# Constraint Programming and Biology: Fragment Assemby with CP

Agostino Dovier

Dept. Math and Computer Science, Univ. of Udine, Italy

ACP Summer School in Constraint Programming
Wrocław, September 2012

# Fragment assembly

- We would like to model the PSP off-lattice, but using finite domain variables.

# Fragment assembly

- We would like to model the PSP off-lattice, but using finite domain variables.
- The main idea is to analyze the known proteins and find some statistics between the angles formed by fragments of 4 (or more) amino acids.

# Fragment assembly

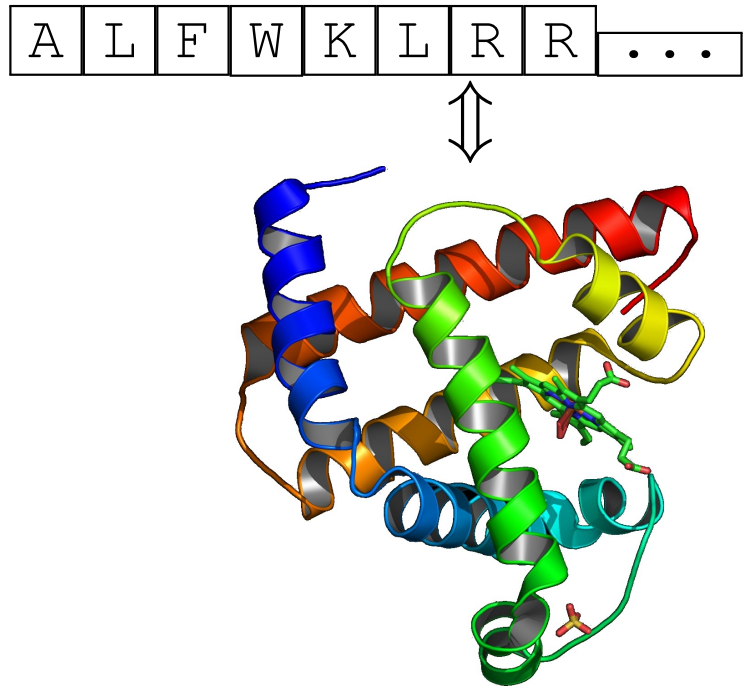
- We would like to model the PSP off-lattice, but using finite domain variables.
- The main idea is to analyze the known proteins and find some statistics between the angles formed by fragments of 4 (or more) amino acids.
- Then, using some clustering (in $\mathbb{R}^3$), assigning a set of available fragments (indexed by an integer) to subsequences of the known protein.

# Fragment assembly

- We would like to model the PSP off-lattice, but using finite domain variables.
- The main idea is to analyze the known proteins and find some statistics between the angles formed by fragments of 4 (or more) amino acids.
- Then, using some clustering (in $\mathbb{R}^3$), assigning a set of available fragments (indexed by an integer) to subsequences of the known protein.
- The approach might be incomplete, however, we (and others) assume that if nature prefers some local shapes $\Longrightarrow$ we should do it as well

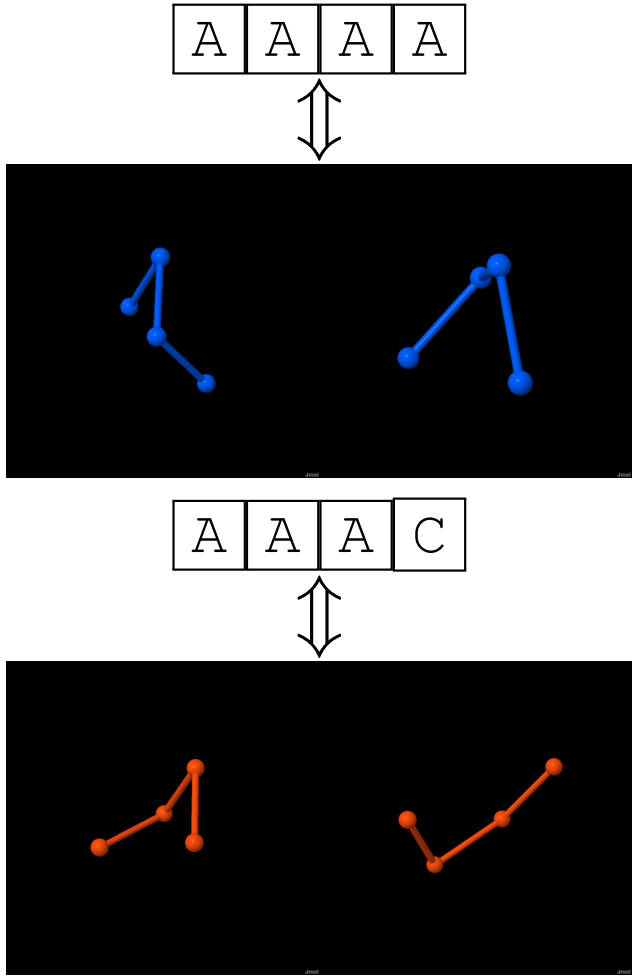
# Preprocessing

The Protein Data Bank contains $\geq 60K$ protein sequences with their observed 3D structures (X-ray/NMR)

