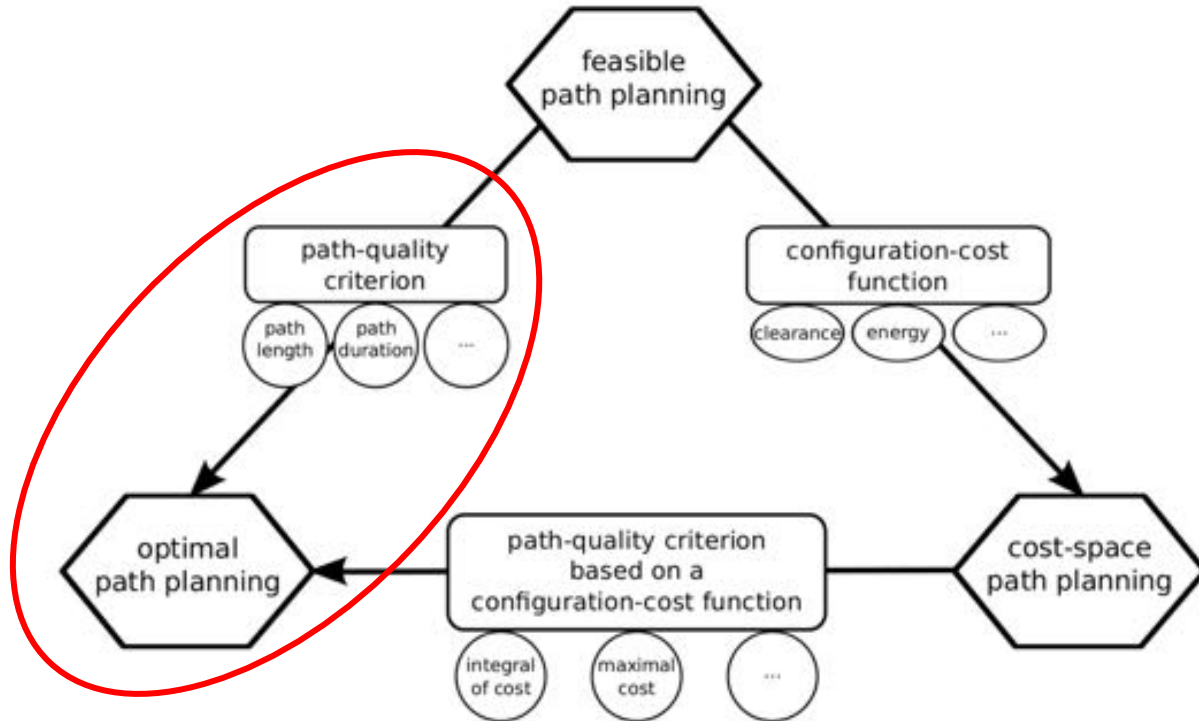
T-RRT



Optimal path : Minimum of $W(P)$

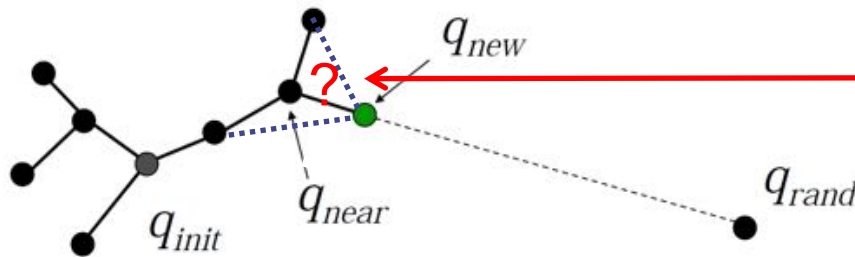
$$W(P) = \int_{s_+} \frac{\partial c_+}{\partial s} ds + \varepsilon \int_s ds$$

Types of Path Planning Problems



Optimal Path Planning : the RRT* algorithm

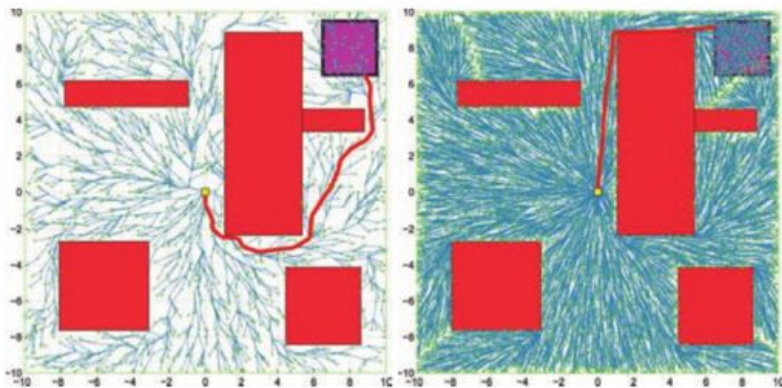
[Karaman and Frazzoli, RSS, 2010]



