Geometric algorithms for conformational analysis of protein loops

J.Cortes, T.Simeon, M. Remaud-Simeon,V. Tran

Presented by Vera Vsevolozhsky

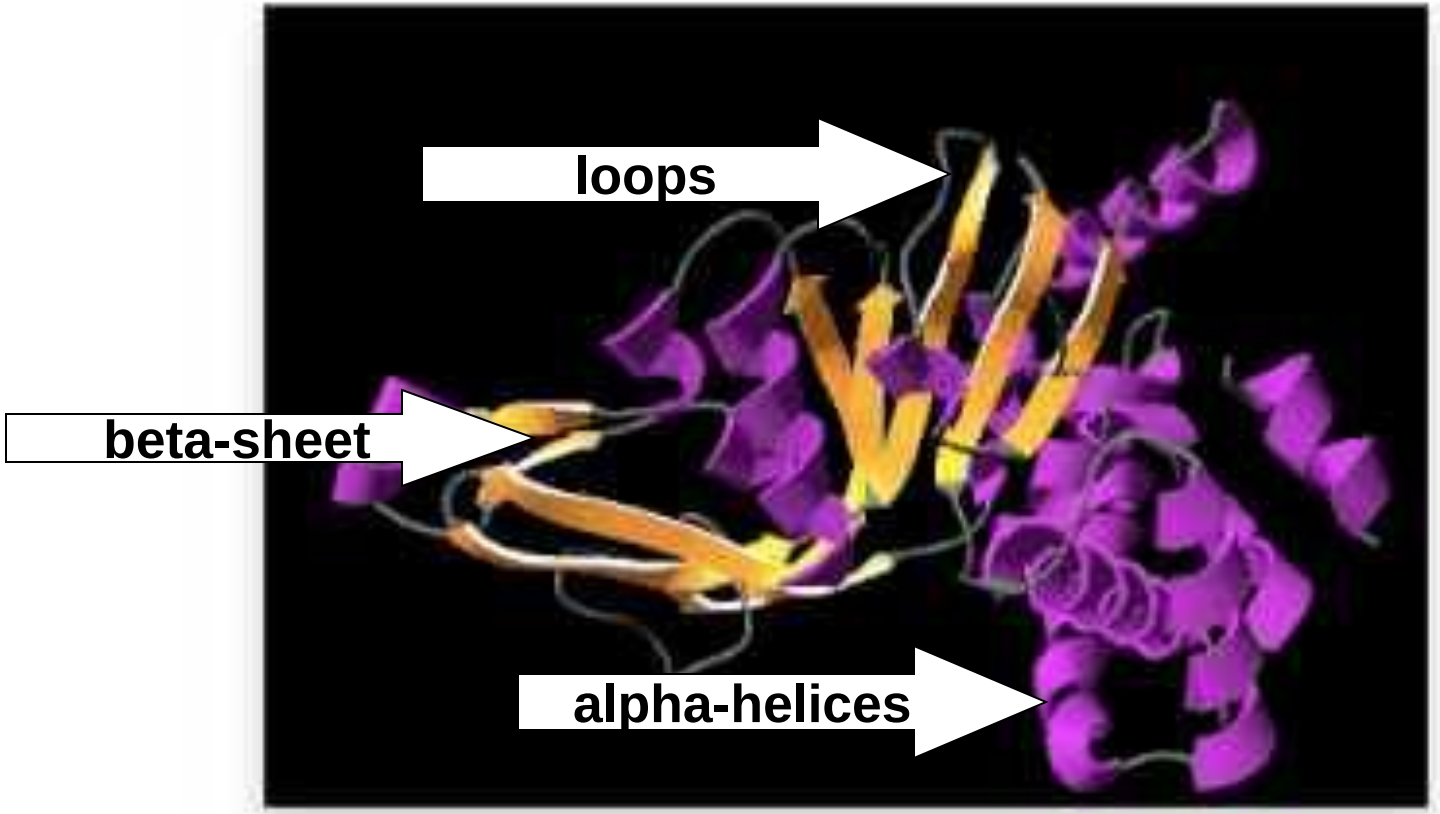
Agenda

- Introduction and Motivation
- Problem Formulation
- Conformational sampling
- Conformational Space Exploration
- Results
- Conclusions
- Prospects

Introduction-outline

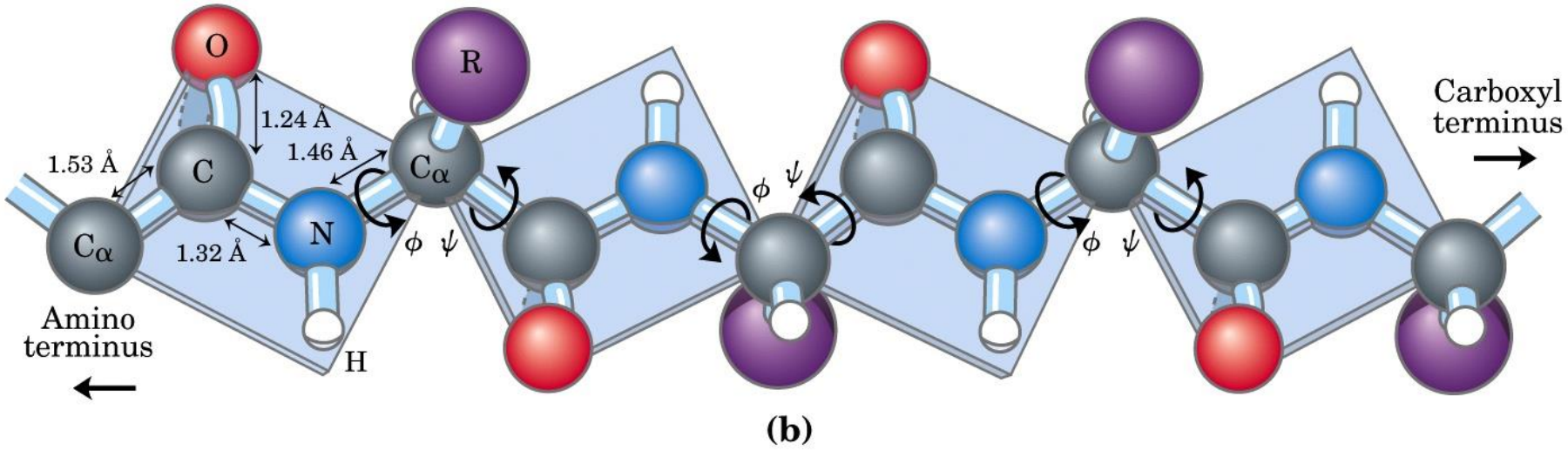
- Protein secondary structure
- Protein loop structure
- Loop closure and steric clashes
- Loop closure/clash filtering approaches
- Analogy to Robotics

Protein secondary structure



Protein loop structure

- Backbone chain
- Side chains (R)