# PDB: extract information

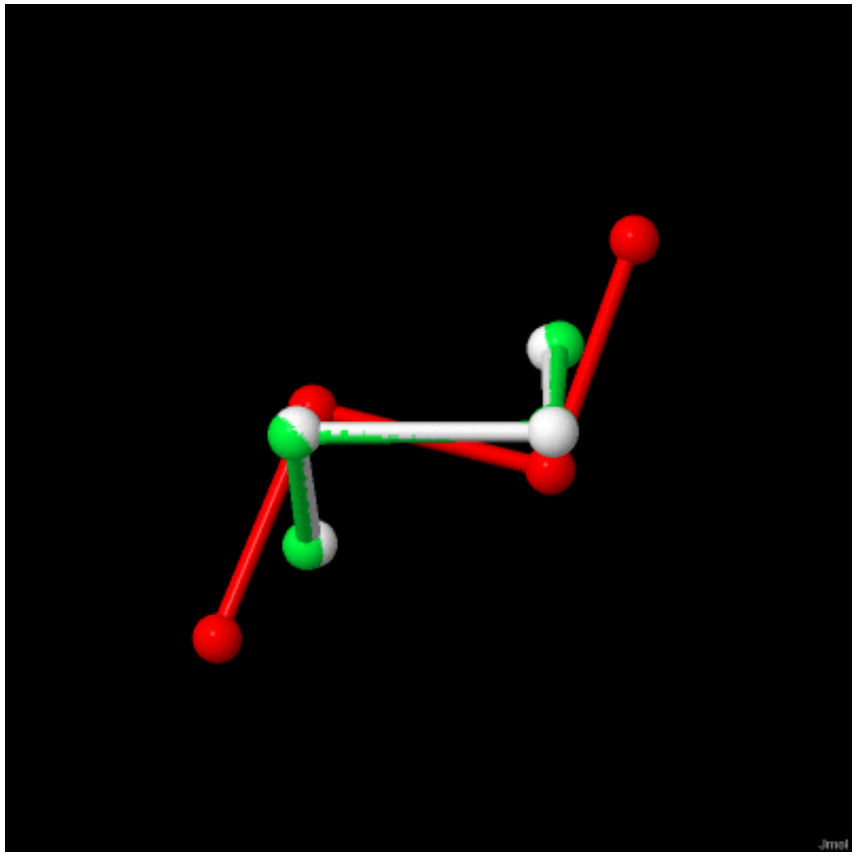
We get fragments composed of 4 consecutive amino acids and collect the corresponding shapes (indexed by sequence)

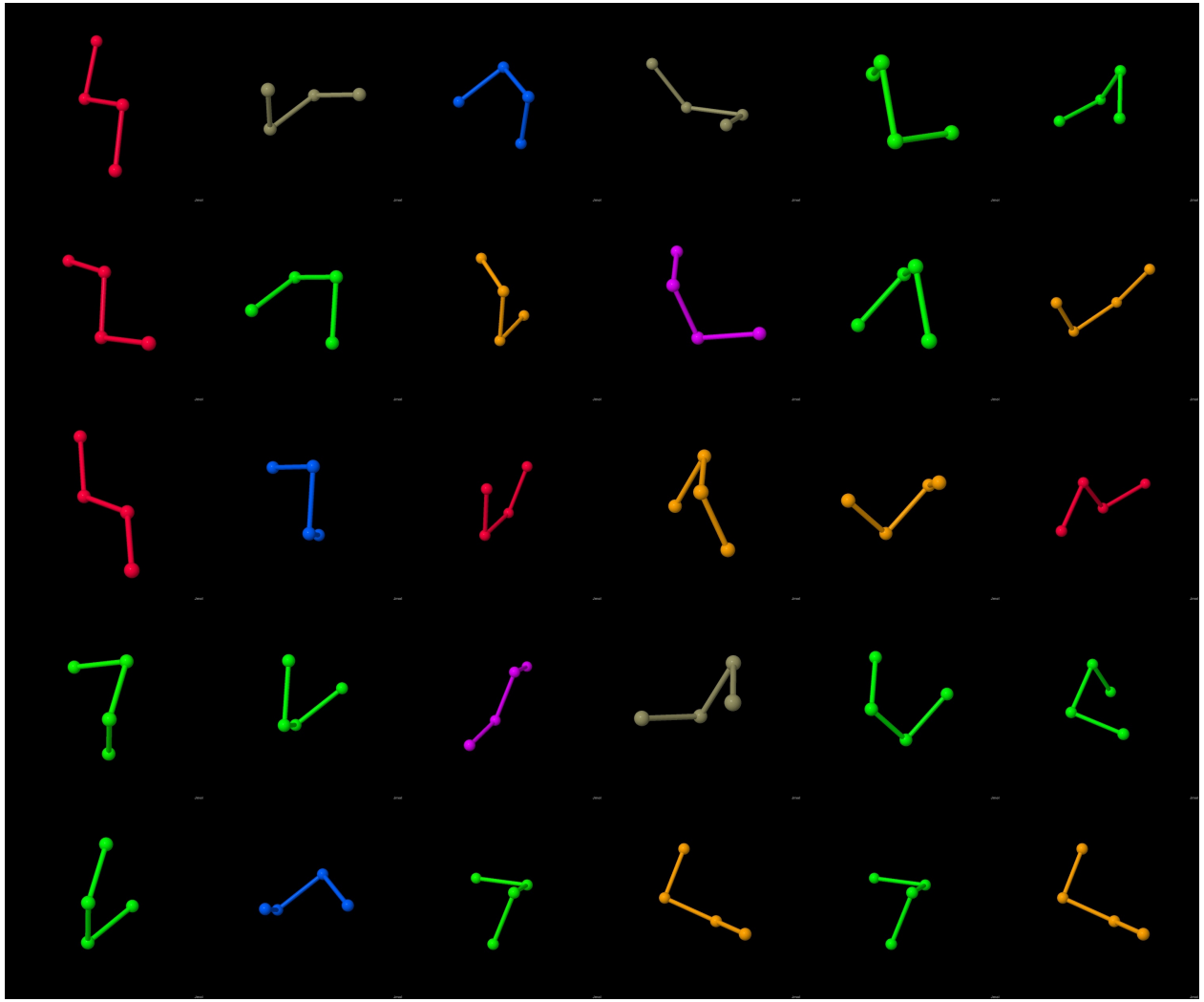

...