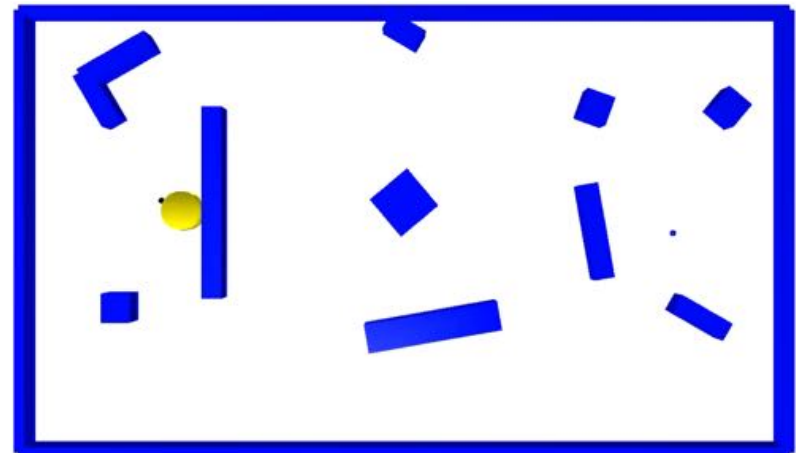
“Re-wiring” process

the edge created is the one minimizing the cost from q_{init} to q_{new} among the neighbors of q_{new}

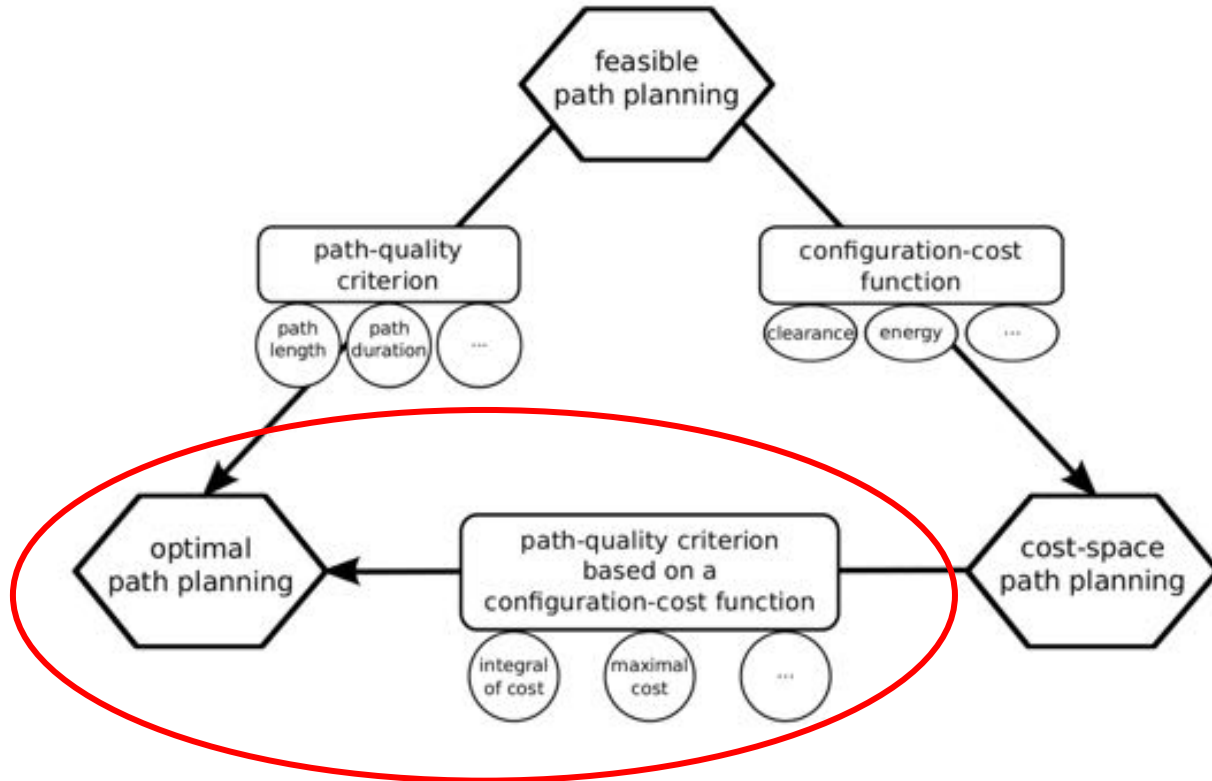
- **Asymptotic convergence** to the global optimum
- Basically conceived to minimize an additive cost along the path (e.g. length)