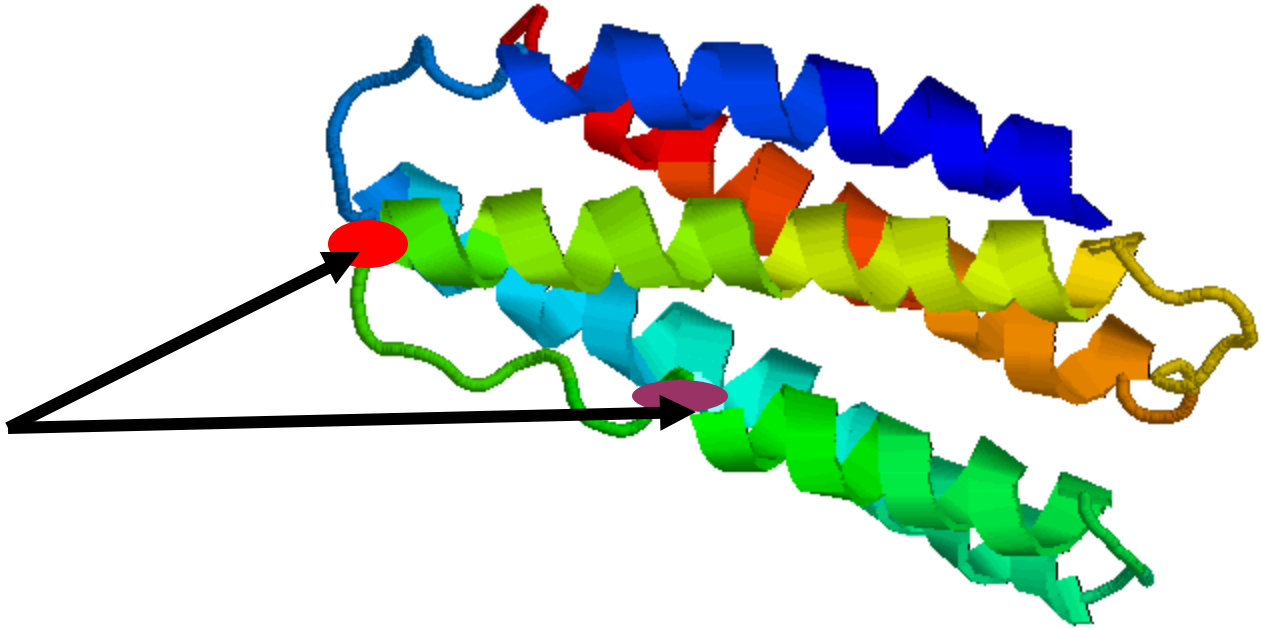


Constants **: bond lengths, bond angles**

Variables **: dihedral angle**

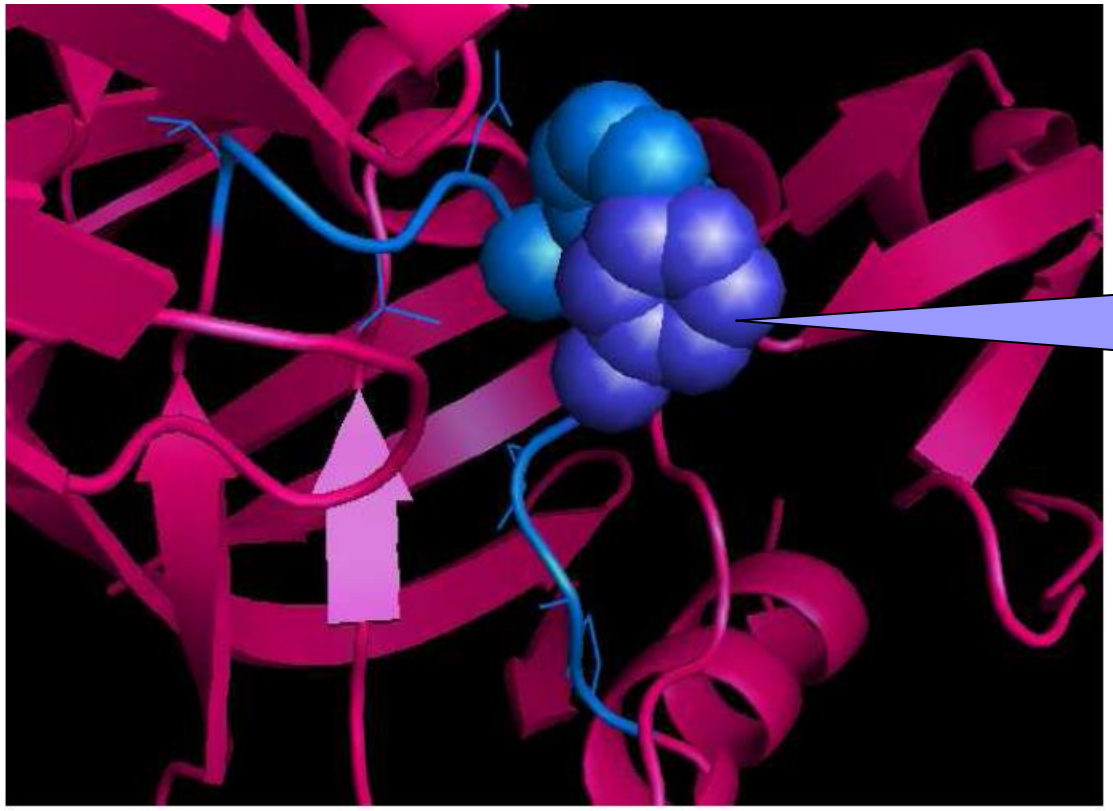
Loop closure

- The first and last atoms of the treated segment of a molecular chain must remain bonded with their neighbor atoms



Steric clash

- Distance(Atom1, Atom2) > Sum ***van der Waals*** radii of (Atom1, Atom2)



(a)

Example of clashes between 2 side-chains in 1MPP

Self-clashes and Internal clashes are forbidden

Loop modelling approaches

- *Ab initio* loop prediction

  -Conformational search for minimum energy structures (simulating annealing, gradient decent)
- Database approach to loop prediction

Loop Closure Approaches(1)

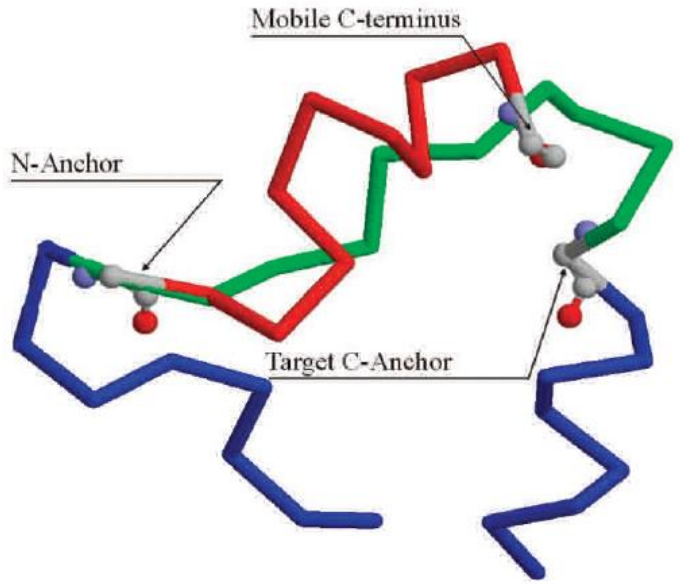
- Analytical – inverse kinematics

  -Wedemeyer and Scheraga(1999) solution for tripeptides with 6 degree of freedom, etc.
- Database based methods