# Clustering (same 4-ple, different shapes)

Clustering according to their similarity (RMSD $\leq$ threshold)
White and green form a single cluster



# Clustered conformations for AAAA

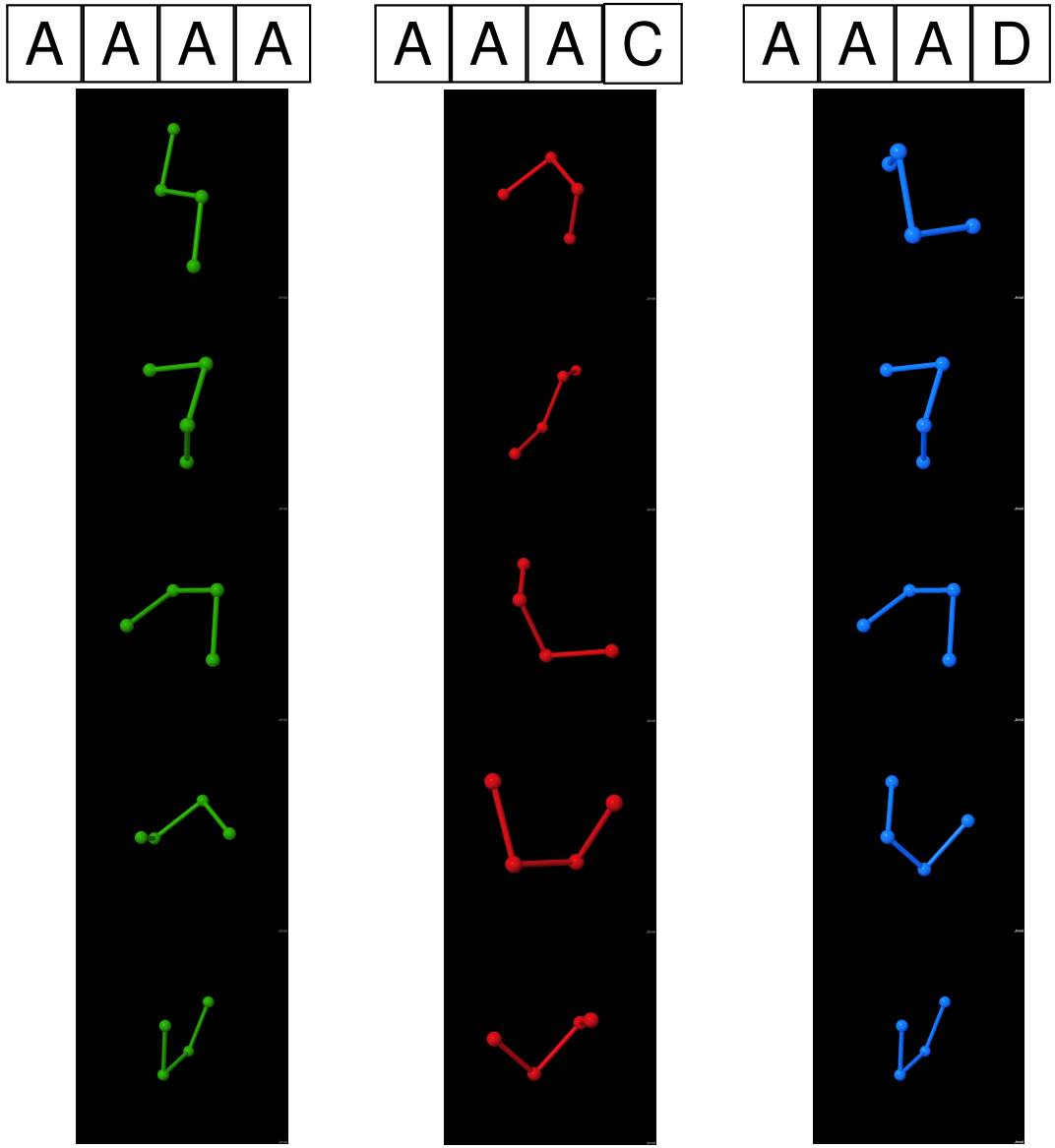


Each color has a representative and frequency count

# Library of fragments

For each 4 aa sequence, store the clustered representatives (RMSD $\leq$ .5Å)