courtesy: Sertac Karaman



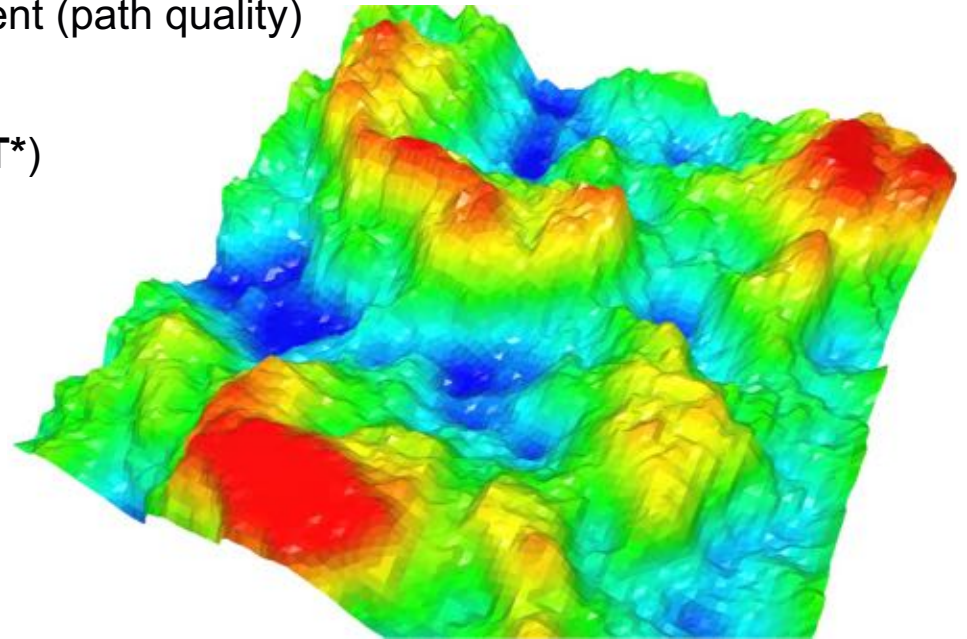
Types of Path Planning Problems



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 $(\mathcal{C}, q_{\text{init}}, q_{\text{goal}}, c, c_p)$

output: the graph \mathcal{G}

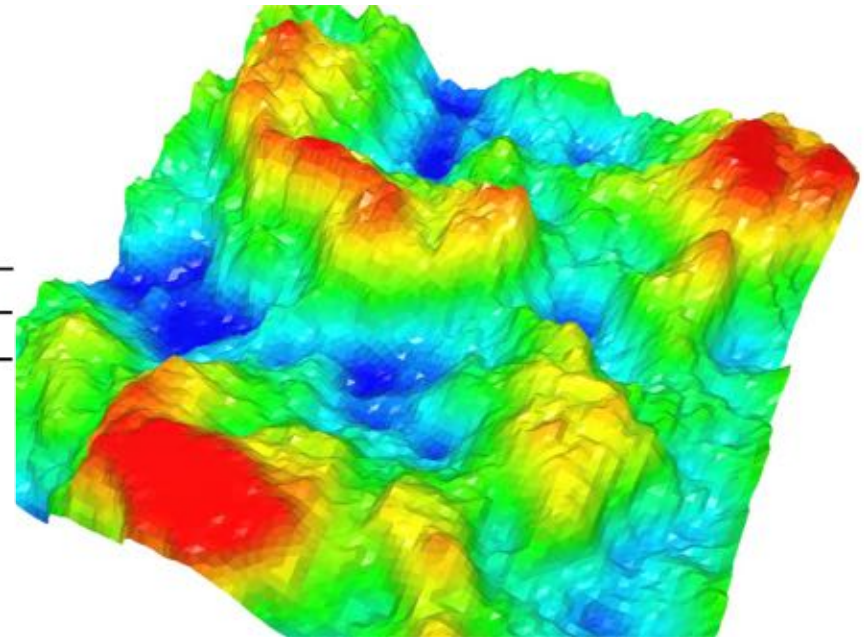
```
1  $\mathcal{G} \leftarrow \text{initGraph}(q_{\text{init}})$ 
2 while not stoppingCriteria( $\mathcal{G}$ ) do
3    $q_{\text{rand}} \leftarrow \text{sampleRandomConfiguration}(\mathcal{C})$ 
4    $q_{\text{near}} \leftarrow \text{findNearestNeighbor}(\mathcal{G}, q_{\text{rand}})$ 
5    $q_{\text{new}}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

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

input: the dimension d of the \mathcal{C} -space

the γ constant derived from the volume of $\mathcal{C}_{\text{free}}$

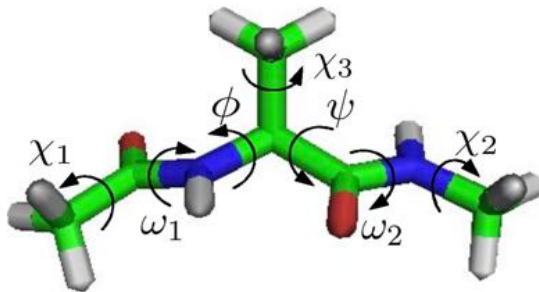
```
1  $n \leftarrow \text{numberOfNodes}(\mathcal{G})$ 
2  $Q_{\text{near}} \leftarrow \text{nodesInBall}(\mathcal{G}, q_{\text{new}}, \gamma (\log(n)/n)^{1/d})$ 
3 foreach  $q_n \in Q_{\text{near}}$  do
4    $\pi_g \leftarrow \text{pathInGraph}(\mathcal{G}, q_{\text{new}}, q_n)$ 
5    $\pi_s \leftarrow \text{pathInSpace}(q_{\text{new}}, q_n)$ 
6   if  $c_p(\pi_s) < c_p(\pi_g)$  and isCollisionFree( $\pi_s$ ) then
7     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**



Energy function:

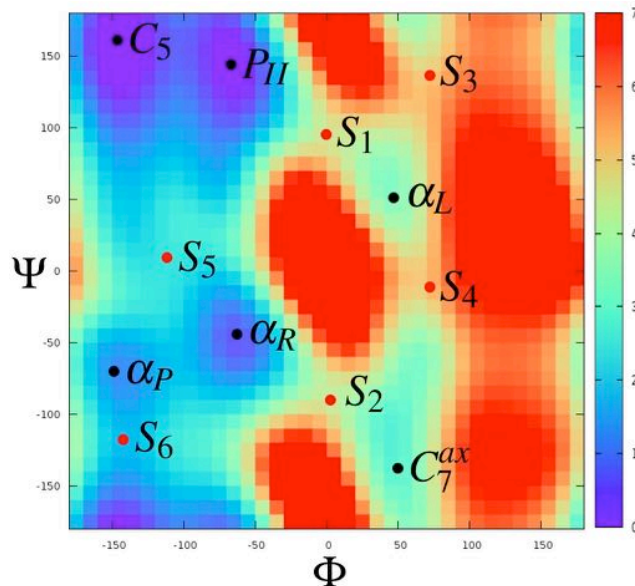
- parm96 AMBER force field
- implicit solvent (GB)

