A FILTERING TECHNIQUE FOR FRAGMENT ASSEMBLY- BASED PROTEINS LOOP MODELING WITH CONSTRAINTS

F. Campeotto[1,2] *A. Dal Palù*[3] A. Dovier[2] F. Fioretto[1]
E. Pontelli[1]

1. Dept. Computer Science, NMSU

2. Dept. Math and Computer Science, Univ. Udine

3. Dept. Math and Computer Science, Univ. Parma

CP'12

Summary

Background

Joined Multibody Constraint

NP-completeness

Filtering

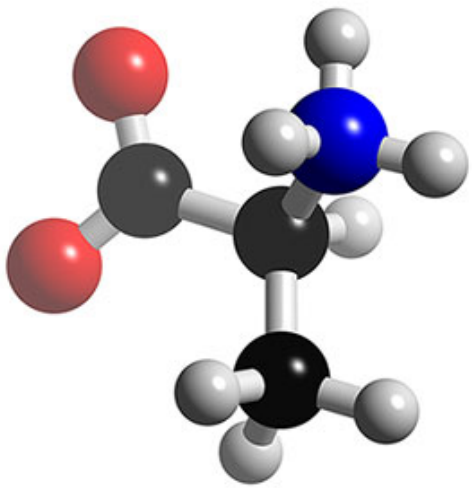
Results

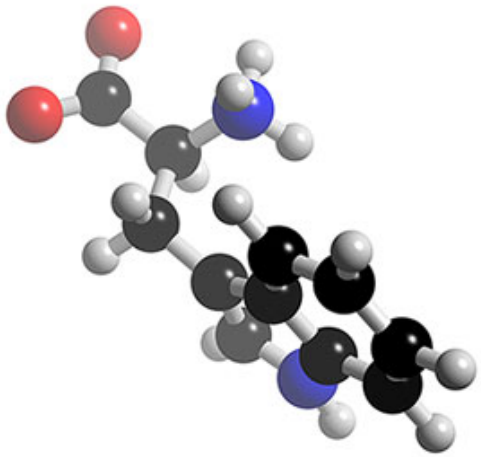
Conclusions

PROTEINS

AMINO ACIDS

- Proteins are molecules made of amino acids



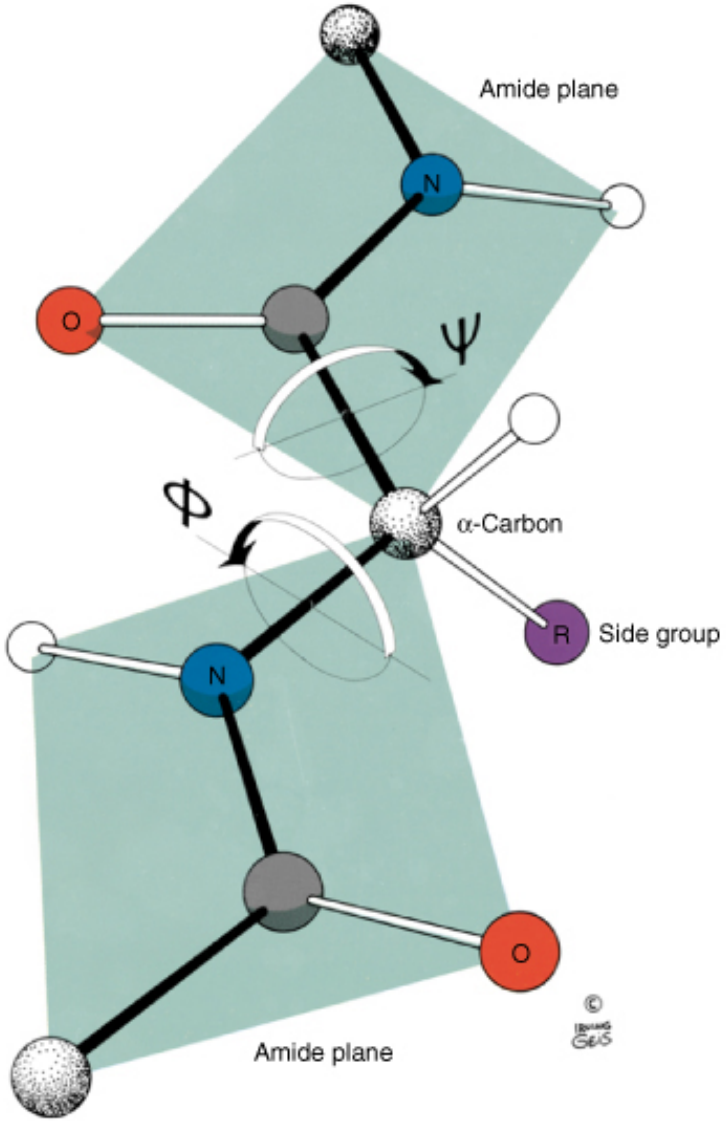


$C_3H_7NO_2 \rightarrow$ 13 atoms $\quad C_{11}H_{12}N_2O_2 \rightarrow$ 27 atoms

- Amino acids show different chemical properties depending on the specific atoms

PROTEINS

PEPTIDE BOND



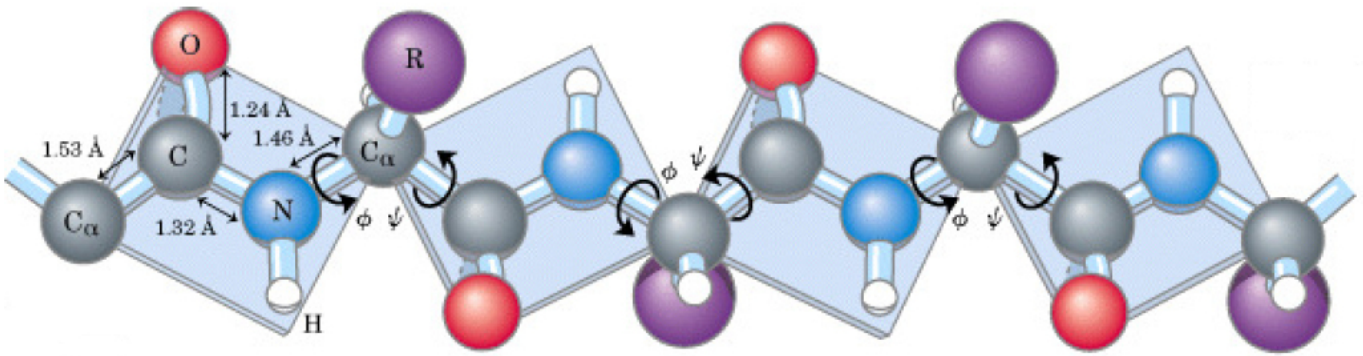
- When connected together, amino acids can rotate along some bonds (Φ and Ψ angles).
- Rotation is rigid and the angles domains are continuous in nature.
- Some angles are forbidden because of atoms clashes

Figure copyrighted by Irving Geis.
Copyright 1999 John Wiley and Sons, Inc. All rights reserved.

PROTEINS

BACKBONE

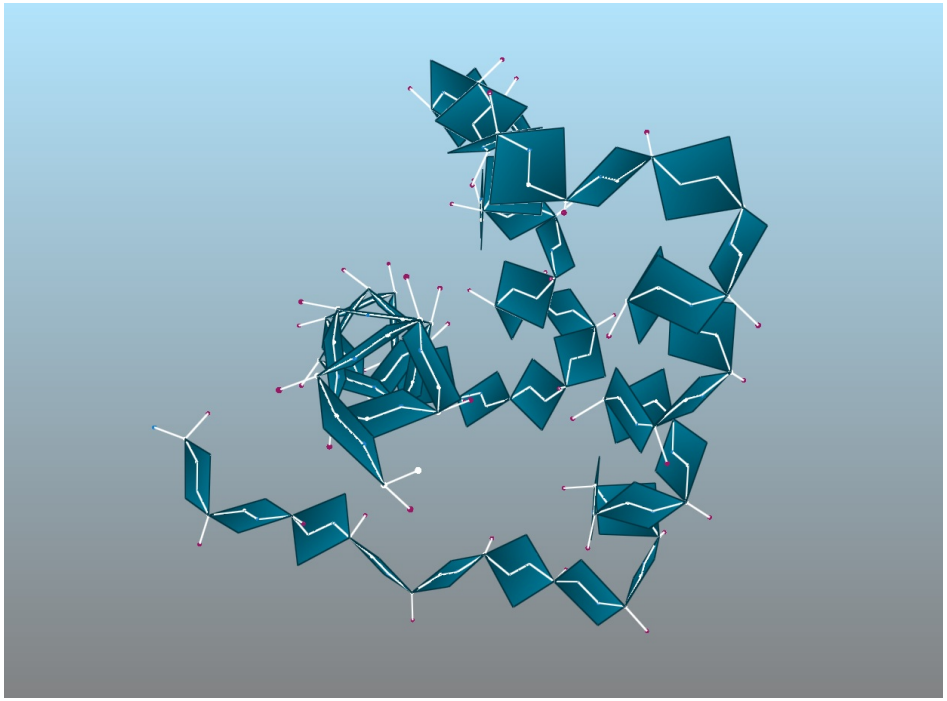
- Combining n amino acids gives at least $2n$ degrees of freedom molecule
- Constraints on angles *and* on atoms positions
- It can be modeled as an extension of a Self Avoiding Walk



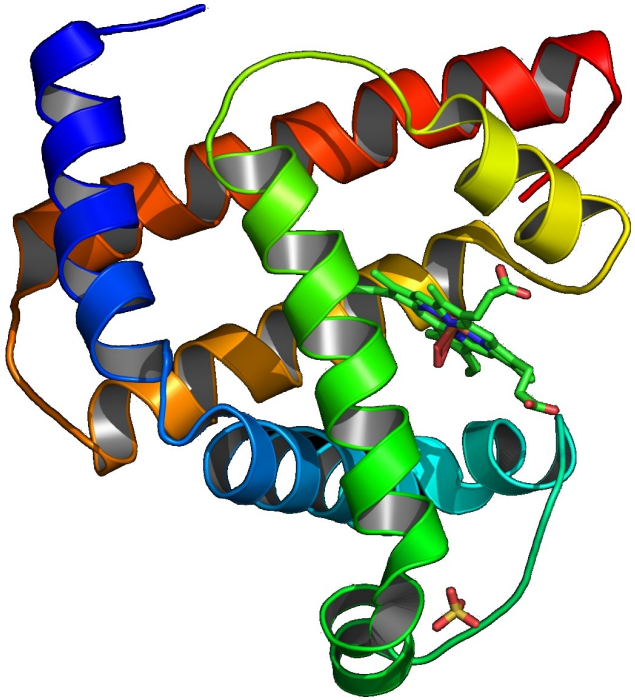
PROTEINS

TERTIARY STRUCTURE

Amino acid primary sequence:
GPEILCGAELVDALQFVCGDRGFYFNKPTGYGSSS...



planes...