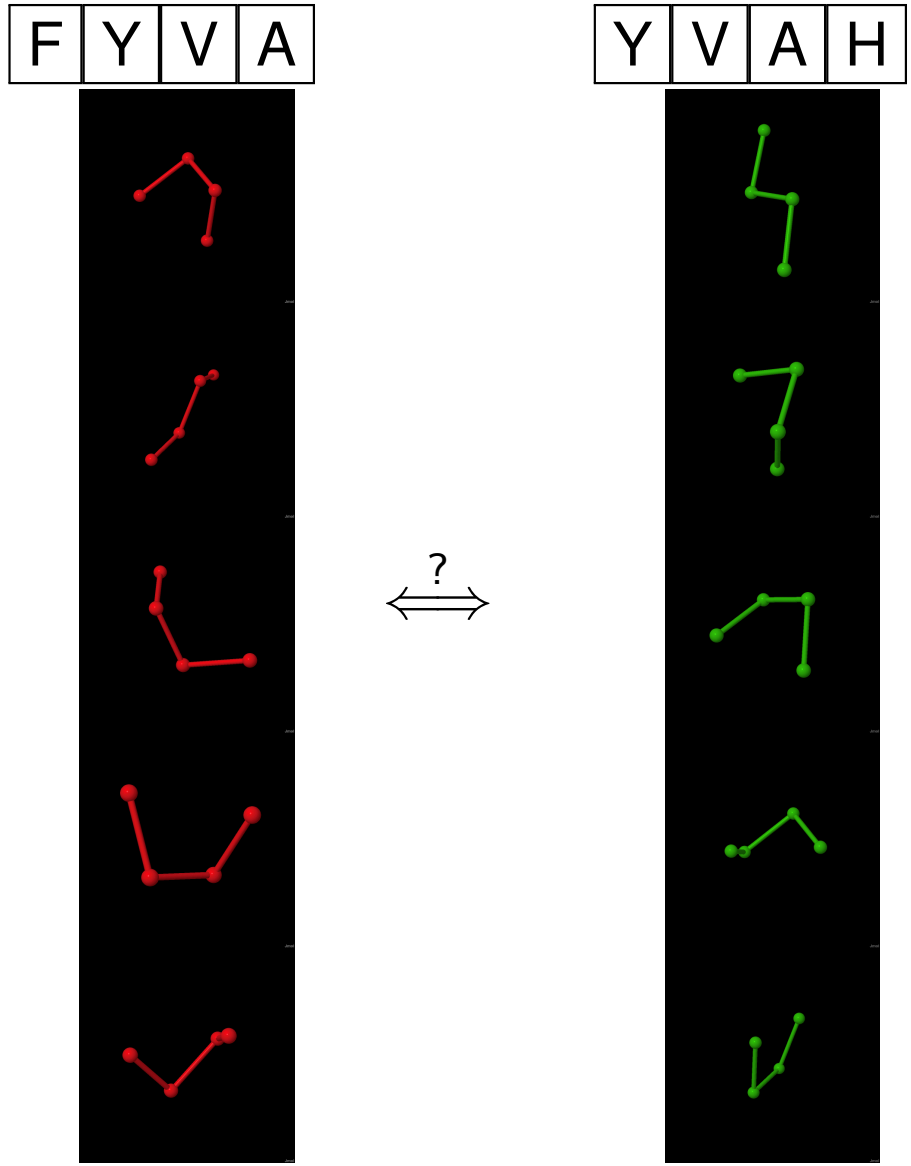
```
tupla([A,A,A,A],
[0.0,0.0,0.0,
 2.5,-2.8,0.3,
 1.9,-3.1,4.0,
 -1.9,-3.4,3.6],
 Freq, ID).
```



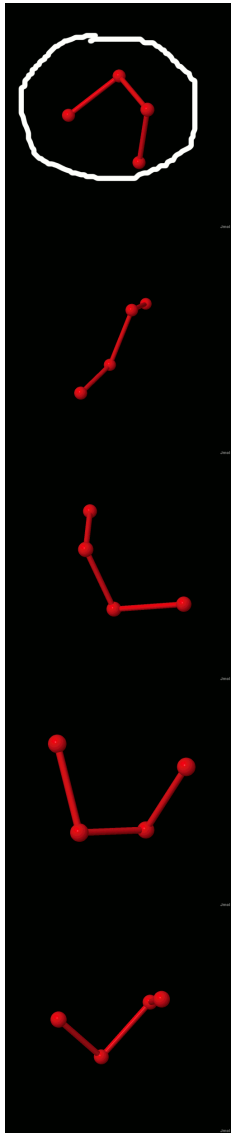
# Combiningthe blocks

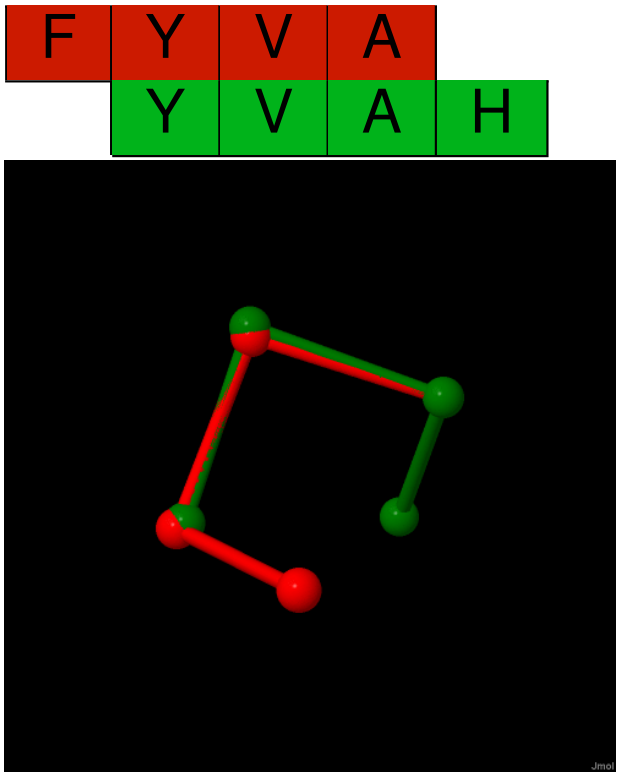
| F | Y | V | A | H | ... |
|---|---|---|---|---|---|
| F | Y | V | A | | |
| | Y | V | A | H | |
| | | V | A | H | ... |