Energy minima:

	P_{II}	α_R	α_L	C_7^{ax}	α_P	C_5
ϕ	-67	-63	47	50	-148	-146
ψ	144	-44	51	-138	-70	162
E	0.3	1.1	4.4	4.2	1.7	0.0

Main transition states:

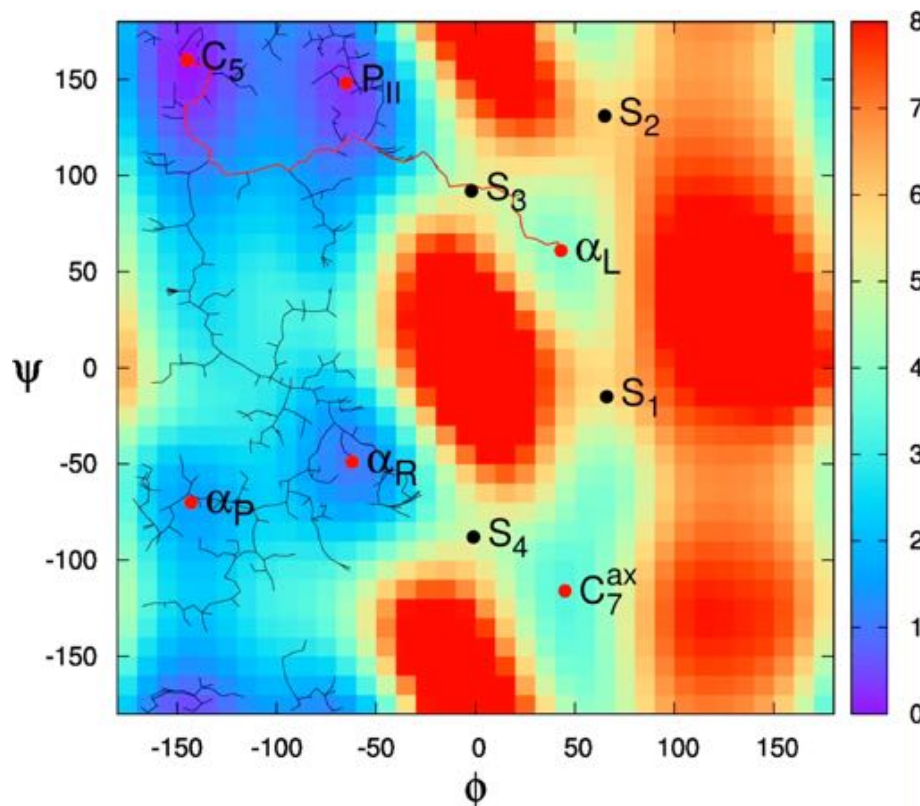
	S_1	S_2	S_3	S_4	S_5	S_6
ϕ	0	3	72	74	-111	-142
ψ	95	-90	137	-8	10	-118
E	7.3	7.7	7.3	7.7	3.4	2.6



An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

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

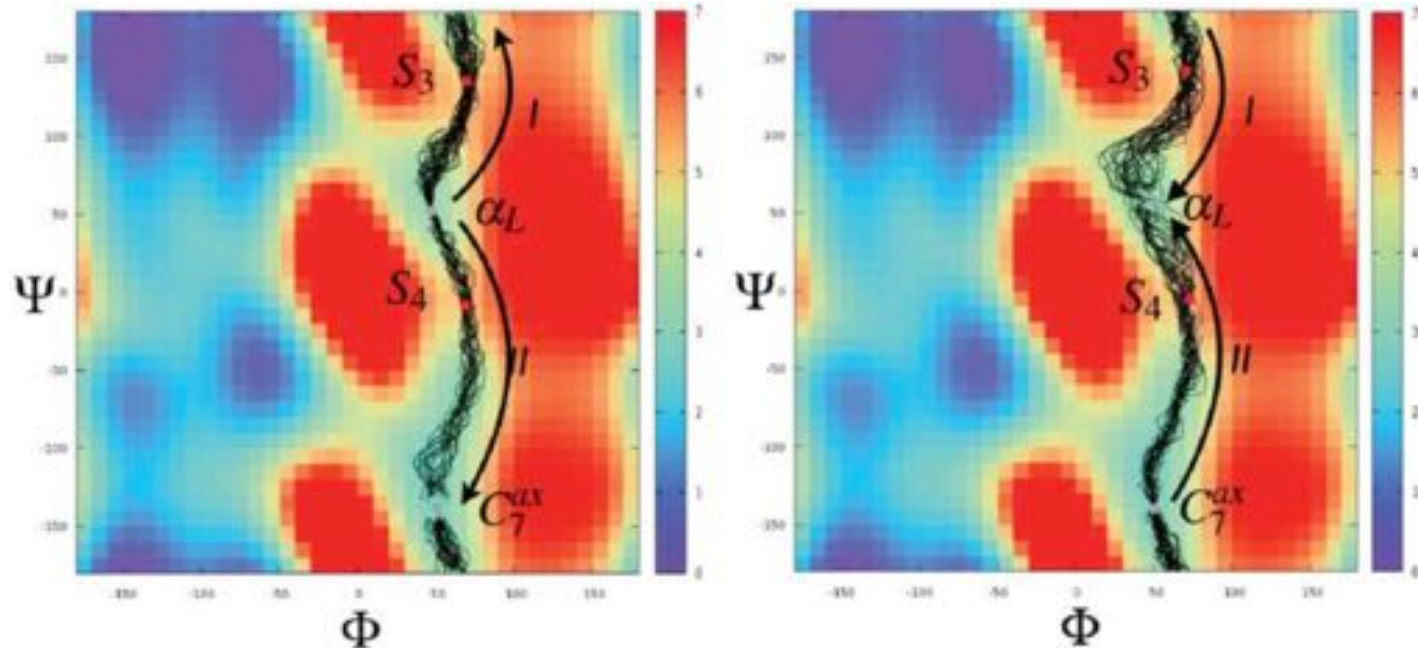
- **Alanine dipeptide** : Example of one run of T-RRT for $C_5 \rightarrow \alpha_L$
CPU time \cong 1 sec. (on a single processor)