Difficulty increases with the length of the molecular chain

Loop Closure Approaches(2)

- Optimization – e.g. CCD



Difficulty increases with the length of the molecular chain

Clash Filtering Approaches

- Energetic – accepting/rejecting a conformation according to some energetic

  (repulsive VdW energy) cutoff
- Geometric – “clash grids”

The higher the number of atoms ⇨ more critical efficiency

In Robotics – motion planning

- Collision with obstacles/self-collisions are forbidden
- For kinematic loops - closure constrains exists
- Exploration of the conformations space, searching for feasible conformations
- Sampling-based techniques capture the topology of the feasible space within a data-structure (graph or a tree) by performing random exploration

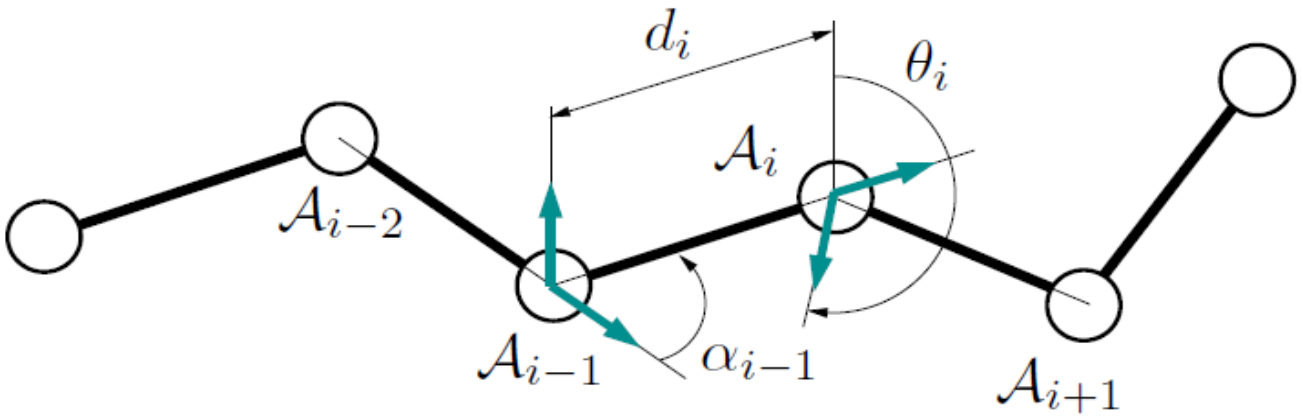
Motivation

- Filter unfeasible loop conformations to aid searching conformational space for various application:
  - Protein loop modeling
  - Molecular simulations: conformational changes under environmental conditions

Problem formulation - outline

- Geometric Model of the molecule
- Geometric Constraints

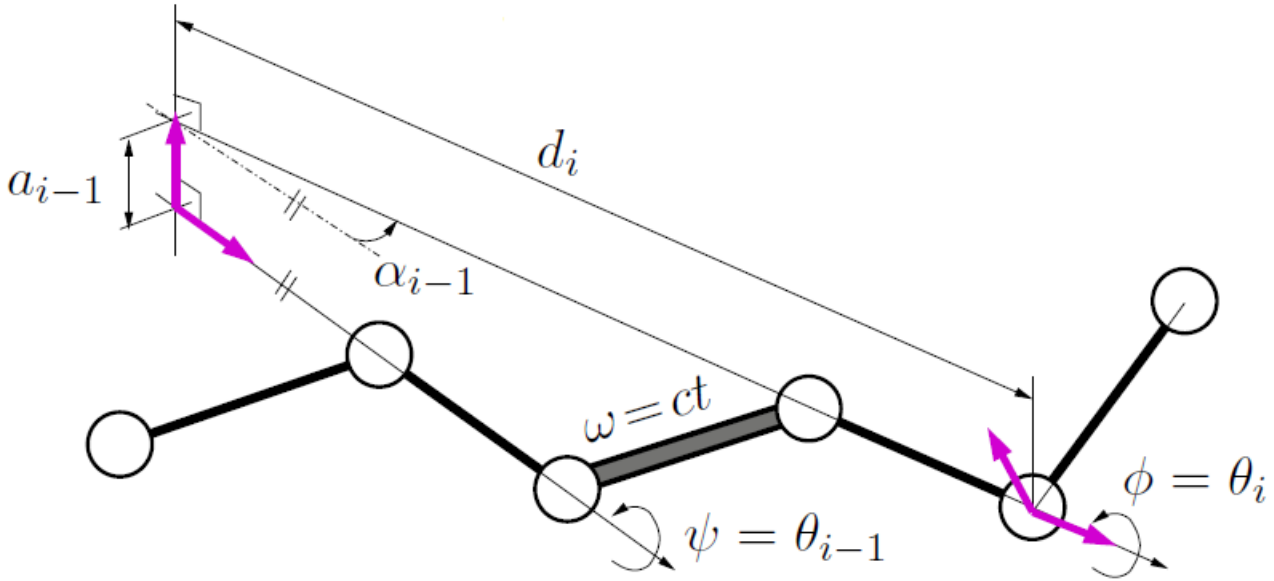
Geometric Model: Molecular chain model



- Molecules – $A_{i-2},\ldots,A_{i+1}$
- Bond length, d_i, between A_{i-1}, A_i
- Bond angle, α_{i-1}, between A_{i-2}, A_{i-1}, A_i
- Dihedral angle, θ_i, formed by $A_{i-2}, A_{i-1}, A_i, A_{i+1}$

Geometric Model: Articulated mechanism

- We assume that only dihedral angles are variable parameters (rigid geometry)



The Homogeneous Transformation Matrix

- We defines the relative location of consecutive frames as follows:

$$ {}^{i-1}T_i = \begin{pmatrix} \mathrm{C}\theta_i & -\mathrm{S}\theta_i & 0 & 0 \\ \mathrm{S}\theta_i\mathrm{C}\alpha_{i-1} & \mathrm{C}\theta_i\mathrm{C}\alpha_{i-1} & -\mathrm{S}\alpha_{i-1} & -\mathrm{S}\alpha_{i-1}d_i \\ \mathrm{S}\theta_i\mathrm{S}\alpha_{i-1} & \mathrm{C}\theta_i\mathrm{S}\alpha_{i-1} & \mathrm{C}\alpha_{i-1} & \mathrm{C}\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

- All parameters as in Molecular chain model that was defined previously
- S - sinus
- C - cosines

Geometric Constraints

- Loop Closure

$$^{0}T_{n} = {}^{0}T_{1}\,{}^{1}T_{2} \ldots {}^{n-1}T_{n}$$

- Clash avoidance(70% of VdW radii)

(Condition must be satisfied between atoms of the articulated segment and between atoms of the rest of the molecule)

Conformational Sampling - outline

- Random Loop Conformation
- Backbone Conformation with closure
- Loop decomposition
- RLG algorithm
- RLG Algorithm: example
- Clashes and Side-Chain Conformation

Random Loop Conformation

Compute random conformation achieving loop-closure and clash avoidance constraints in 3D.