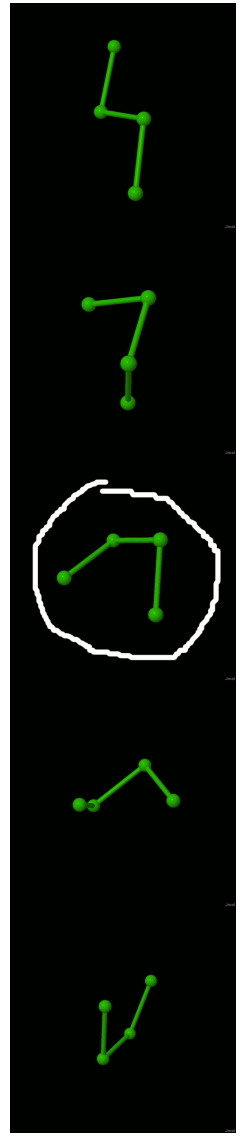
How to assemble fragments?



# Inductive step: combine the blocks

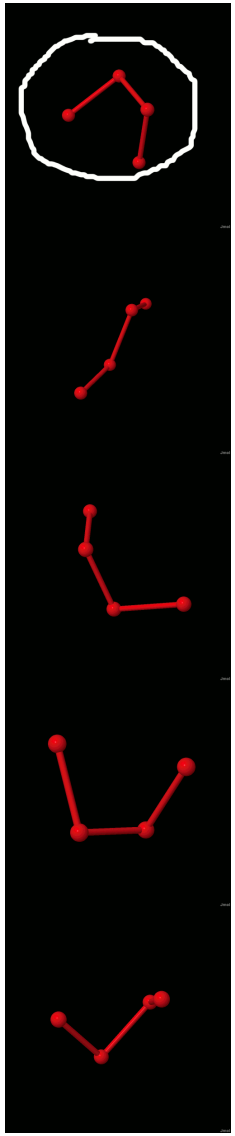


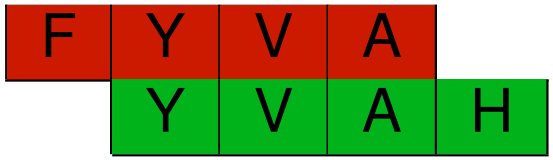


Two fragments are *compatible* only if the 3 common amino acids have a low RMSD (similar bend angle)

# Inductive step: combine the blocks

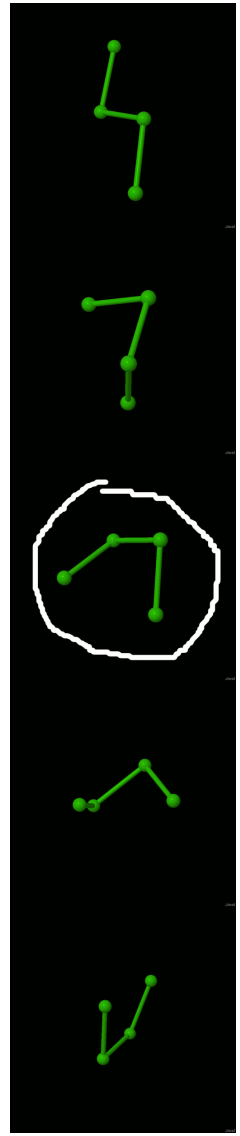





Each compatible pair of fragments is stored as

$$next(F_i, F_j, M)$$

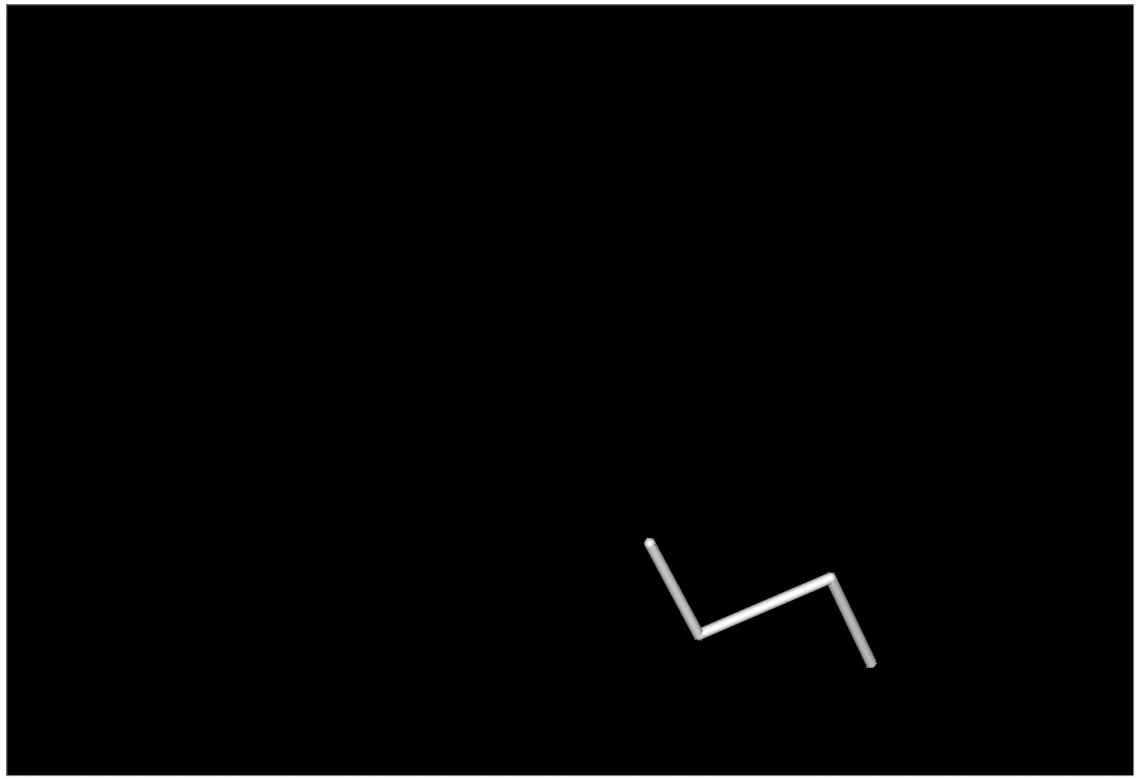
with optimal rotation matrix M (that rotates $F_j$ in the reference of $F_i$)