... tertiary structure

- Proteins FOLD spontaneously into their functional pose

PROTEINS

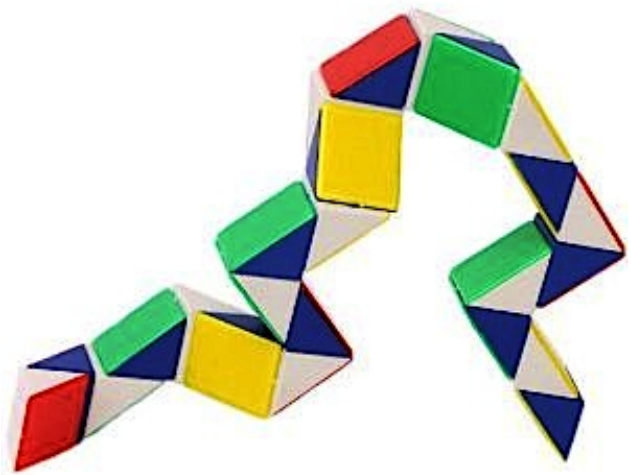
OUR CONTRIBUTION

- *Finite Domains* and *non linear* constraints model
- Goal: Fast geometrical exploration (independent of energy)
- Introduce *approximated propagators* for 3D points

PROTEINS

OUR MODEL

- Each torsional angle is associated to a discrete set of angles
- Chemists are happy with a large set ($\sim$ 100 sampled rotations per bond)
- Combinatorial explosion: 100^{2n}

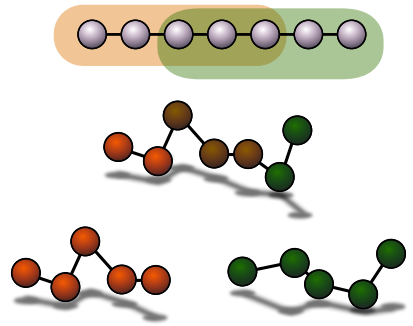


A simple instance of discretized angles

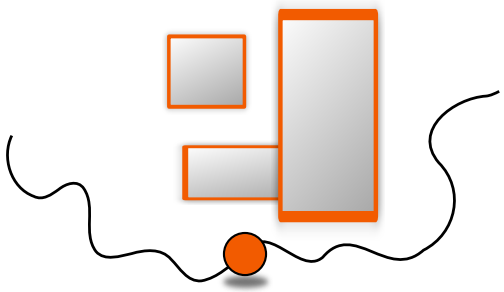
PROTEINS

CONSTRAINTS

1. Backbone structural constraints

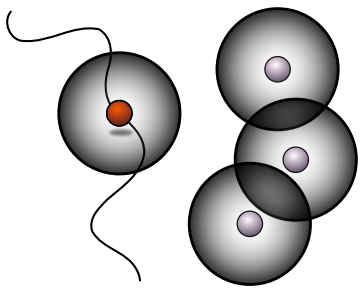


2. Amino acids local relationships



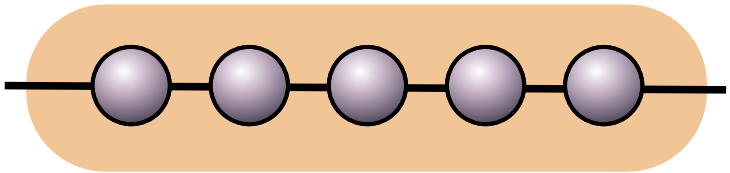
3. Spatial positions of atoms

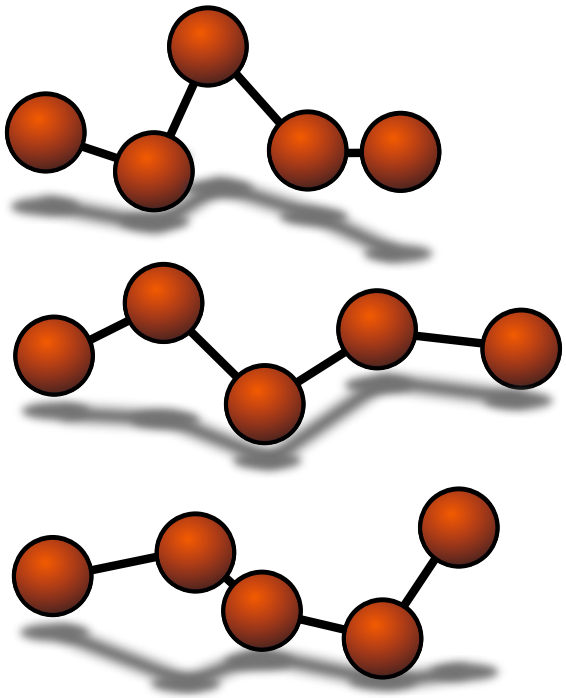
4. Non overlap of atoms



CONSTRAINTS

FRAGMENTS AND BODY





- Local structural behavior $\Rightarrow$ fragments
- A set of fragments (Body) is a *local spatial dictionary*

FRAGMENTS

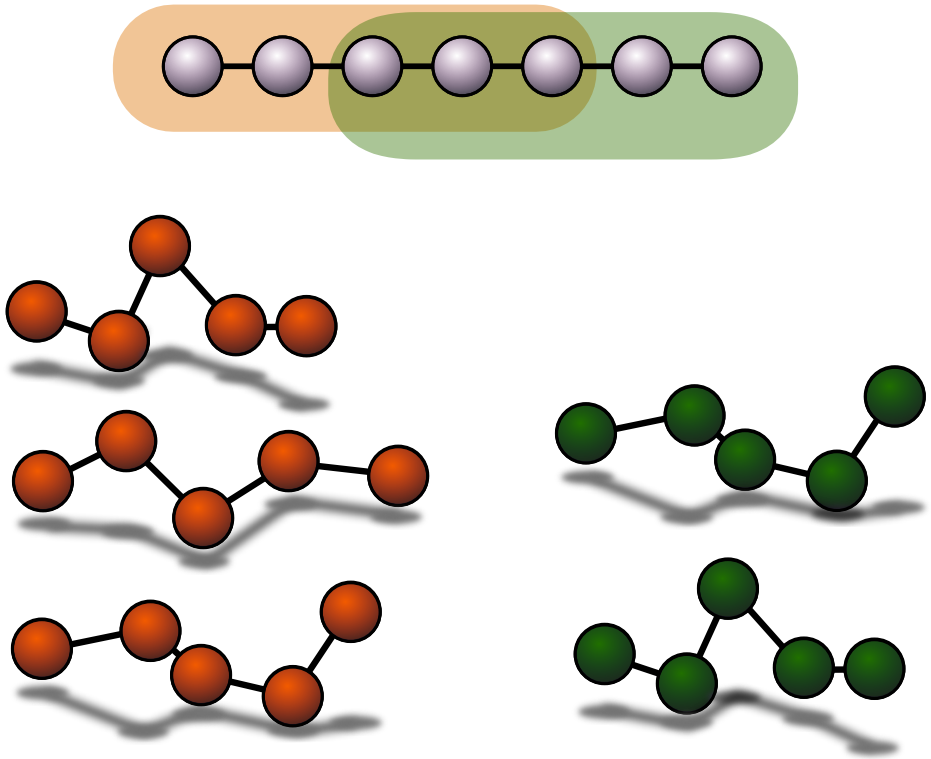
BODY

- Body = FD variable, Fragments = Domain
- A body can be associated to 1 .. n amino acids
- It can be seen as a constraint over Φ and Ψ angles
- It is similar to a `table` constraint over angles
- It affects 3D positions of atoms

FRAGMENT ASSEMBLY MODEL

BODY COMBINATION

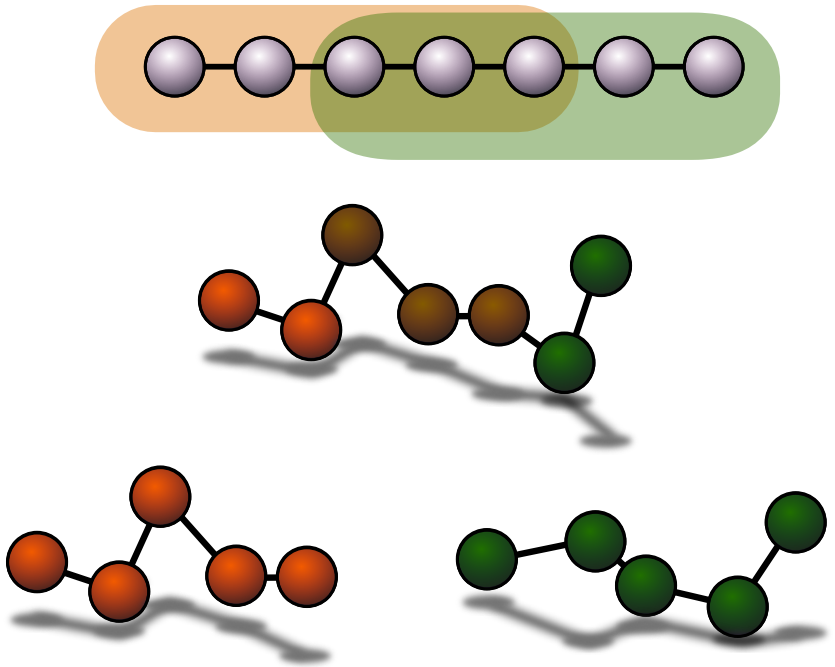
Example of domains for two overlapping bodies



FRAGMENT ASSEMBLY MODEL

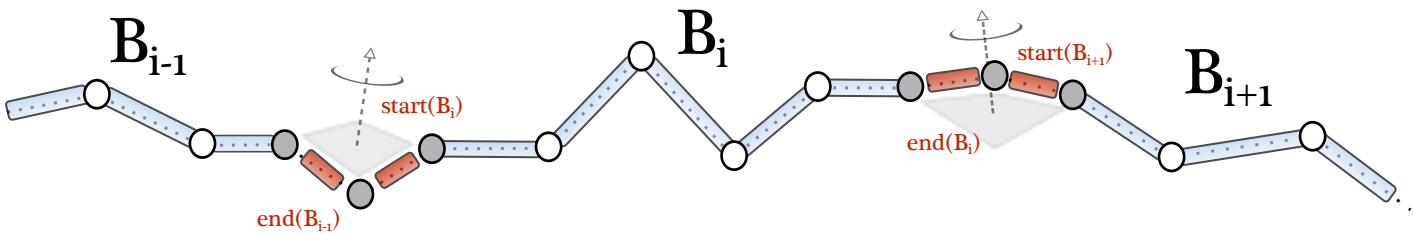
BODY COMBINATION

Constrain two selected fragments:
they must overlap in the space at anchors



MULTIBODY

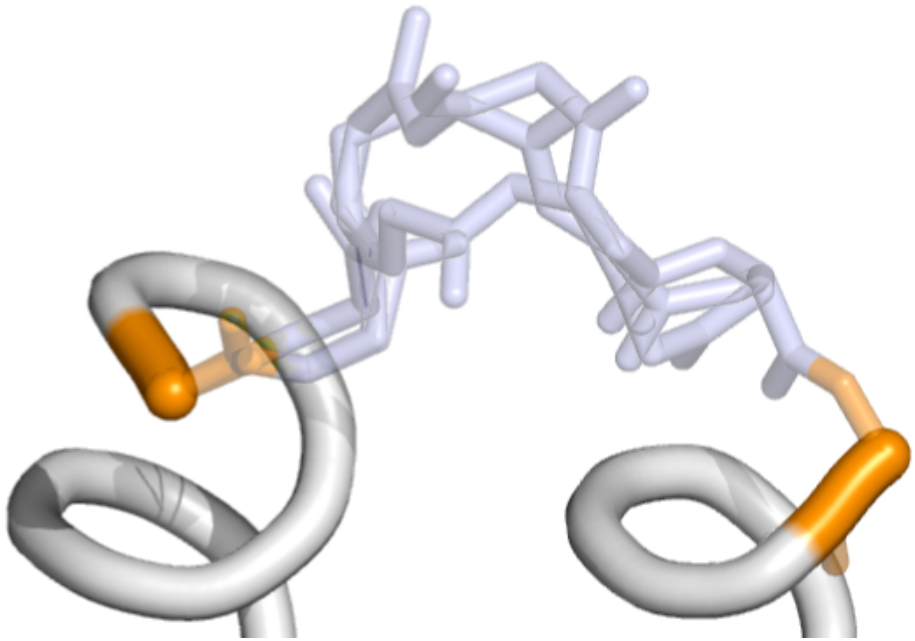
- A Multibody is a set of Bodies that covers a protein
- The protein is partitioned into Bodies that overlap at the anchors