Algorithm 1: RANDOMLOOPCONF

```
input  : the loop, the rest of the protein
output : the conformation q
begin
    q_b ← RANDOMBACKBONECONF(loop.bkb);
    if not CLASHCHECK(q_b, loop.bkb, protein) then
        if q_s ← GENERATESIDECHAINS(q_b, loop, protein) then
            q ← COMPOUNDCONF(loop, q_b, q_s);
        else return Failure;
    else return Failure;
end
```

Array of dihedral angles: $\theta_1,\theta_2,\ldots\theta_n$

A generic 3D collision detection algorithm (T. Siméon, C. van Geem, 2001)

Sample angles randomly at random side-chain order. Check for clashes

Solving multivariable non-linear algebraic equations $^0T_n = {}^0T_1\,{}^1T_2 \ldots {}^{n-1}T_n$

- No efficient general solution exists
- Known – 6 variables are dependent
- How to obtain independent variables satisfy closure equations?

Random Loop Generator (RLG)

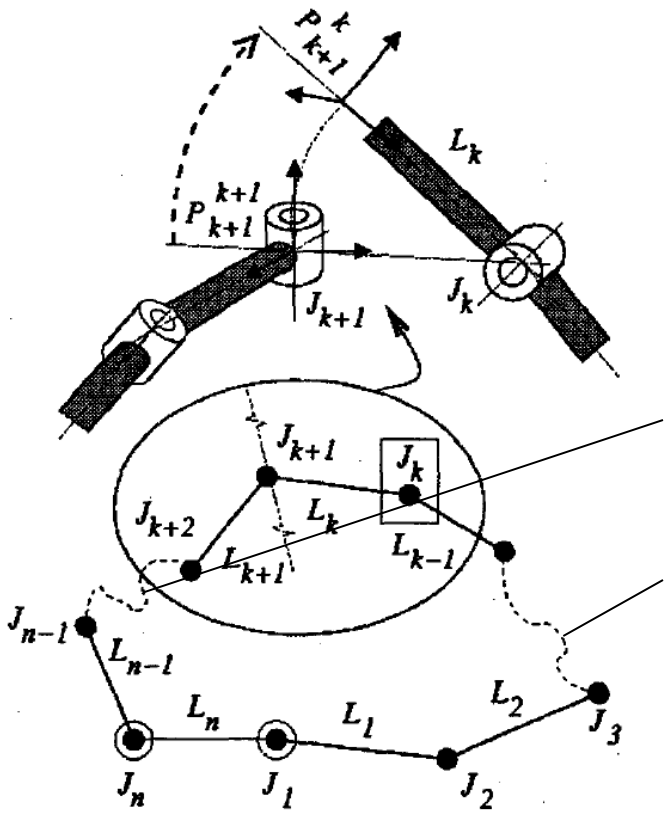


Figure 3: Illustration of the principle of RLG.

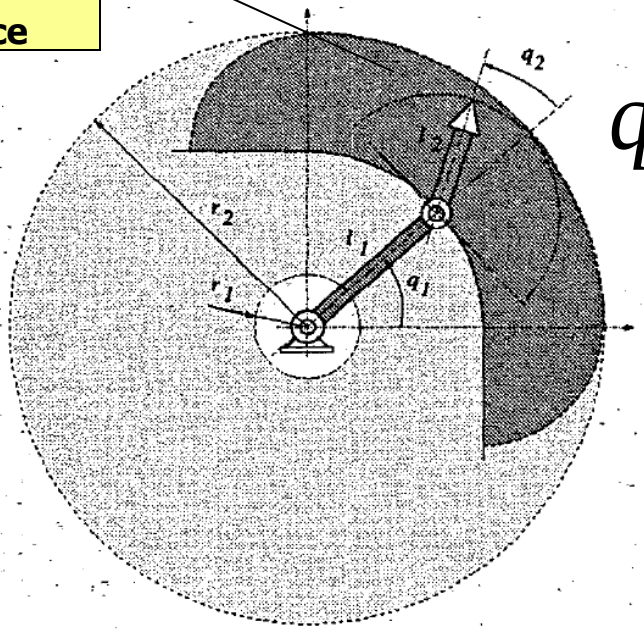
Closure interval of Jk - values of Jk that make possible to close the mechanism are the ones for which P^k_{k+1} is reachable (in position and orientation) by the chain $J_n \rightarrow J_{k+1}$

Passive chain

Active chain

RLG: Approximated reachable Workspace

Reachable workspace



$$q_1 \in [0, \pi/2]; q_2 \in [-\pi/2, \pi/2]$$

Figure 4: Conservative approximation of the reachable workspace of a RR planar chain.

RLG: Closure Interval Computation

Closure interval of Jk

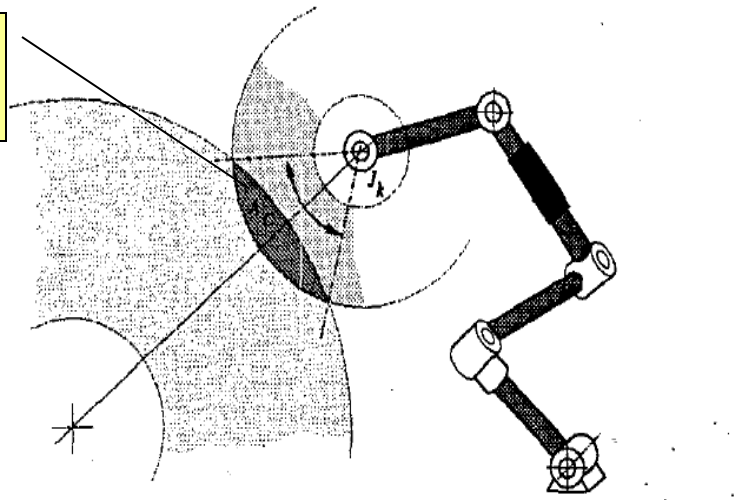
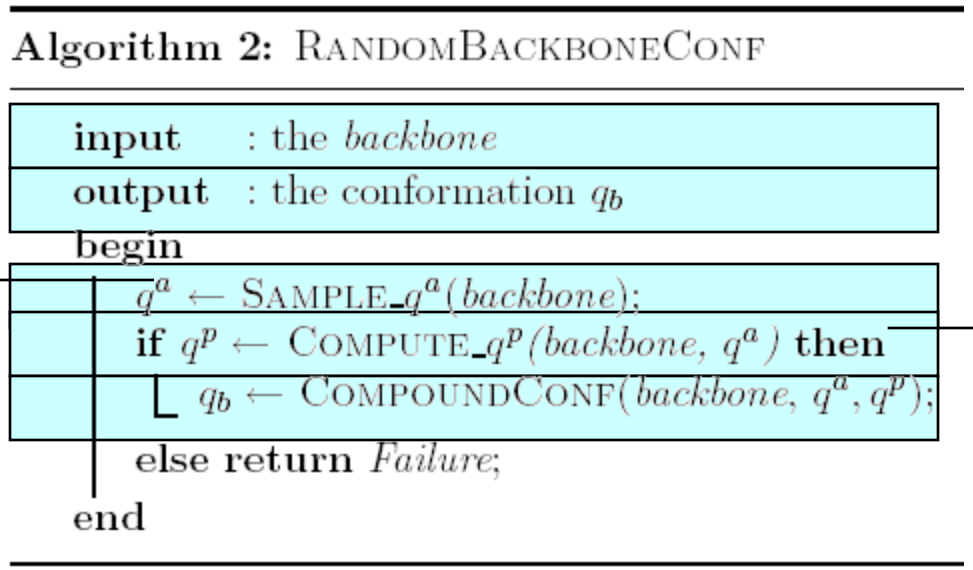


Figure 5: Illustration of the intersections computed for the closure interval estimation in the case of a revolute joint.