# Inductive step: combine the blocks

## The assembly

Given a target sequence, pick the first 4-aa fragment.
The protein is grown by attaching compatible fragments (*next*).

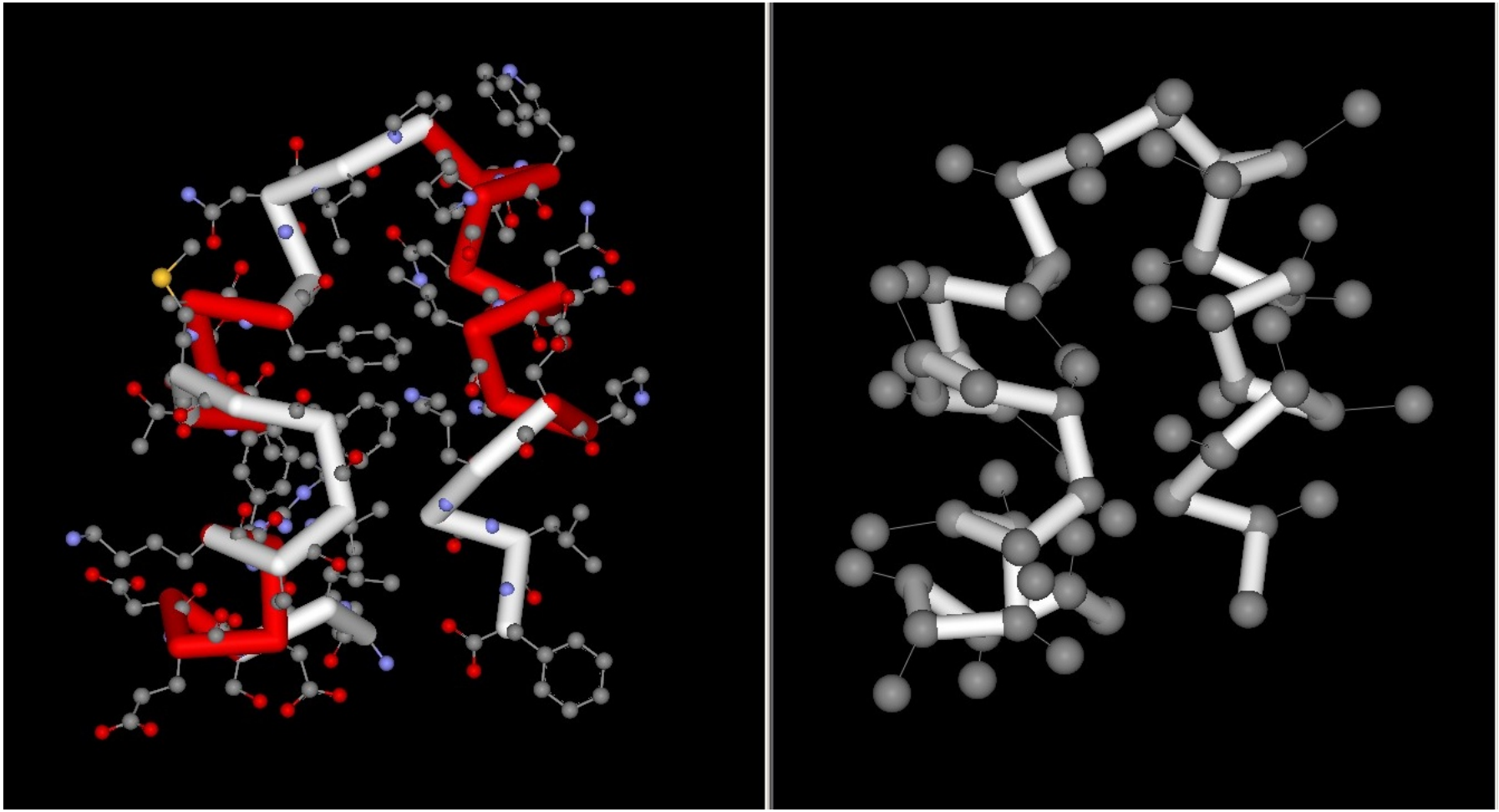


# Enriching the model

- Given a $C\alpha$ 4-tuple in 3D, a small degree of freedom for the position of the side chain is allowed
- Different amino acids have different occupation
- A pure $C\alpha$-$C\alpha$ model does not keep into account these differencies
- We consider the positions of the centroids of the side chains.
- Roughly, a centroid is the expected center of mass of the side chain