MULTIBODY

MORE CONSTRAINTS

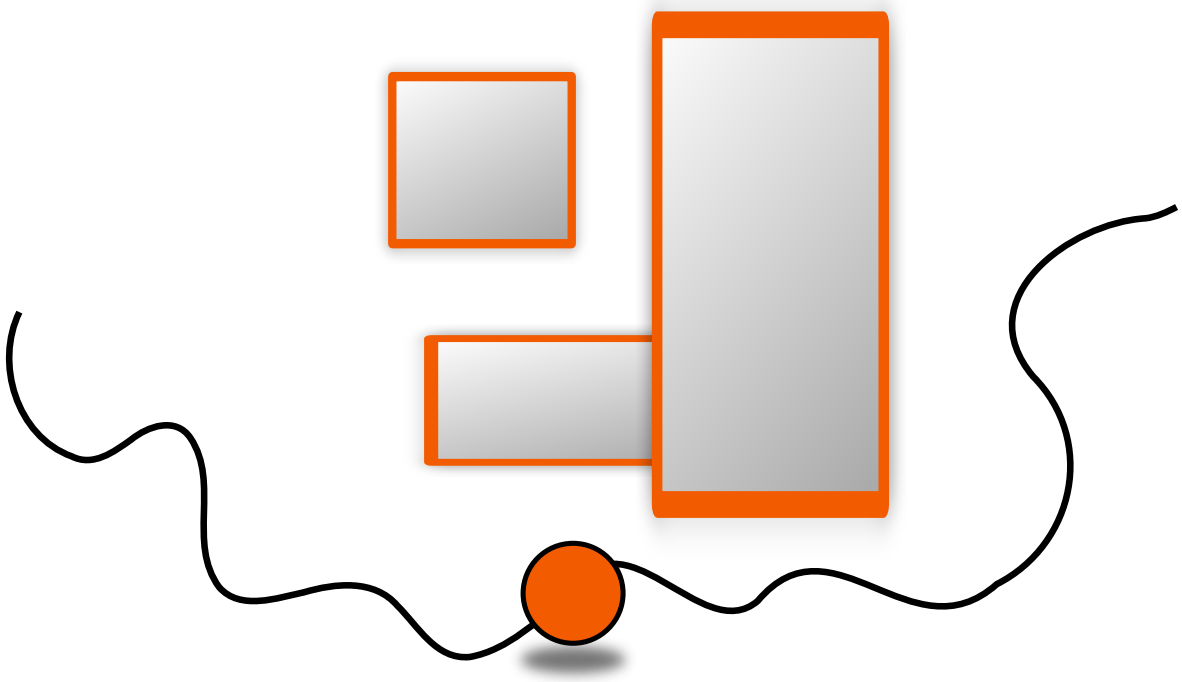
- Restricted spatial positions for some anchors
- Convenient modelling of accurate mobility (eg. loops, flexibility, ligand interactions, typical subunits arrangements)



MULTIBODY

SPATIAL CONSTRAINTS

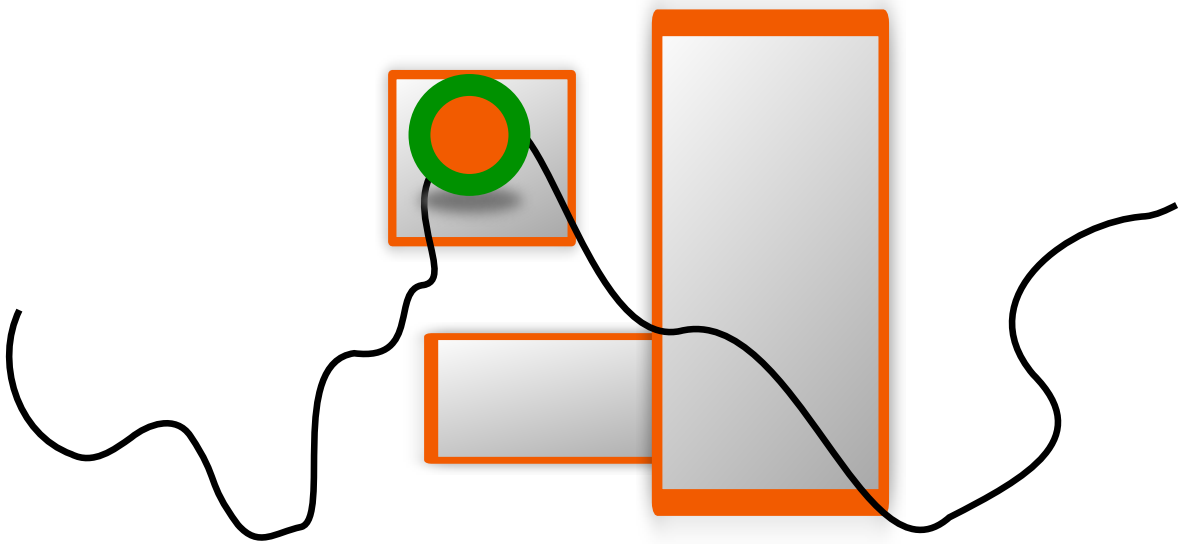
Each atom can be constrained to a specific region



MULTIBODY

SPATIAL CONSTRAINTS

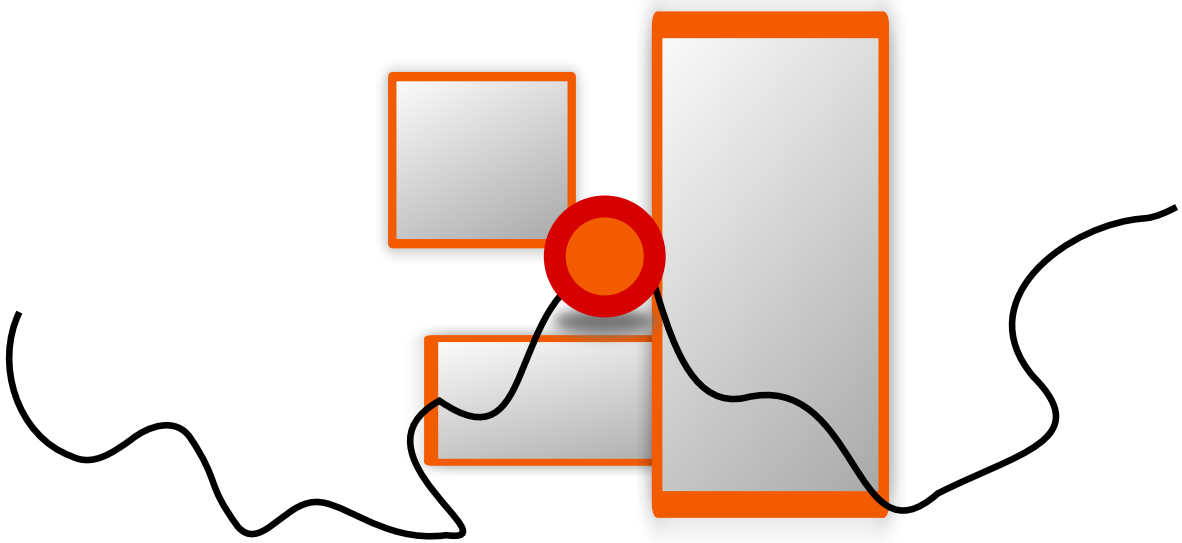
Each atom can be constrained to a specific region



MULTIBODY

SPATIAL CONSTRAINTS

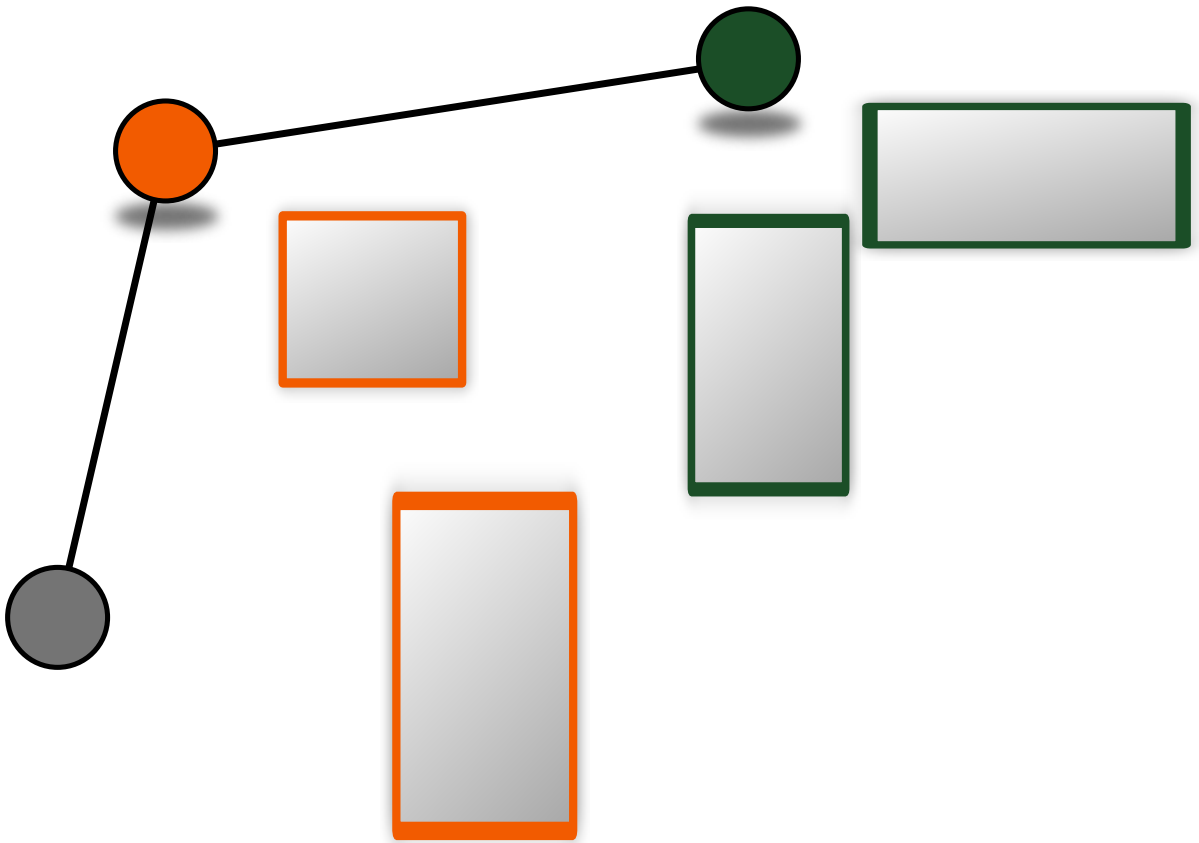
Each atom can be constrained to a specific region