RLG Algorithm



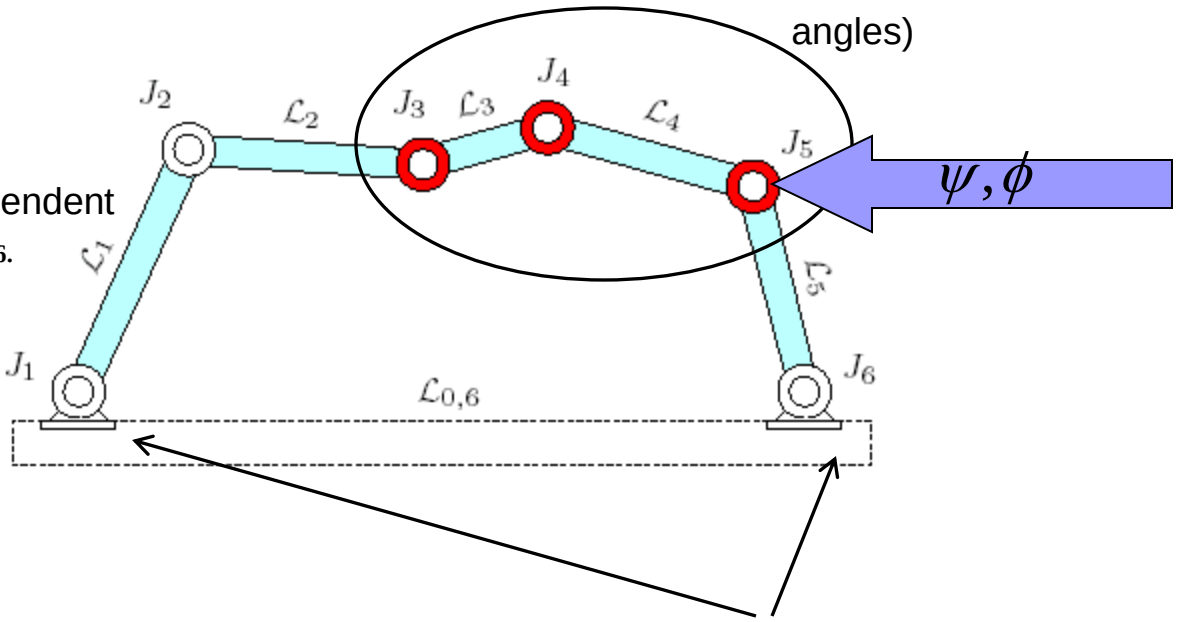
Compute configuration parameters of active sub-chain

A standard inverse kinematics problem

Loop decomposition-6R planar linkage

Passive sub-chain: dependent variables **J_3, J_4, J_5**. (Corresponding to three residues and six dihedral angles)

Active sub-chain: independent variables **J_1, J_2, J_6**.

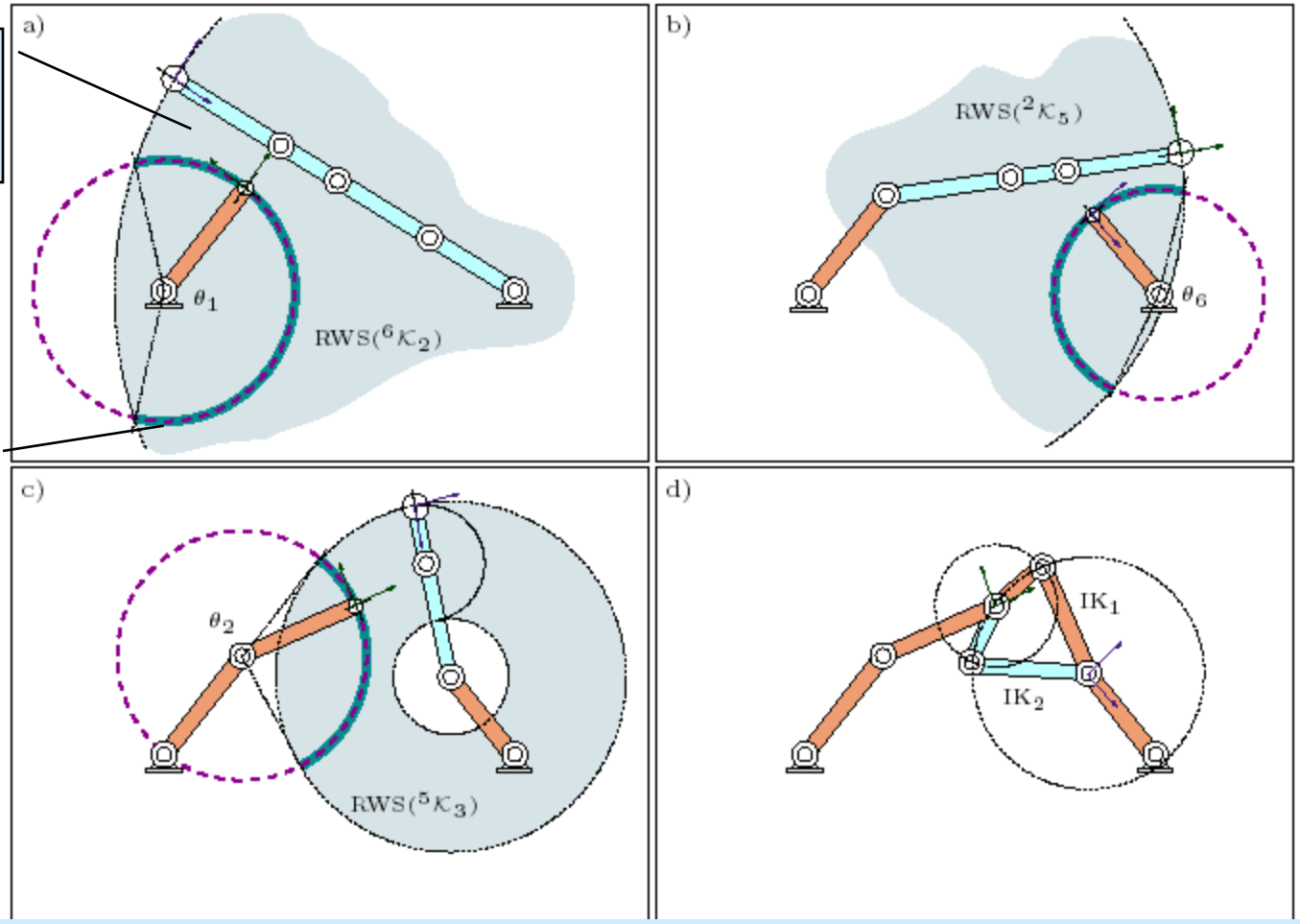


Closed Loop

RLG Algorithm: example

Reachable WorkSpace of Chain6-2

Closure Interval of θ_1



→ Solving the positional-reachable problem is simple and fast approximation to the exact closure range