# Enriching the model

## The centroids



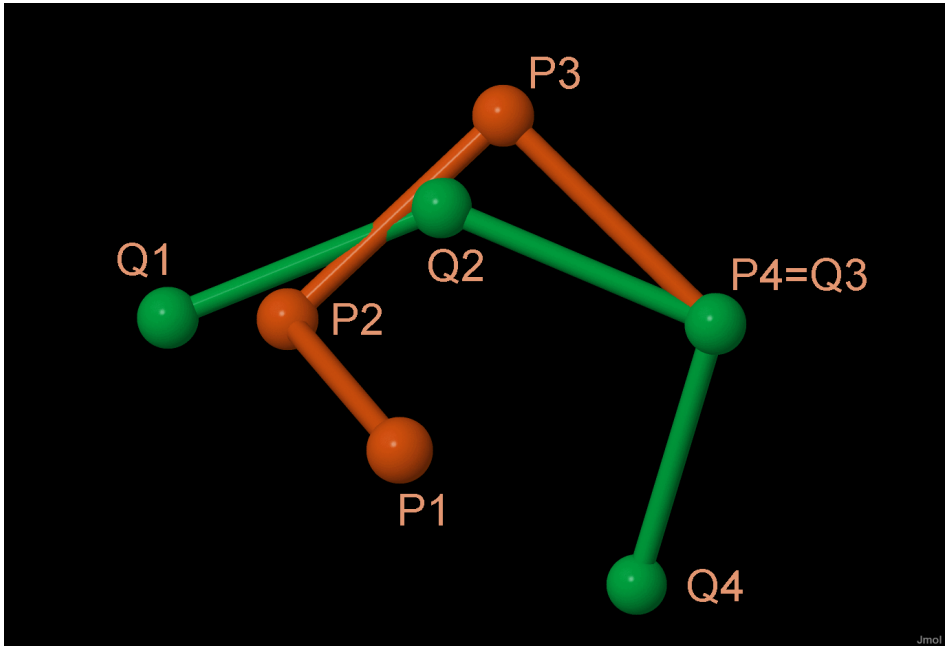
# Variables and domains

- Given amino acid sequence: $[a_1, a_2, \ldots, a_n]$,
- Amino acid positions: $X_1^\alpha, Y_1^\alpha, Z_1^\alpha, \ldots, X_n^\alpha, Y_n^\alpha, Z_n^\alpha$, $X_2^c, Y_2^c, Z_2^c, \ldots, X_{n-1}^c, Y_{n-1}^c, Z_{n-1}^c$
- Domains: discretized (0.01Å) 3D positions.
- Fragments: $F_1, F_2, \ldots, F_{n-3}$.
- Domains: fragments ID from the library for $[a_i, a_{i+1}, a_{i+2}, a_{i+3}]$.
- Variables for rotation matrices: $R_1, R_2, \ldots, R_{n-3}$.

# Main constraints

- $F_i$ and $F_{i+1}$ can take only compatible assignments (described by $next(F_i, F_{i+1}, M_{i,(i+1)})$)
- the constraint is posted using `table` builtin
- All fragments are stored in the same reference, thus need to rotate reference, while building the fragments:
- $R_{i+1} = M_{i,(i+1)} \cdot R_i$ implemented as a constraint

# Link positions to fragments




- $Q_4 = R_{i+1} \cdot F_{i+1} + \mathit{shift}$ ($P$s already placed $= F_i$)
- correct shift overlaps $P_4$ and $Q_3$, so that distance between $(X^{\alpha}_{i+3}, Y^{\alpha}_{i+3}, Z^{\alpha}_{i+3})$ and $(X^{\alpha}_{i+4}, Y^{\alpha}_{i+4}, Z^{\alpha})$ is $\sim$3.81Å.

# Distance constraints

- For each $i$, $j$, introduce a distance var $D_{i,j}$
- $D_{i,j} = (X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2$
- $D_{i,j} \geq$ *min_dist* (all_distant) — (*)
- $D_{i,j} \leq \texttt{diameter}^2$ (compact_factor)
- If ssbond(i,j), $D_{i,j} \leq 6$Å.

# Distance constraints

- For each $i$, $j$, introduce a distance var $D_{i,j}$
- $D_{i,j} = (X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2$
- $D_{i,j} \geq$ *min_dist* (all_distant) — (*)
- $D_{i,j} \leq \texttt{diameter}^2$ (compact_factor)
- If ssbond(i,j), $D_{i,j} \leq 6$Å.
- $D_{i,j} \leq 3.81 * |i - j|$ (max stretch)
- Triangular inequality (over $D_{i,j}, D_{i,k}, D_{j,k}$)

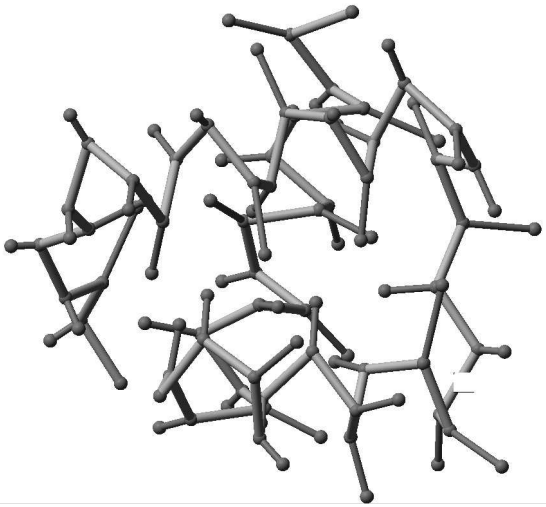
# The Energy Function

- Each conformation has three energy contributions:
- *Contact*: each pair of centroids gives a contribution depending on distance and type.
- *Torsional*: each torsion is scored according to a statistical profile for the local sequence (and a *Torsional correlation* for each pair of consecutive torsion angles is also considered).
- *Parallel*: A component that considers the relative orientation of spatially close triplets (parallel structures are favoured)