MULTIBODY

SPATIAL CONSTRAINTS

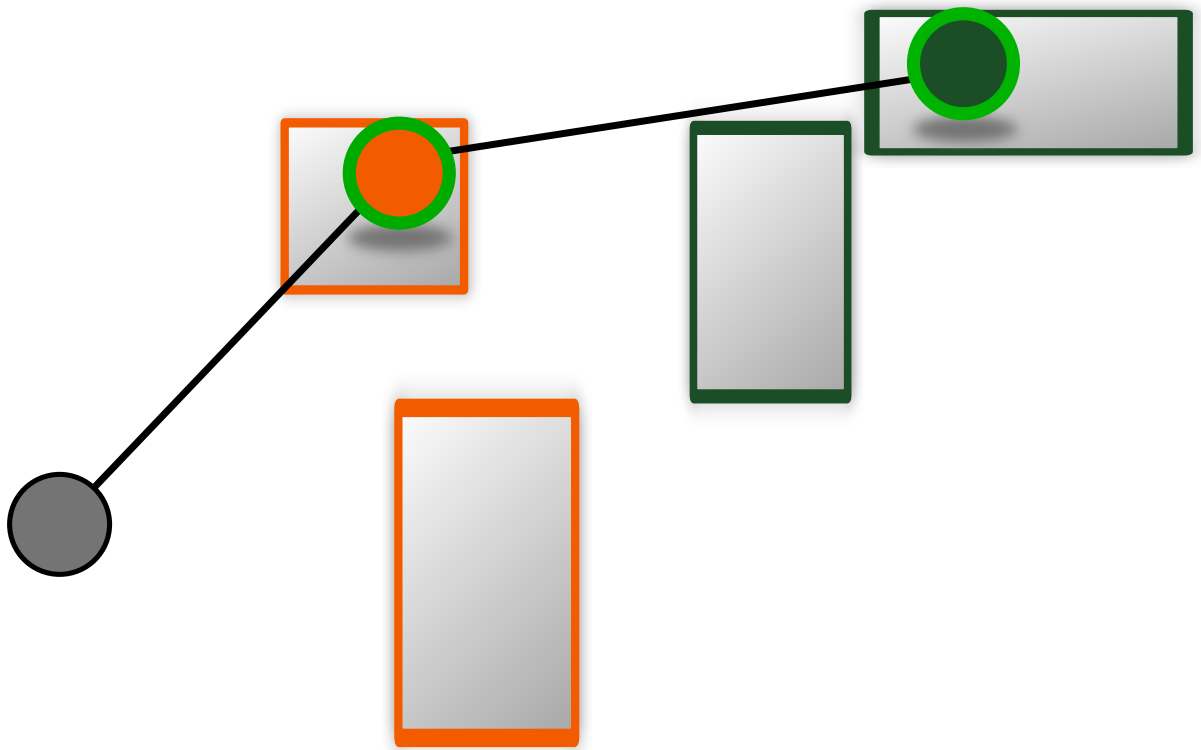
Spatial constraints on different atoms may interact through fragments



MULTIBODY

SPATIAL CONSTRAINTS

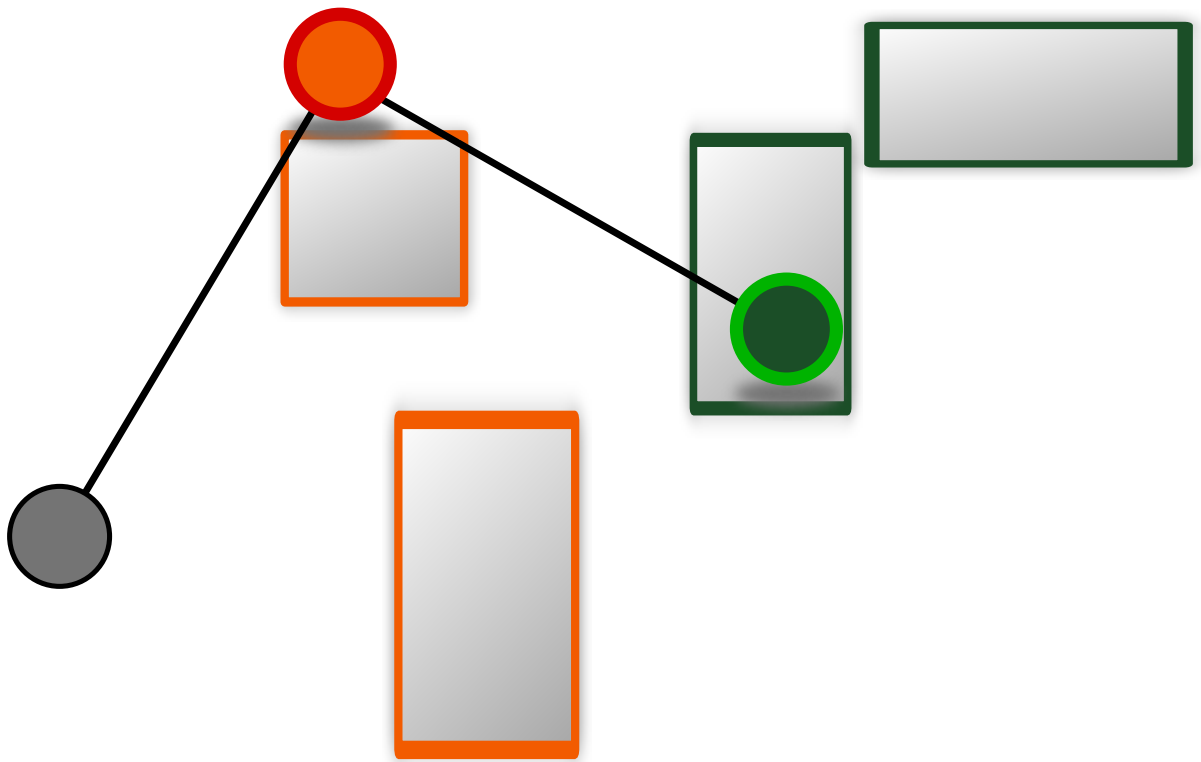
Spatial constraints on different atoms may interact through fragments



MULTIBODY

SPATIAL CONSTRAINTS

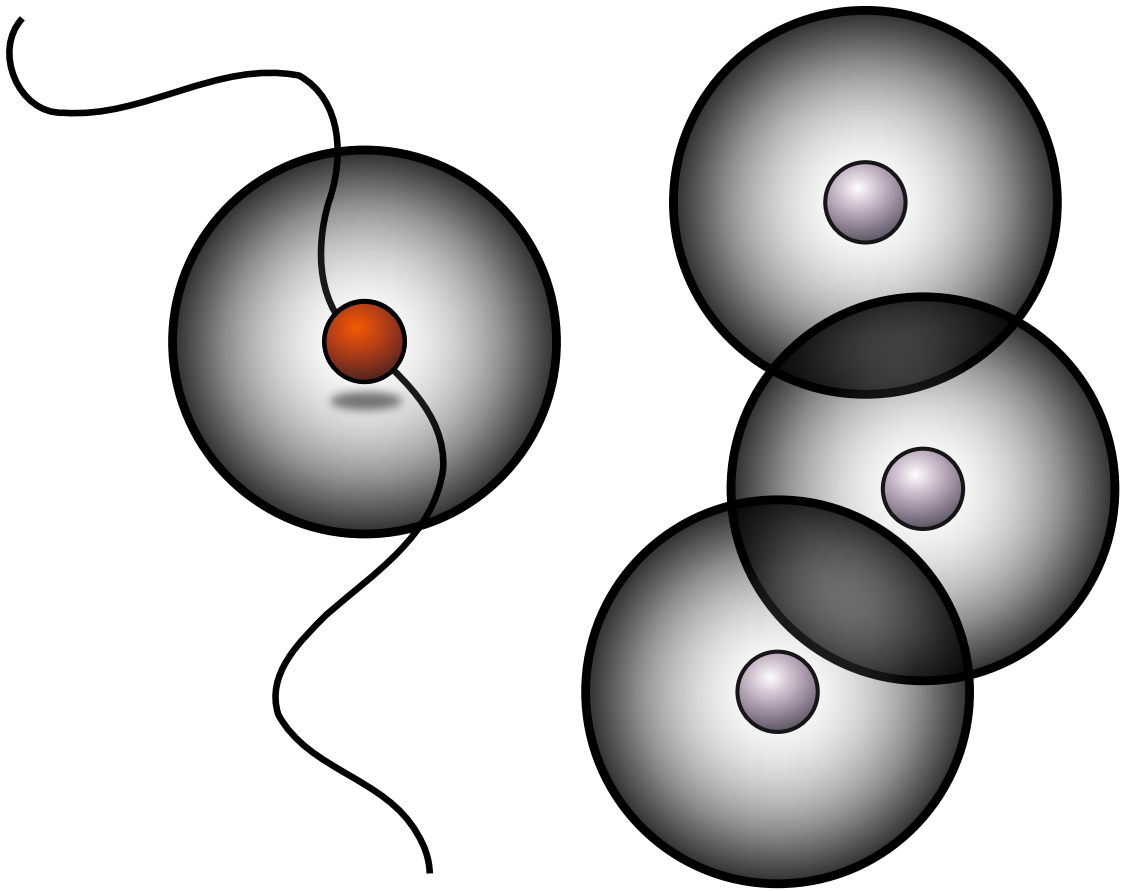
Spatial constraints on different atoms may interact through fragments