An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

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

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

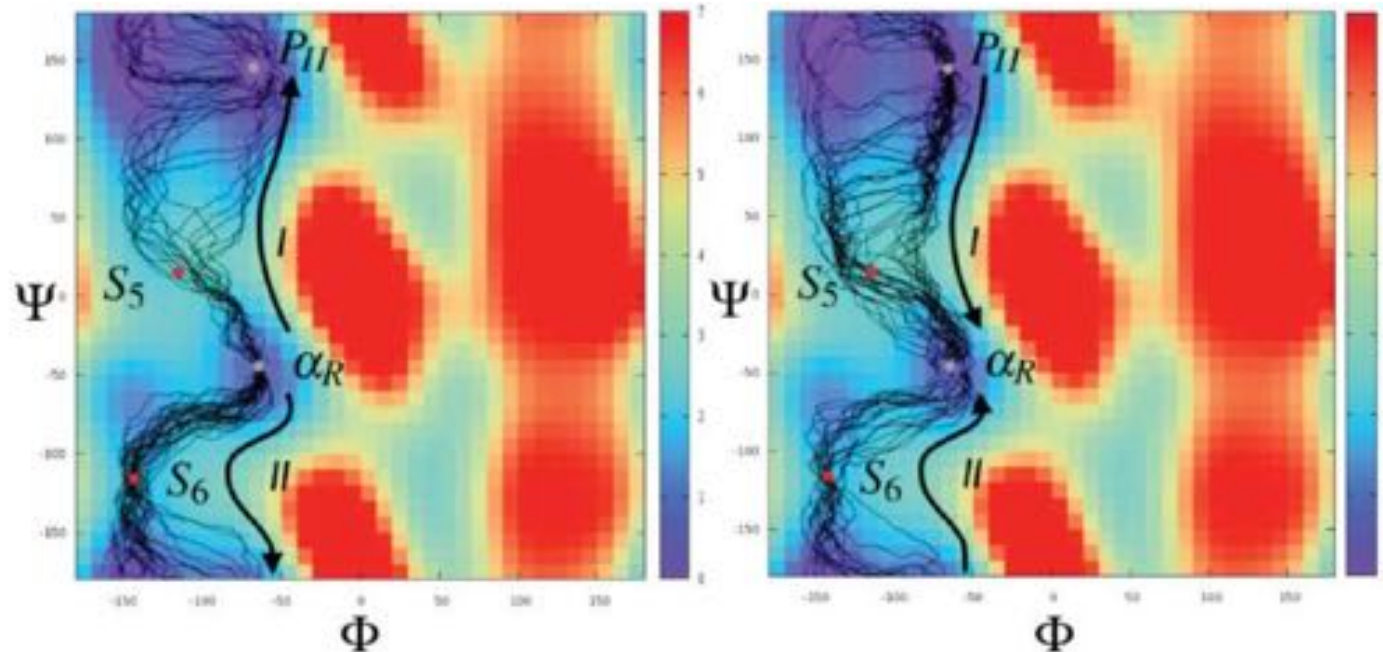


	I (%)	II (%)
$\alpha_L \rightarrow C_7^{ax}$	62	38
$C_7^{ax} \rightarrow \alpha_L$	60	40