RLG Algorithm: Backbone Generation

Algorithm 3: SAMPLE_q^a

```
   input  : the backbone
   output : the active variables q^a
   begin
1      (J_b, J_e) ← INITSAMPLER(backbone);
       while not ENDACTIVECHAIN(backbone, J_b) do
           I_c ← COMPUTECLOSURERANGE(backbone, J_b, J_e);
           if I_c = ∅ then goto line 1;
           SETJOINTVALUE(J_b, RANDOM(I_c));
           J_b ← NEXTJOINT(backbone, J_b);
           if not ENDACTIVECHAIN(backbone, J_e) then
               SWITCH(J_b, J_e);
   end
```

Sampling side-chain conformation

- Built upon a feasible backbone conformation
- Generated by randomly sampled dihedral angles
- Tested for collisions
- Only the conformation of clashing side-chains is resampled

Conformational Space Exploration - outline

- Sampling-based Motion Planning Techniques
- RTT Algorithm

Probabilistic RoadMap(PRM) principle

- Graph (roadmap) captures the topology of collision-free configuration
- Nodes are randomly sampled
- Edges are short feasible paths linking “nearby” nodes

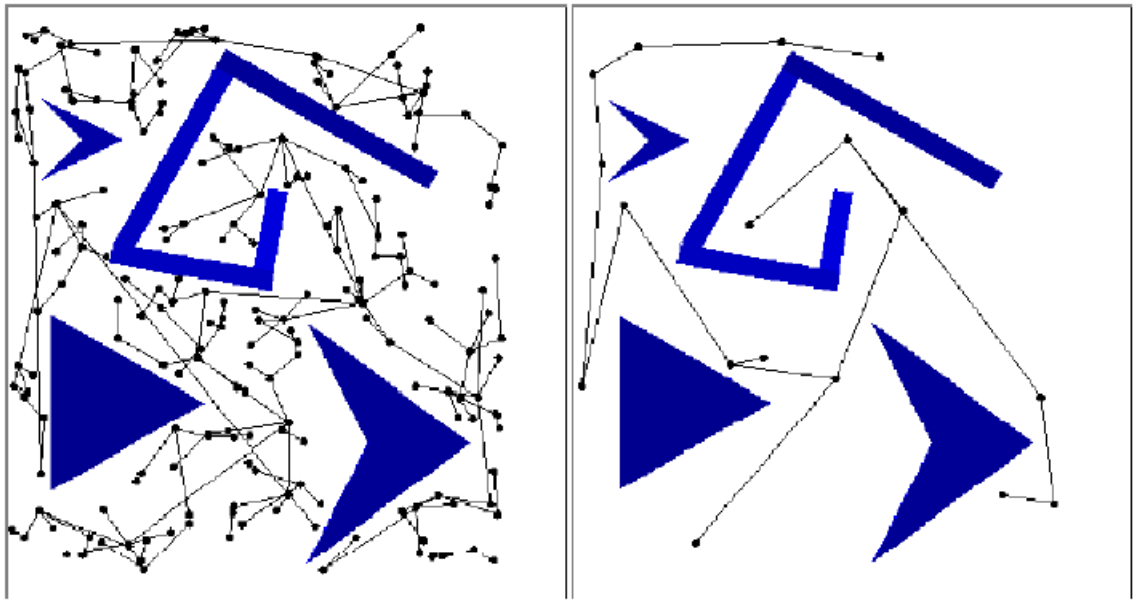
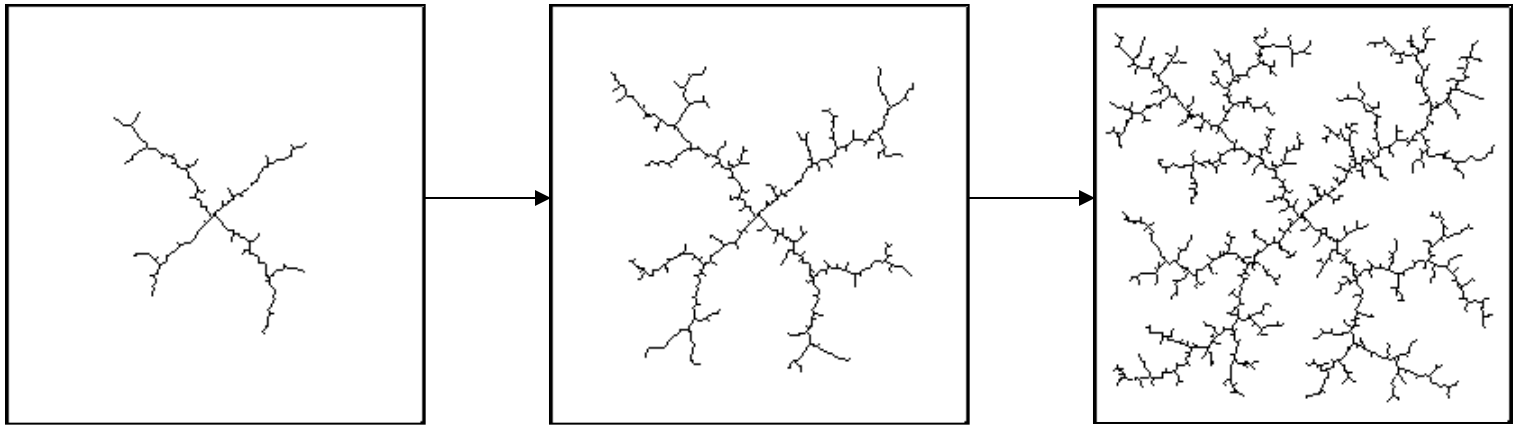


Figure 2: Basic-PRM and visibility-PRM in the same 2D environment.