MULTIBODY

DISTANCE CONSTRAINTS

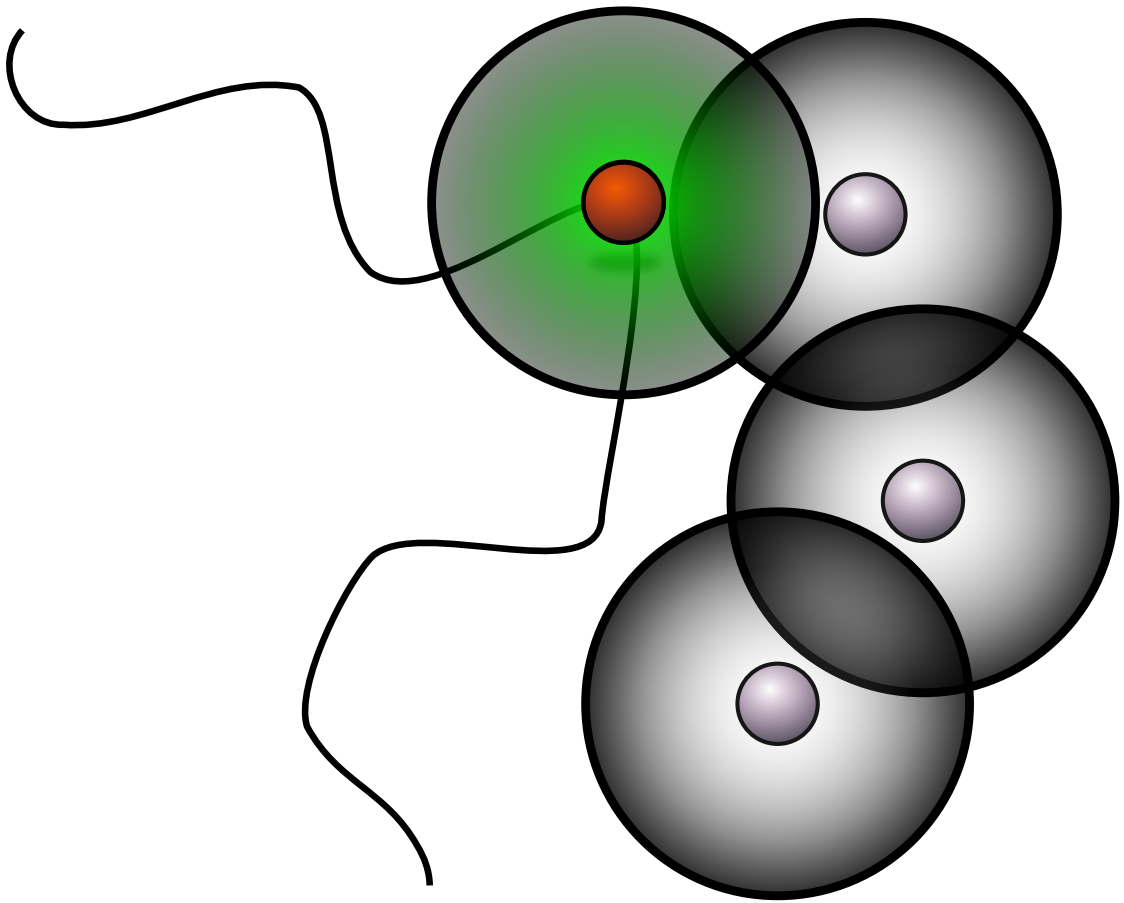
Minimal distance between pairs of atoms



MULTIBODY

DISTANCE CONSTRAINTS

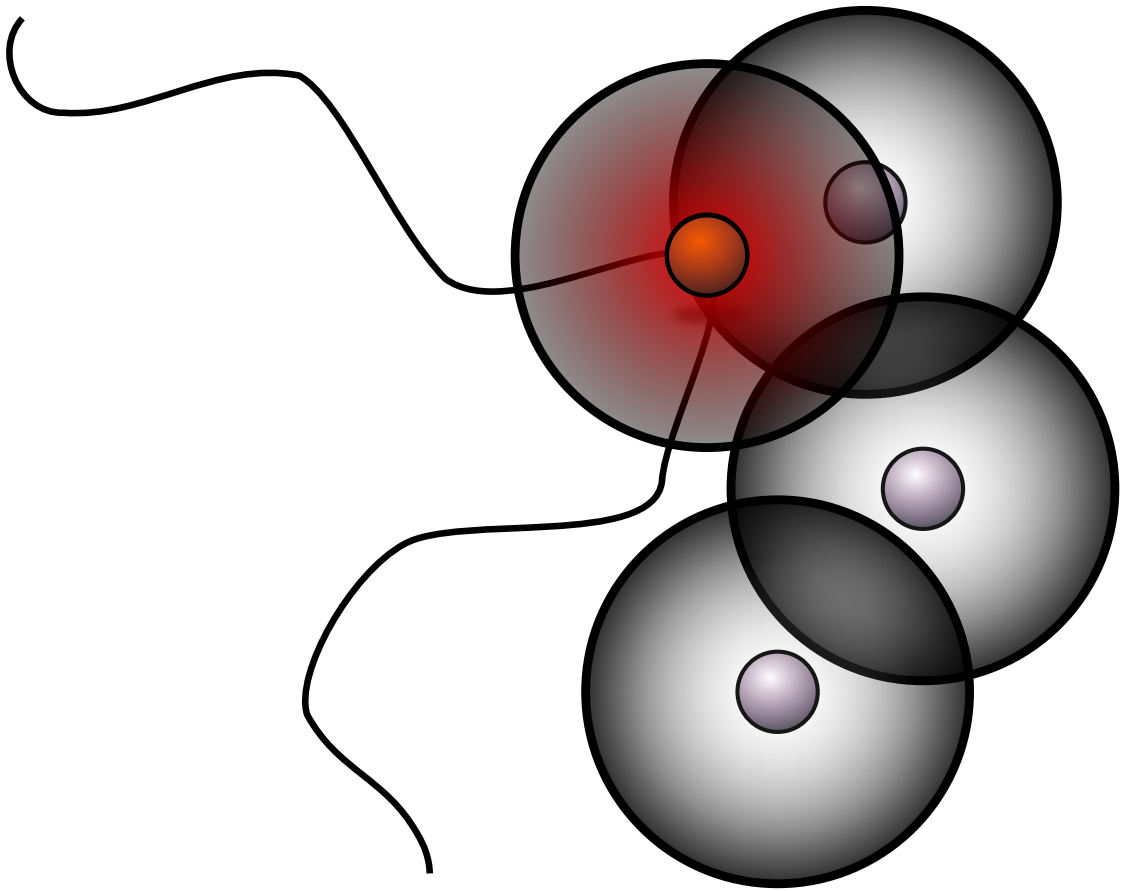
Minimal distance between pairs of atoms



MULTIBODY

DISTANCE CONSTRAINTS

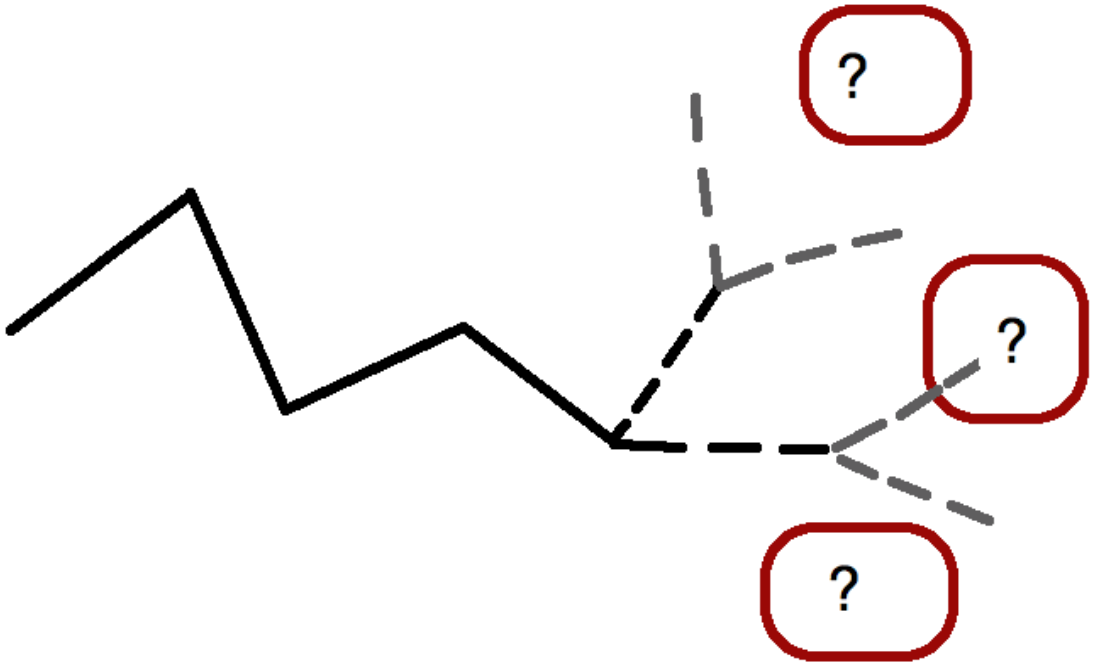
Minimal distance between pairs of atoms



THE JOINED-MULTIBODY (JM) CONSTRAINT

INTUITION

- Given a set of fragments for each body
- Given a set of admissible atom volumes
- Given a minimal inter-atomic distance
- We want to identify the fragments combinations that satisfy all the constraints



THE JOINED-MULTIBODY (JM) CONSTRAINT

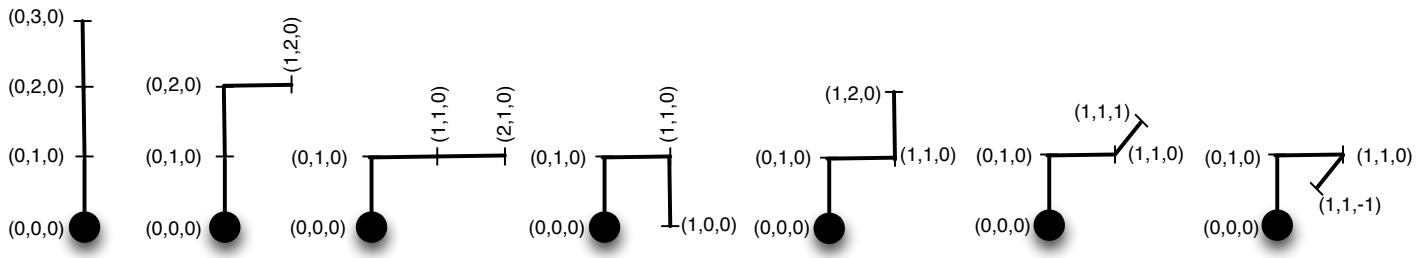
COMPLEXITY

THEOREM

The problem of determining consistency of JM constraints (i.e., the existence of a solution) is NP-complete.

PROOF.

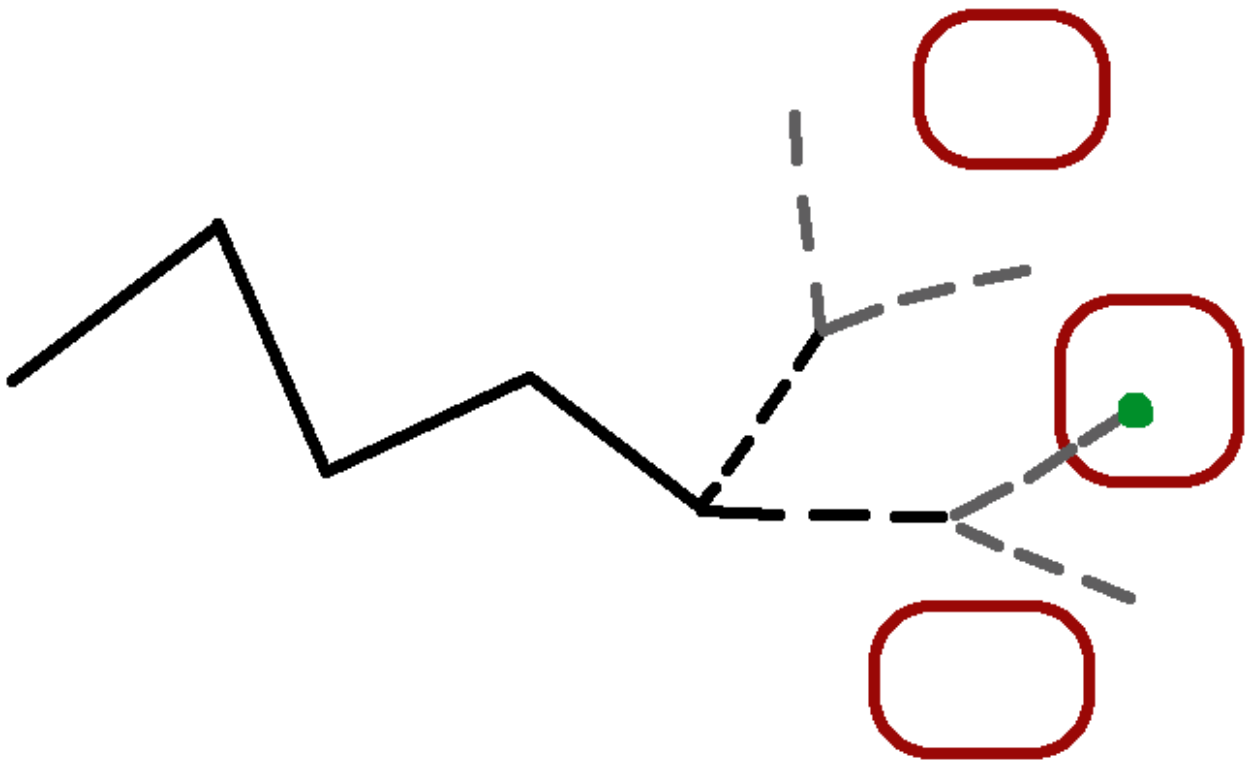
Sketch: reduction from Self Avoiding Walk on 3D cubic lattices. Idea: create a set of discrete fragments and force them on a lattice. □