An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

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

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

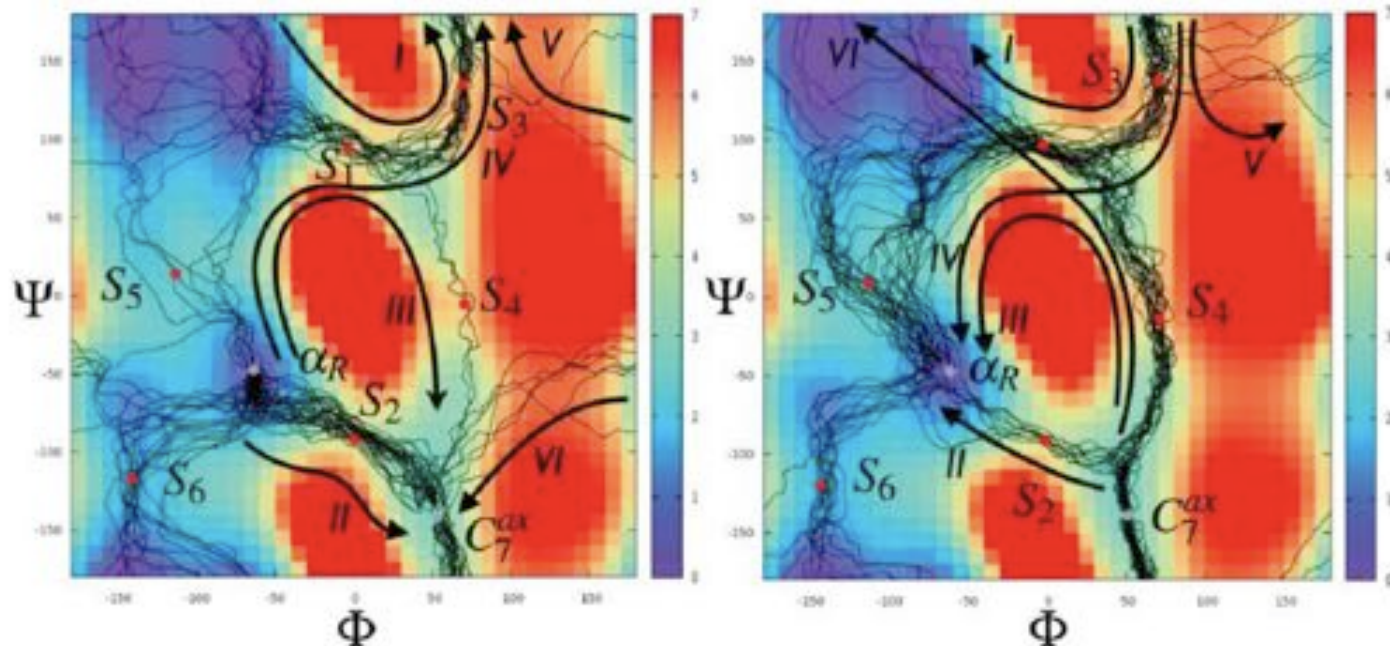


	I (%)	II (%)
$\alpha_R \rightarrow P_{II}$	34	66
$P_{II} \rightarrow \alpha_R$	64	36

An Example of Application in Structural Biology: Modeling Peptide Conformational Transitions

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

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



	I (%)	II (%)	III (%)	IV (%)	V (%)	VI (%)
$\alpha_R \rightarrow C_7^{ax}$	21	54	2	8	2	13
$C_7^{ax} \rightarrow \alpha_R$	9	17	31	27	11	5

The Ordering-and-Pathfinding Problem

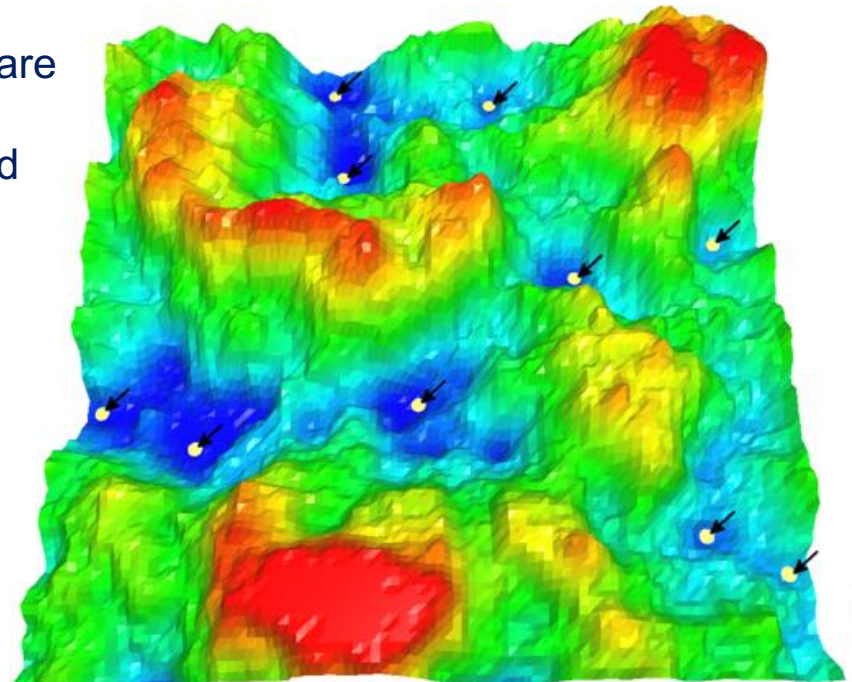
«Off-road» TSP

- Paths (and costs) between pairs of states are unknown *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

Can be interleaved and solved in an anytime manner



The Ordering-and-Pathfinding Problem: Solved with the Anytime Multi-T-RRT Algorithm