Rapidly Exploring Random Tree (RRT)

developed by M.LaValle and J.Kuffner



Properties:

- Expands quickly
- Unbiased relative to random walk.
- Vertices are uniformly distributed
- Short paths
- Quickly searches high-dimensional spaces

We search for….

$$C_{lowE} \subset C_{feas} \subset C$$

$$C_{feas} = C_{clos} \cap C_{free}$$

Assumption: All energetically feasible configurations are considered

RTT Algorithm

Algorithm 4: ExploreByRRT

```
input  : the loop, the rest of the protein, q_init
output : the tree T
begin
    G ← InitTree(q_init);
    n_fail ← 0;
    while not StopCondition(T) do
        q_rand ← GuidedRandomConf(loop);
        q_near ← NearestNeighbor(q_rand, T);
        q_feas ← q_near;
        state ← OK;
        while state = OK do
            q_step ← MakeStep(q_feas, q_rand);
            if FeasibleConf(q_step) then q_feas ← q_step;
            else state ← FAIL;
        if not TooSimilarConf(q_near, q_feas) then
            q_new ← IntermediateConf(q_near, q_feas);
            GrowTree(q_new, q_near, T);
            n_fail ← 0;
        else n_fail ← n_fail + 1;
end
```

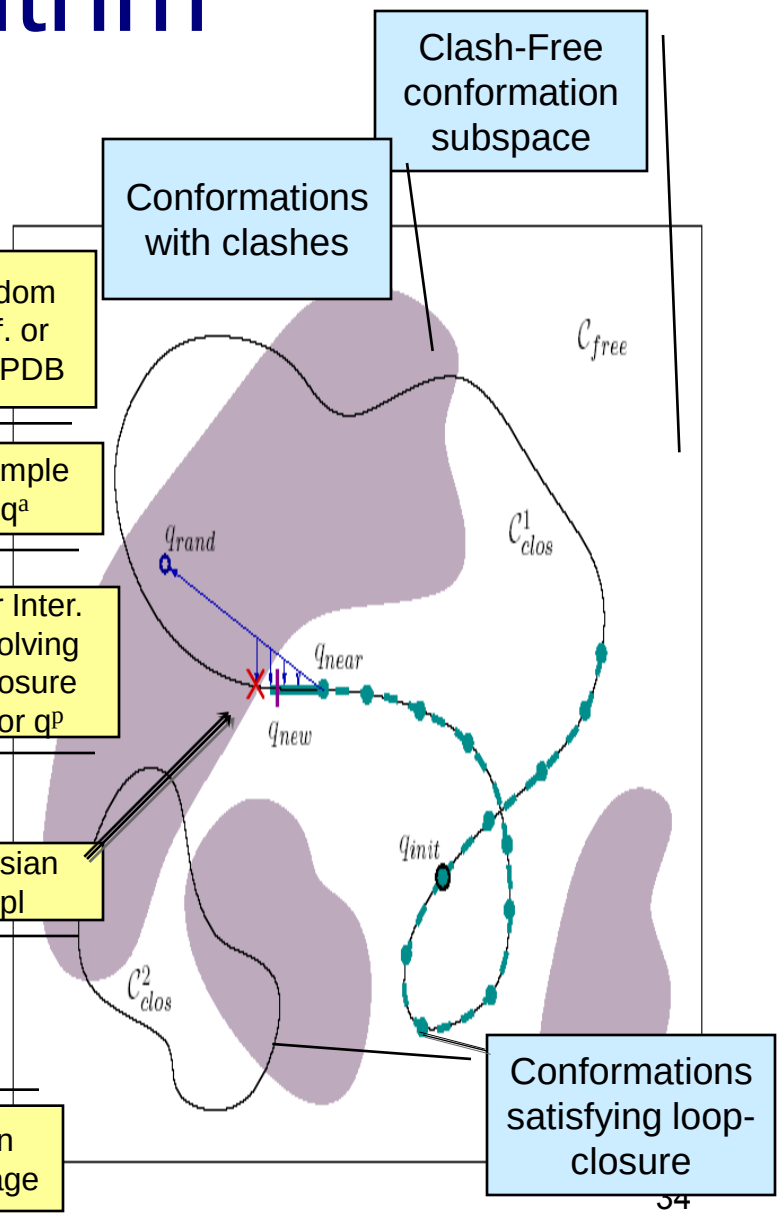
Random conf. or from PDB

Sample q^a

Linear Inter. and solving the closure eq. for q^p

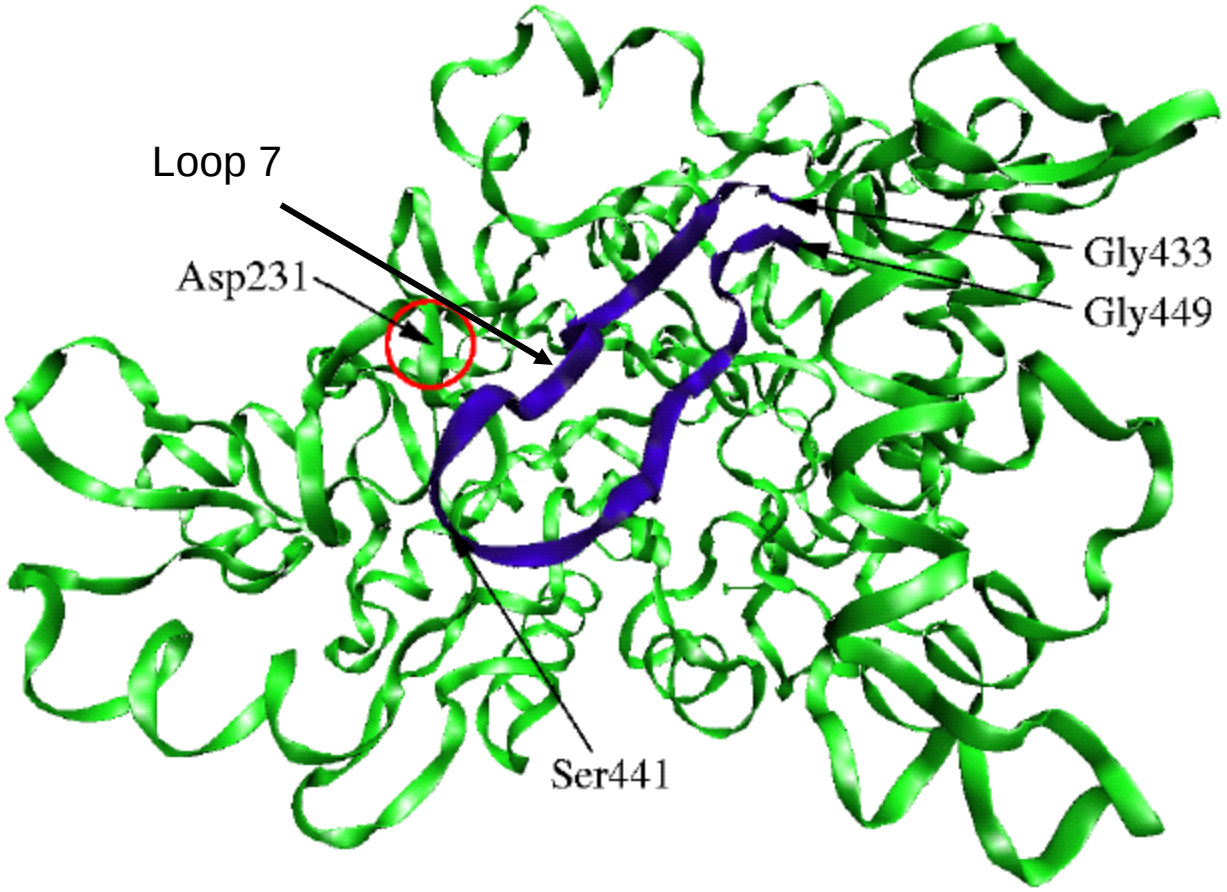
Gaussian smpl

Believed to be an estimate to coverage