FILTERING ALGORITHM

APPROXIMATED POLYNOMIAL TIME ALGORITHM

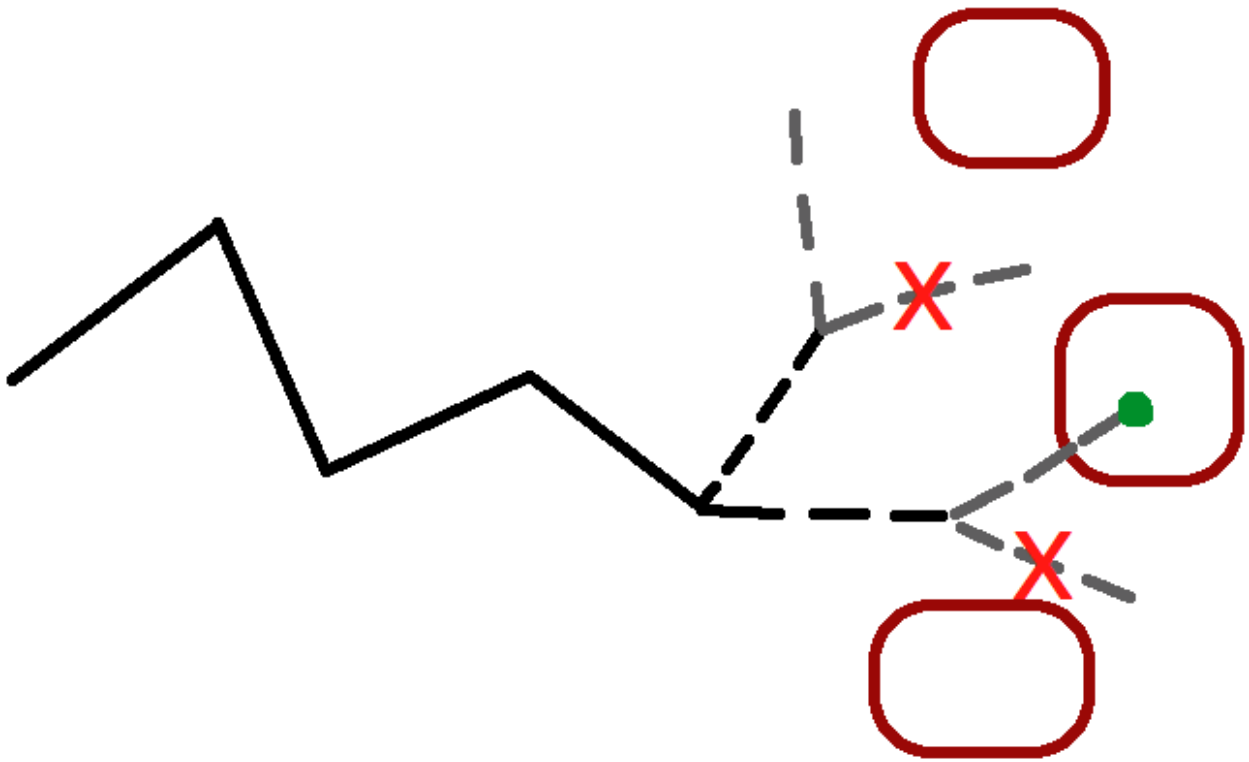
- Identify those fragments in each body that are of no support
  1. Need a fast approximation of reachable volumes
  2. Intersect it with JM spatial domains
  3. Filter non supported fragments



FILTERING ALGORITHM

APPROXIMATED POLYNOMIAL TIME ALGORITHM

- Identify those fragments in each body that are of no support
  1. Need a fast approximation of reachable volumes
  2. Intersect it with JM spatial domains
  3. Filter non supported fragments



FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

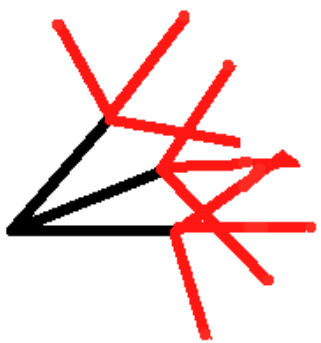
ALMOST NO FILTERING

FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

ALMOST NO FILTERING

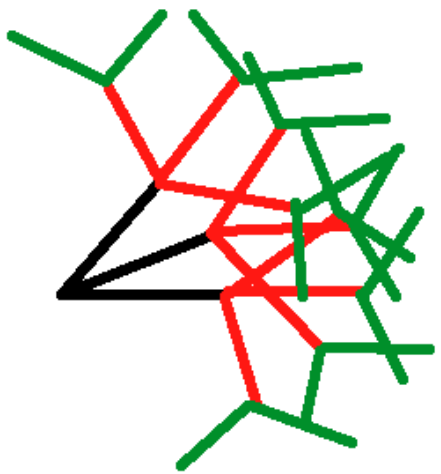


FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

ALMOST NO FILTERING

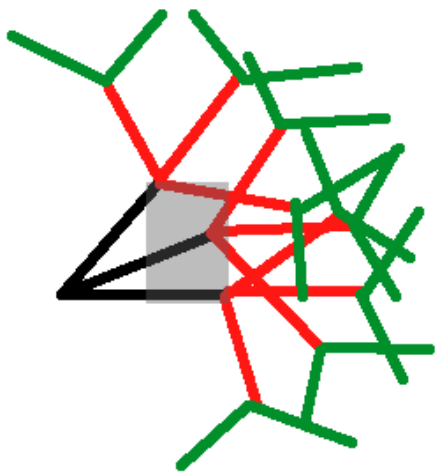


FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

ALMOST NO FILTERING

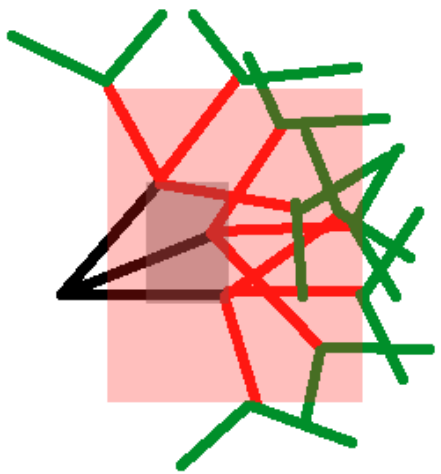


FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

ALMOST NO FILTERING

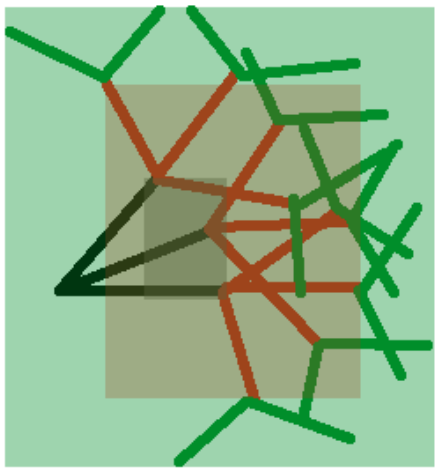


FILTERING ALGORITHM

BOUNDS CONSISTENCY?

- Compute spatial bounding box for end points of assembled bodies
- Simple bounds consistency degenerates into a large box
- The bounds degenerate immediately

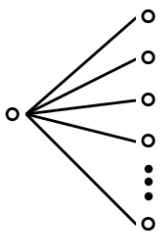
ALMOST NO FILTERING



FILTERING ALGORITHM

CLUSTERING: MORE ACCURATE DESCRIPTION

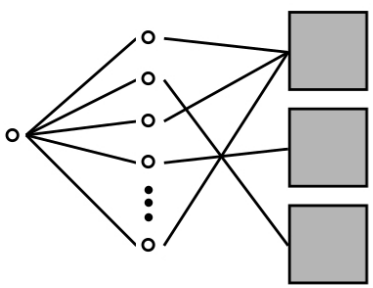
- Clustering both in space and angle
- Each fragment in a body is assembled and clustered
- Number of clusters and errors are controlled
- Balance between simple bounds consistency and complete explosion
- Clusters are intersected with JM spatial domains



FILTERING ALGORITHM

CLUSTERING: MORE ACCURATE DESCRIPTION

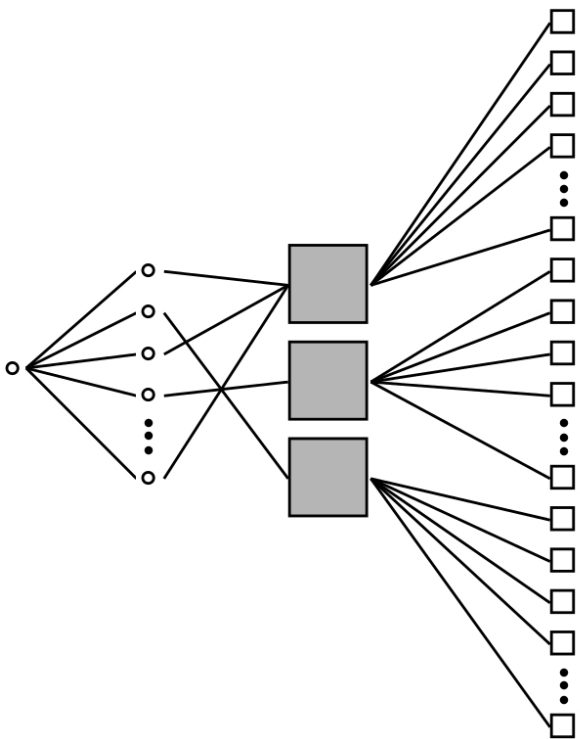
- Clustering both in space and angle
- Each fragment in a body is assembled and clustered
- Number of clusters and errors are controlled
- Balance between simple bounds consistency and complete explosion
- Clusters are intersected with JM spatial domains