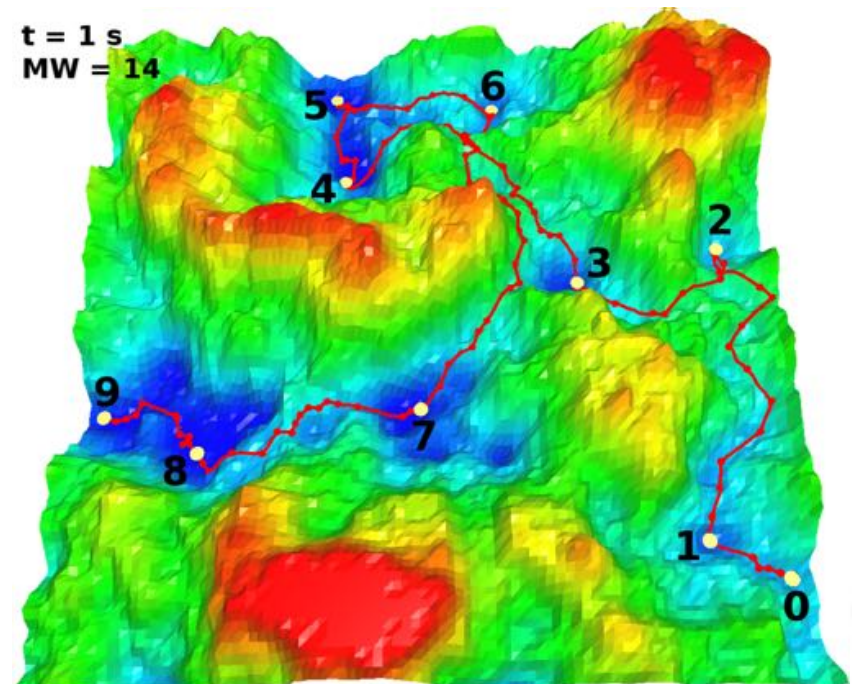
[Devaurs *et al.*, IEEE IROS, 2014]

T-RRTs construction

- Build n trees rooted at the given states
- Until all trees are connected

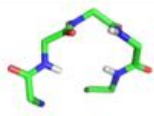

Useful cycle addition

- Incremental local improvements
- Guarantees asymptotic convergence to the global optimum



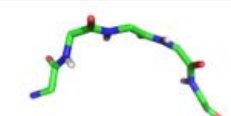
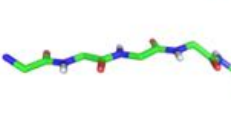
A more interesting peptide: Met-enkephalin

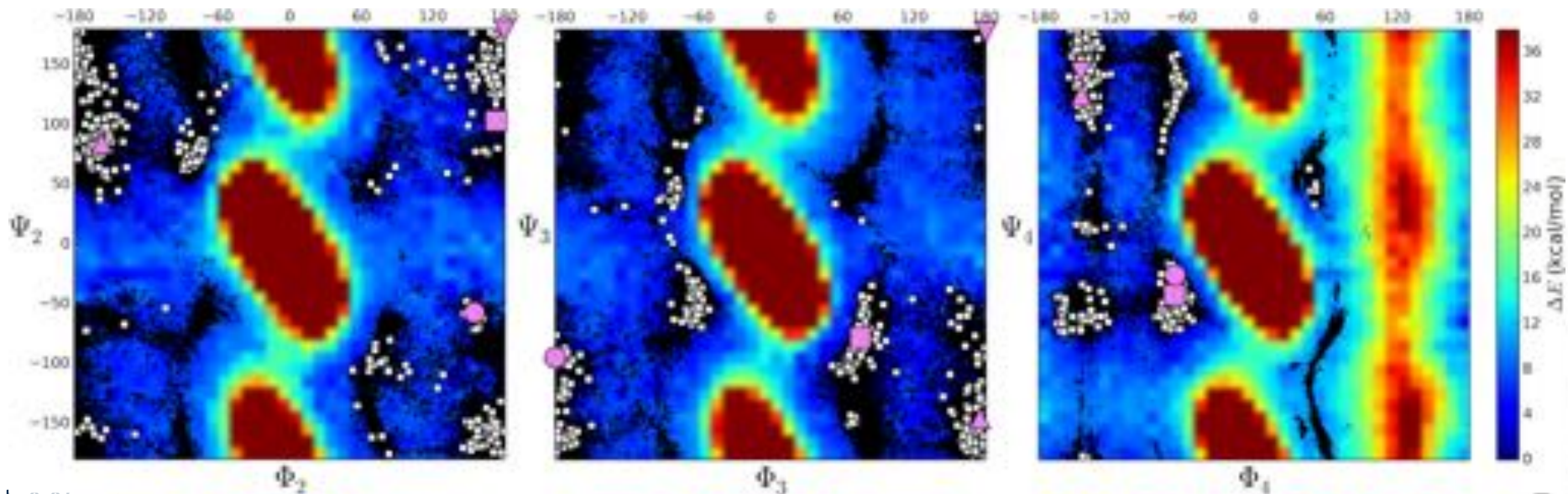
[Devaurs *et al.*, IEEE TNB, 2015]

Symbol	Conformation	4-sign code	Energy (kcal/mol)
■		+ - - +	-217.9
●		- - - -	-216.5

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

[Banerjee&Cukier, J Phys Chem B, 2014]