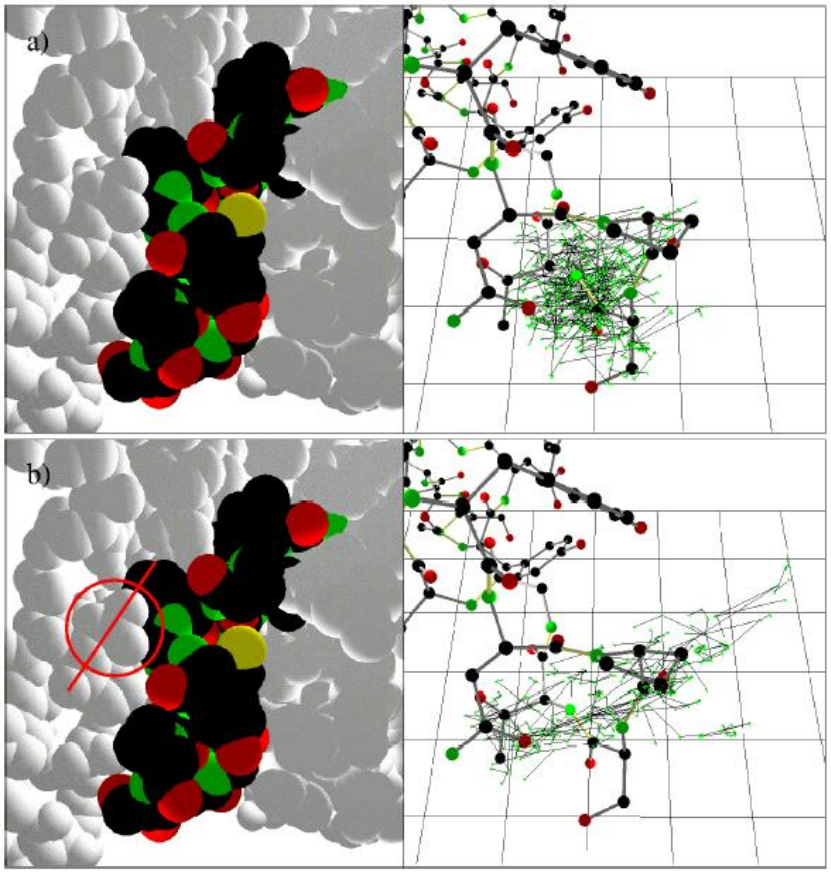


Results-structure of Amylosucrase

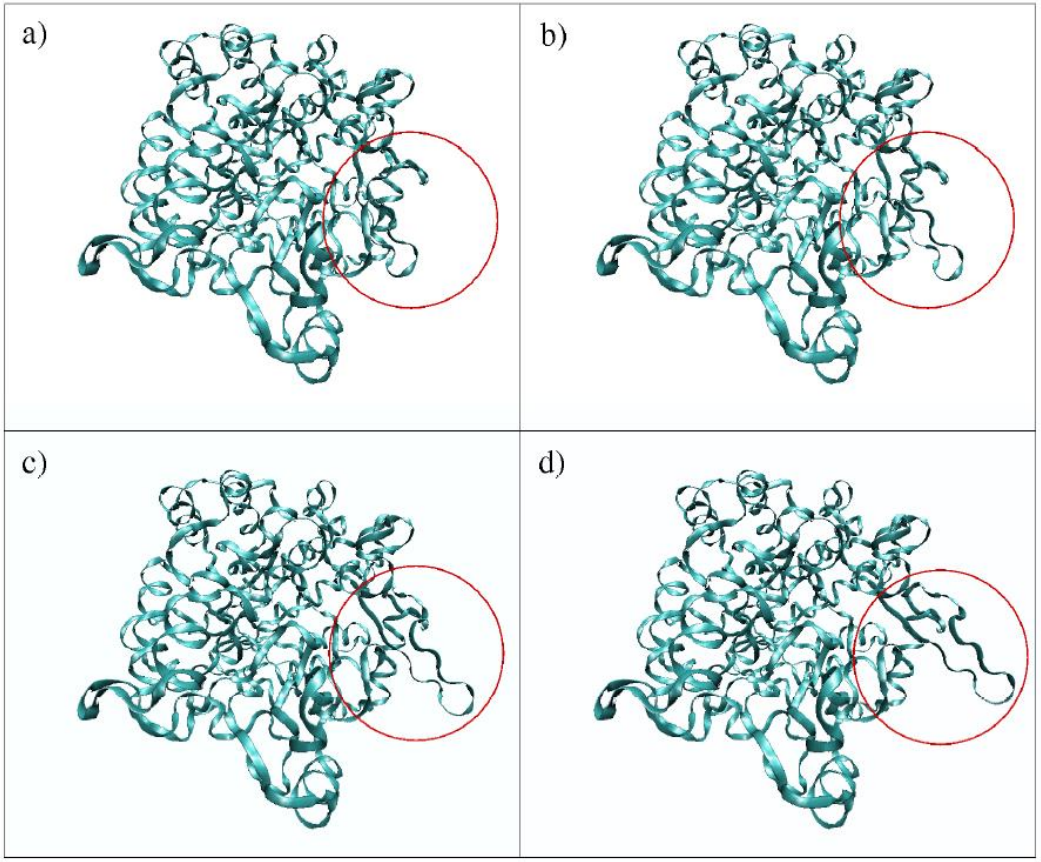
Motion of Loop 7 may have a pivotal rule in facilitating molecules interactions.



Results-exploration with/without side chain of Asp



Results-simulated conformational change in loop 7



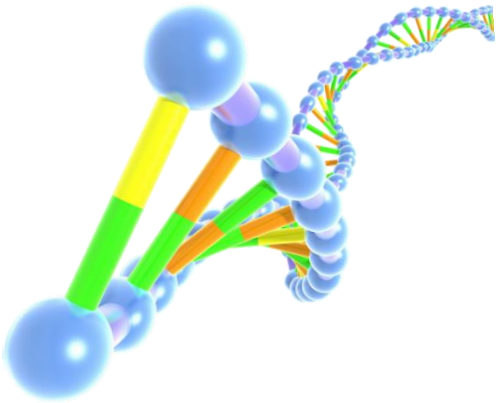
Conclusions

- Filter technique for conformational sampling and search methods was proposed
- Solution to the loop closing problem is computationally efficient
- RLG computes exact closure solutions while CCD outputs approximated solutions

Prospects

- Tailor a collision detection algorithm for the molecular application (Collision detection is by far the most computation expensive task)
- Check clashes after each stage of RLG between backbone & static env.
- Avoid steric clashes by choosing statistically preferred values for dihedral angles
- Incorporate energetic analysis (constraints) into the incremental search technique

Questions ?



Thank You

Polypeptide Extension (approximation)

l_π – length of polypeptide chain when all the dihedral angles at π.

$\tilde{l}$ – upper bound on the chain's length. It is the sum of the distances between consecutive C_α atoms.

The extension of a chain is randomly sampled from a distribution between l_π and $\tilde{l}$.