FILTERING ALGORITHM

CLUSTERING: MORE ACCURATE DESCRIPTION

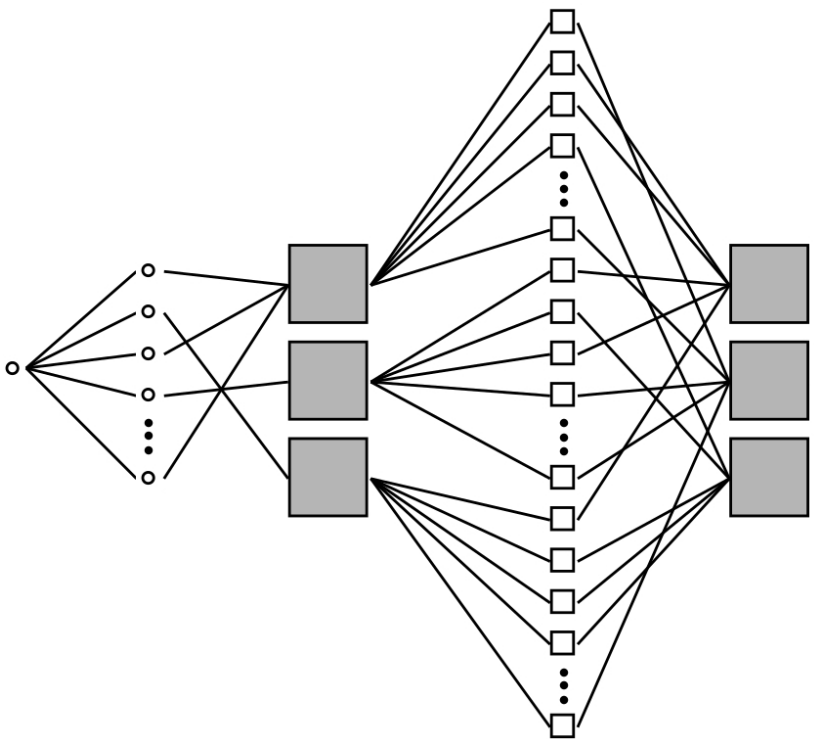
- Clustering both in space and angle
- Each fragment in a body is assembled and clustered
- Number of clusters and errors are controlled
- Balance between simple bounds consistency and complete explosion
- Clusters are intersected with JM spatial domains



FILTERING ALGORITHM

CLUSTERING: MORE ACCURATE DESCRIPTION

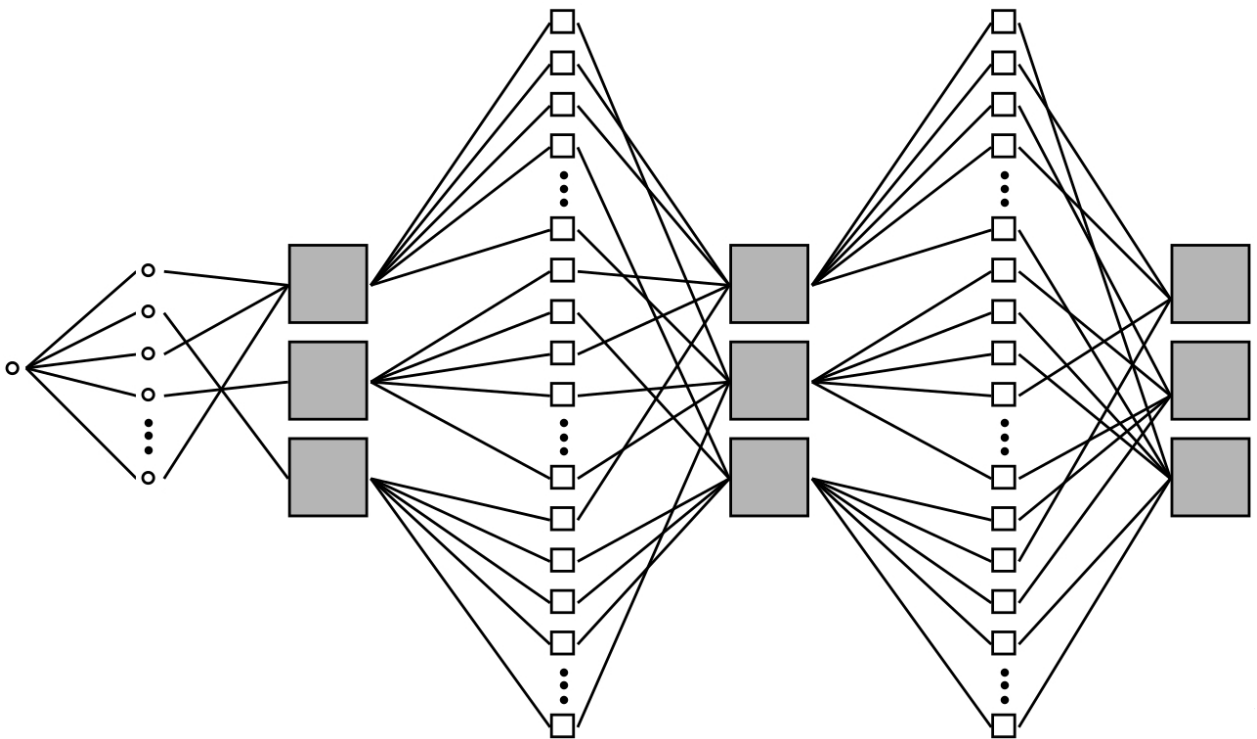
- Clustering both in space and angle
- Each fragment in a body is assembled and clustered
- Number of clusters and errors are controlled
- Balance between simple bounds consistency and complete explosion
- Clusters are intersected with JM spatial domains



FILTERING ALGORITHM

CLUSTERING: MORE ACCURATE DESCRIPTION

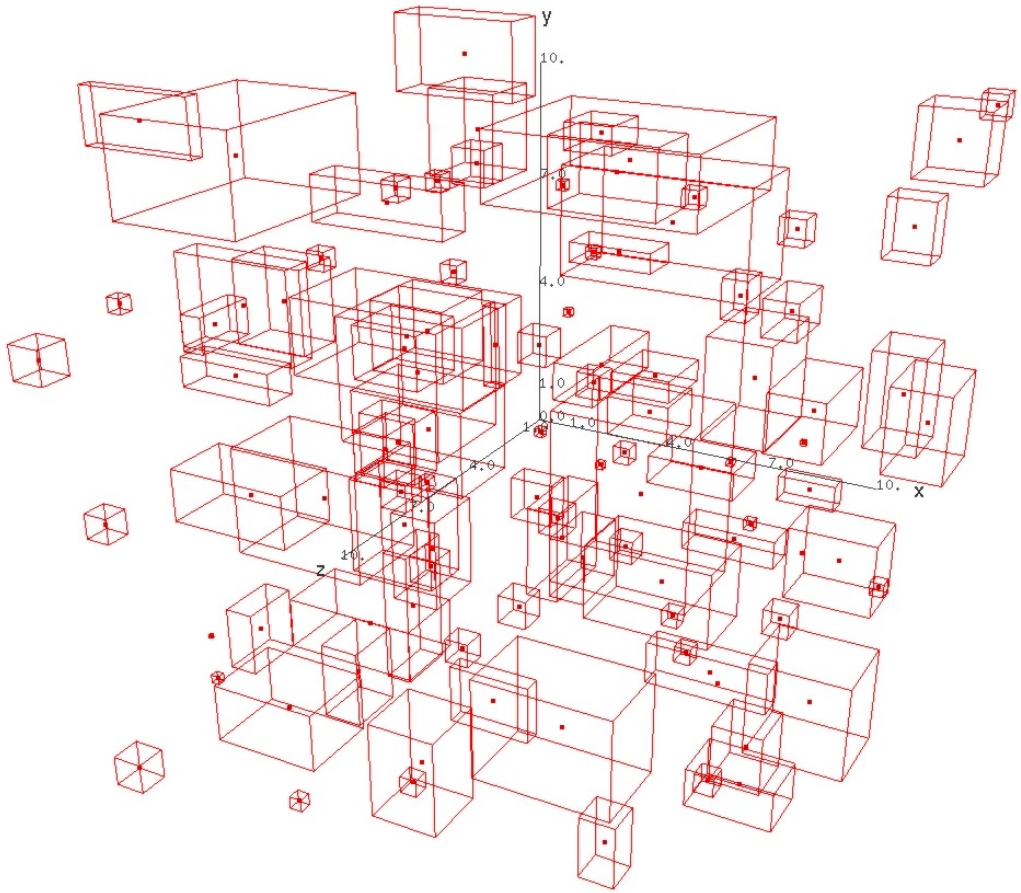
- Clustering both in space and angle
- Each fragment in a body is assembled and clustered
- Number of clusters and errors are controlled
- Balance between simple bounds consistency and complete explosion
- Clusters are intersected with JM spatial domains



FILTERING ALGORITHM

APPROXIMATED POLYNOMIAL TIME ALGORITHM

- Example of 3D clusters for a JM with a few bodies



FILTERING ALGORITHM

APPROXIMATED POLYNOMIAL TIME ALGORITHM

- Each cluster contains similar fragments
- Select a representative for each cluster
- Filter out non representatives (controlled error)
- Keep significantly different arrangements
- It can be seen as approximated direct kinematics

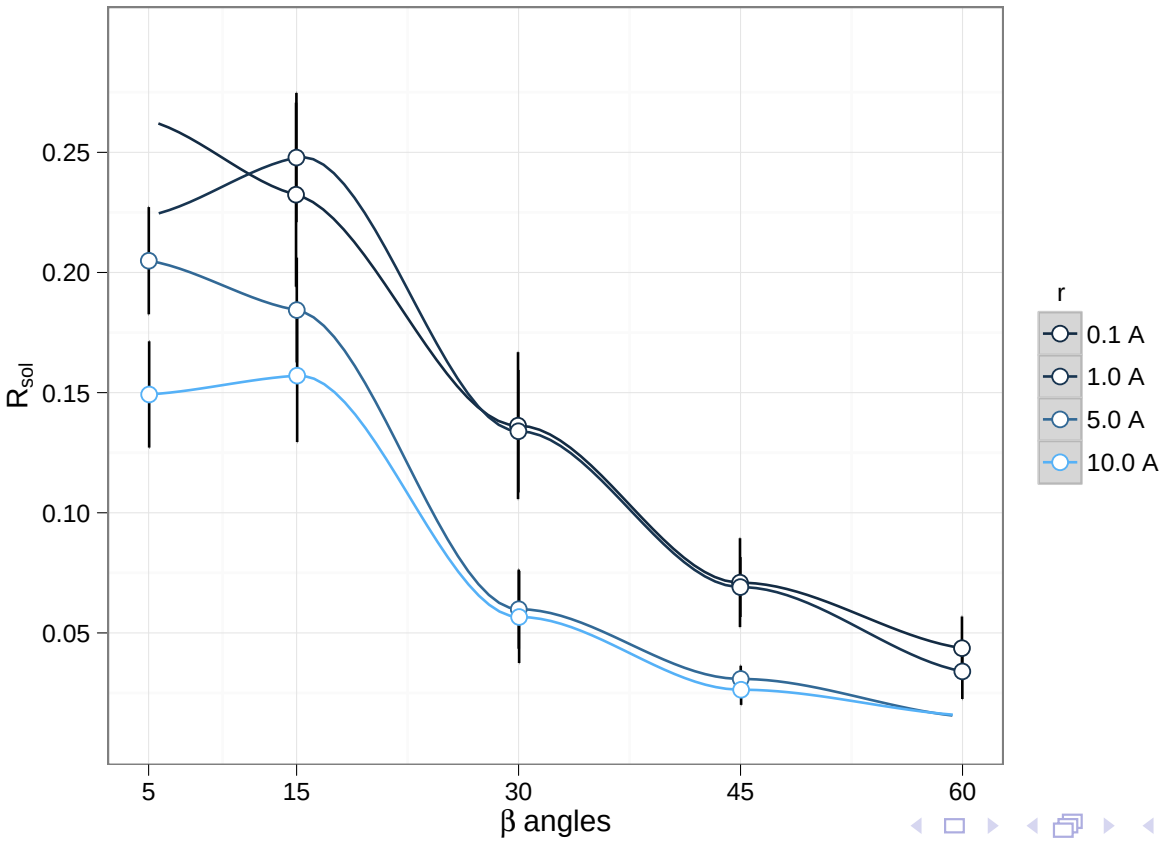
EXPERIMENTS

- We implemented the JM propagator in FIASCO (Fragment-based Interactive Assembly for protein Structure prediction with COnstraints)
- C++ constraint solver
- Host: Linux Intel Core i7 860, 2.5 GHz, 8 GB