▲		+ - + +	-215.9
▼		* + + *	-212.7

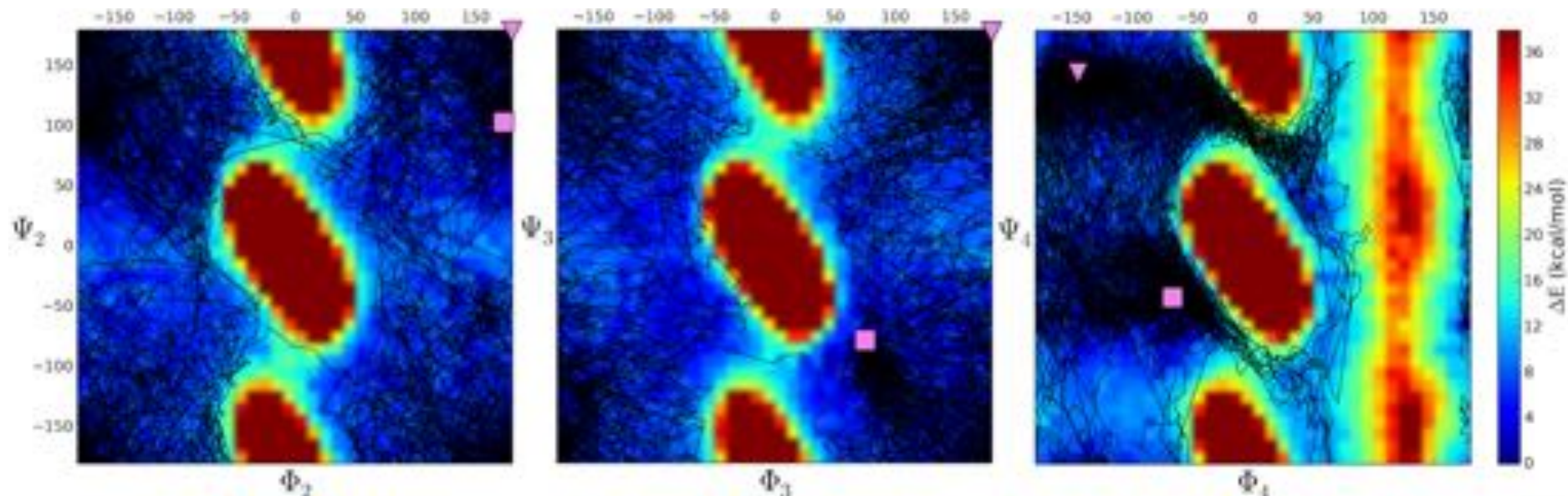


A more interesting peptide: Met-enkephalin

[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

[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

Symbol	Conformation	4-sign code	Energy (kcal/mol)	Transition To							
				Trans Prob	Path Cost (MW)	Trans Prob	Path Cost (MW)	Trans Prob	Path Cost (MW)	Trans Prob	Path Cost (MW)
■		+---+	-217.9	-	-	0.64	125.5 ± 34.7	0.63	105.5 ± 25.9	1.0	48.2 ± 8.9
●		-----	-216.5	0.64	123.8 ± 34.7	-	-	1.0	55.3 ± 18.3	0.86	93.7 ± 27.0
▲		+---++	-215.9	0.63	103.5 ± 25.9	1.0	55.1 ± 18.3	-	-	0.89	72.6 ± 31.3
▼		*+*+*	-212.7	1.0	43.0 ± 8.9	0.86	90.2 ± 27.0	0.89	69.4 ± 31.3	-	-

Parallelization of RRT-based algorithms

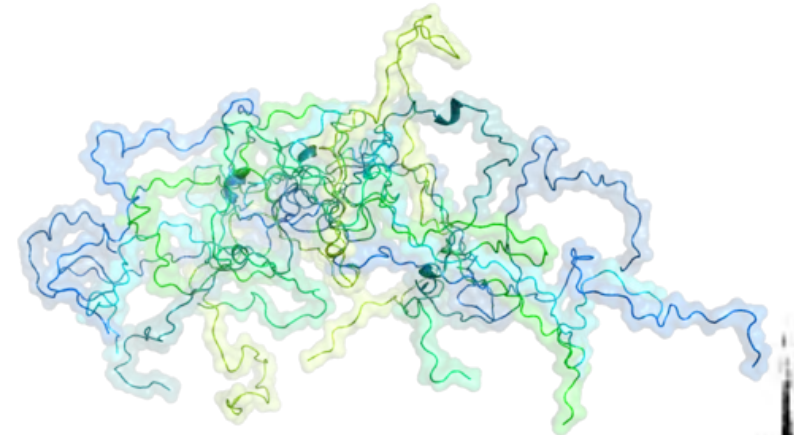
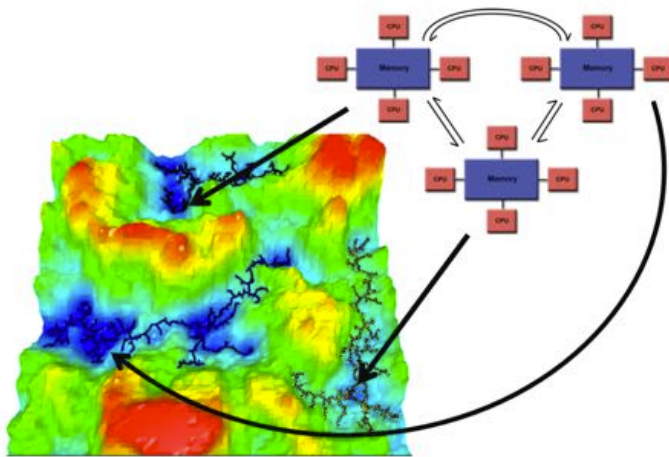
Hybrid shared/distributed-memory implementation:
For execution on computer clusters

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



EOS (CALMIP)

Parallel Multi-TRRT



Study of IDPs/IDRs



Question ?

juan.cortes@laas.fr