EXPERIMENTS

SEARCH SPACE EXPLORED

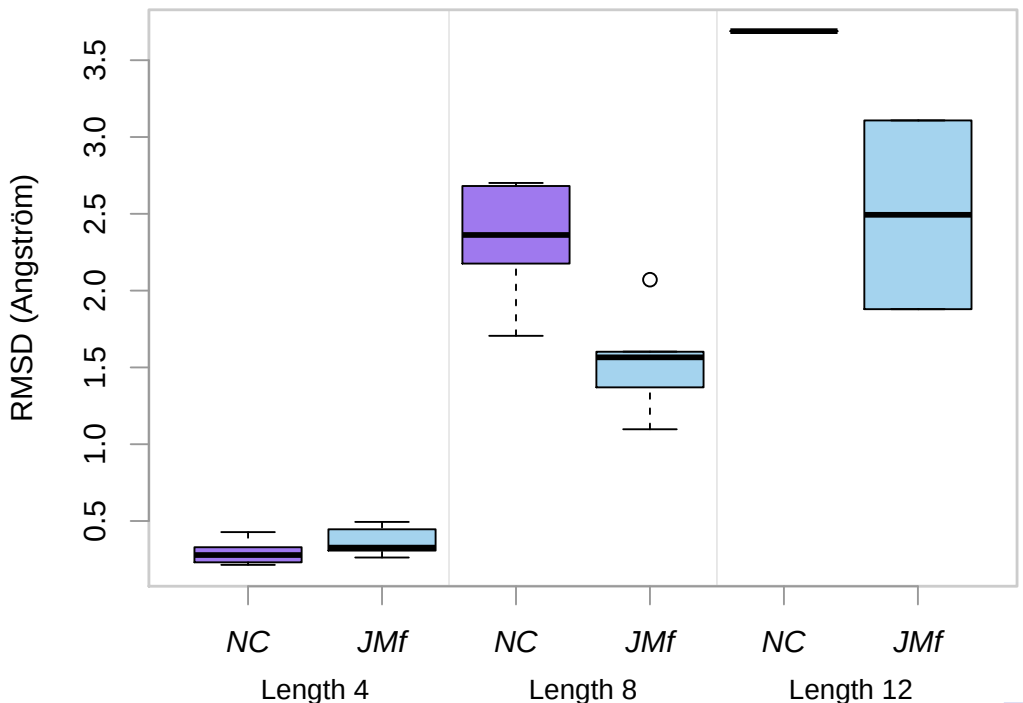
- Loop length 4, 60 fragments per amino acid, at most 100 clusters per anchor
- JM filtered solutions vs unconstrained solutions
- Increasing space and angle clusters size, $\Rightarrow$ less solutions



EXPERIMENTS

QUALITY OF FILTERED SOLUTIONS

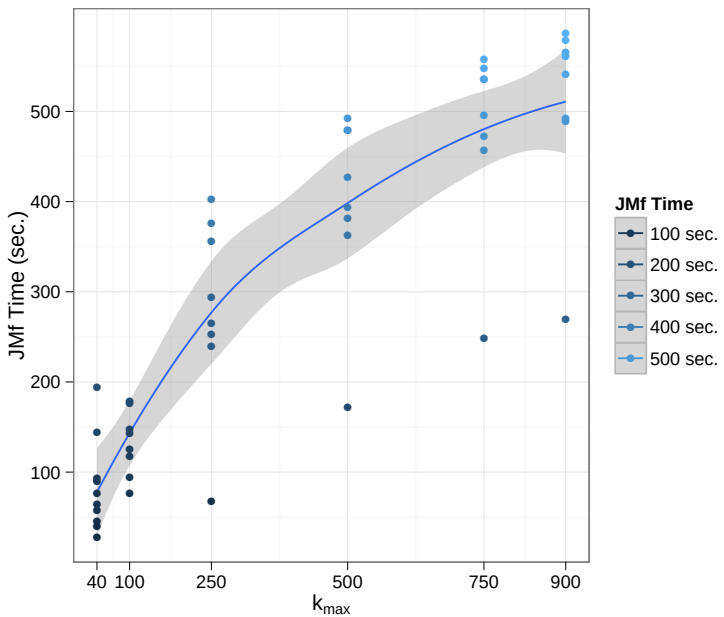
- Can you guarantee that filtered solutions do not deviate from optimal?
- Length 4: 1.0 Å and 15°; Length 8,12: 2.5 Å and 60°
- Comparing best RMSD from original found

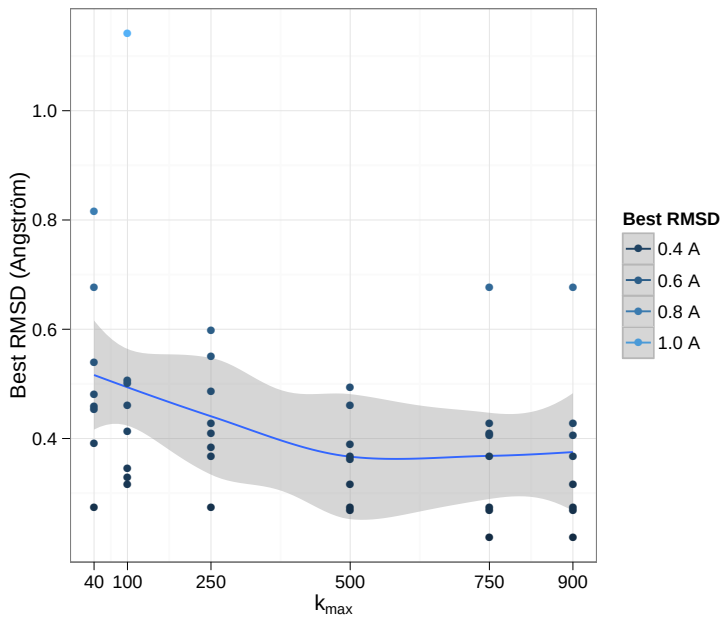


EXPERIMENTS

QUALITY VS EFFICIENCY

- Number of clusters affects quality and computational times
- Clusters features: 1.0 Å and 30°
- After 500 clusters no relevant benefits





EXPERIMENTS

LOOP PREDICTOR COMPARISON

- We compared our tool against other 3 popular predictors
- We used a general fragment database
- Our results are on the same order of magnitude

Loop Length	**Average RMSD**			
	CCD	SOS	FALCm	JMf
4	0.56	0.20	0.22	0.30
8	1.59	1.19	0.72	1.31
12	3.05	2.25	1.81	1.97

CONCLUSIONS

The paper contains:

- Novel constraint to model rigid bodies with degrees of freedom
- NP-completeness proof
- Approximated propagator (directional)
- Experimental validation

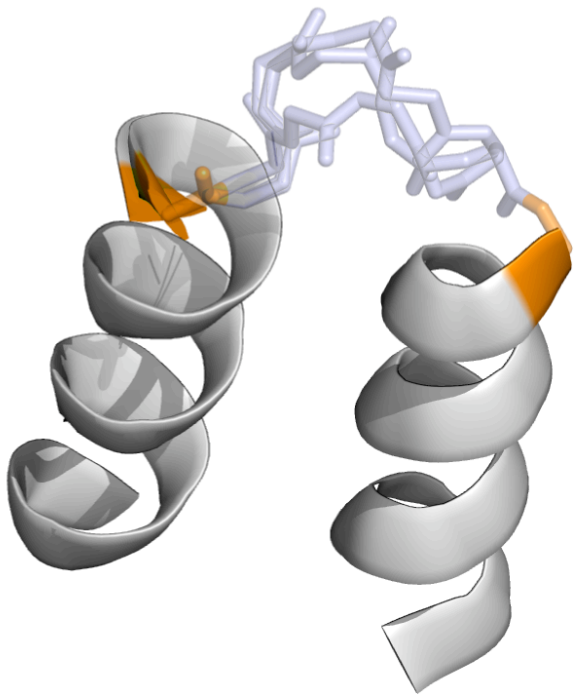
Future work:

- Bidirectional propagator
- Loop fragment databases
- Uniform spatial conformation sampling based on JM
- Interactive feedback (FIASCO GUI)

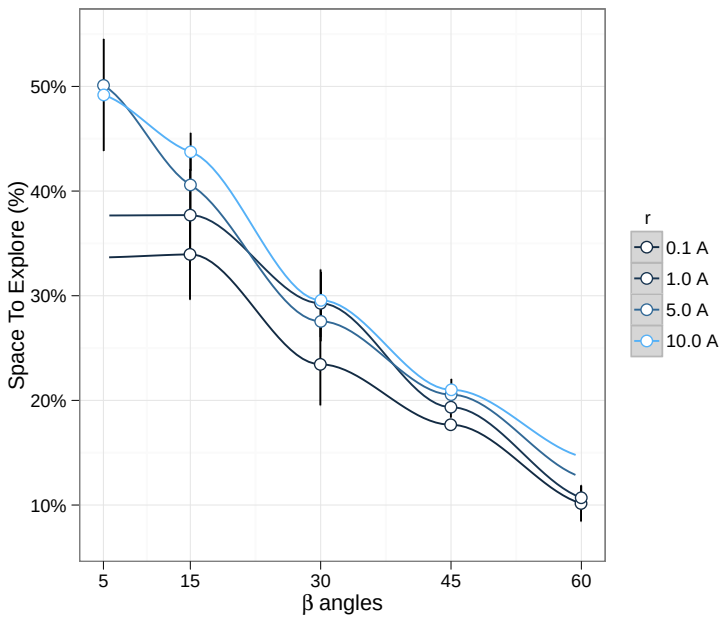
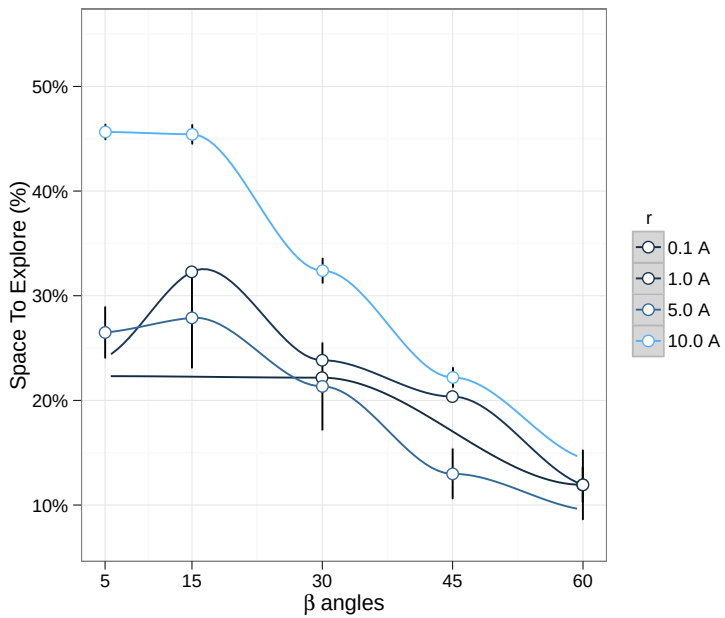
CONCLUSIONS

Thank you!

Please visit `www.cs.nmsu.edu/fiasco/`



EXPERIMENTS

SEARCH SPACE EXPLORED

- Larger loops length (8 and 12), can't do exhaustive search
- Lower bound to the filtering ratio