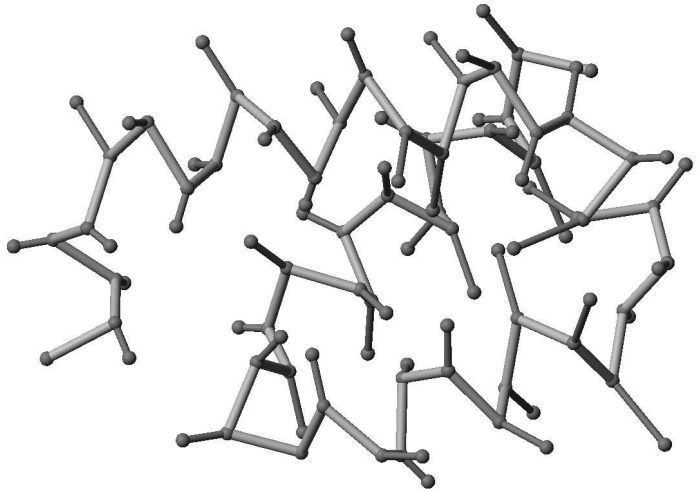
# The Energy Function

- Each conformation has three energy contributions:
- *Contact*: each pair of centroids gives a contribution depending on distance and type.
- *Torsional*: each torsion is scored according to a statistical profile for the local sequence (and a *Torsional correlation* for each pair of consecutive torsion angles is also considered).
- *Parallel*: A component that considers the relative orientation of spatially close triplets (parallel structures are favoured)
- The relative weights of these contributions have been experimentally determined trying to correlate energy and RMSD

# Results: 1ENH

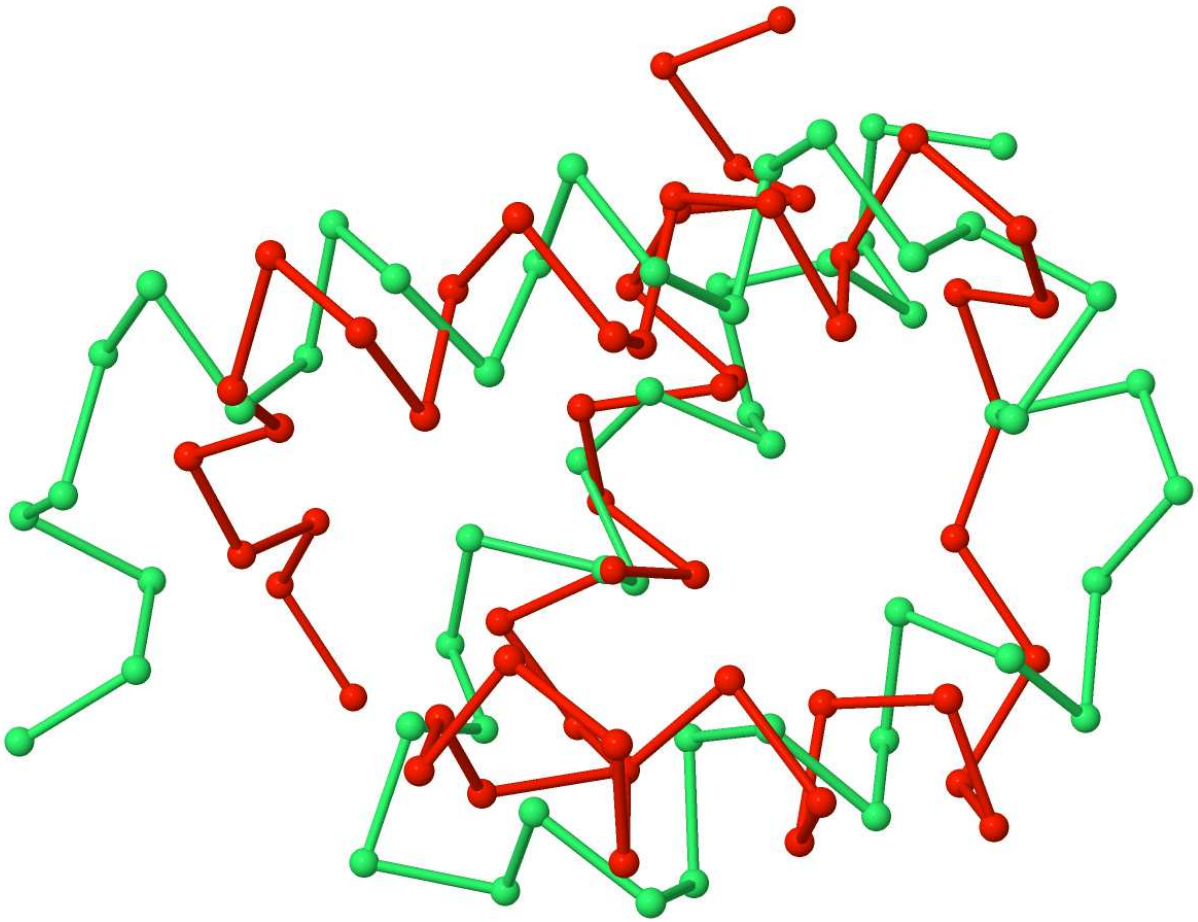


Original



Computed

# Results: 1ENH



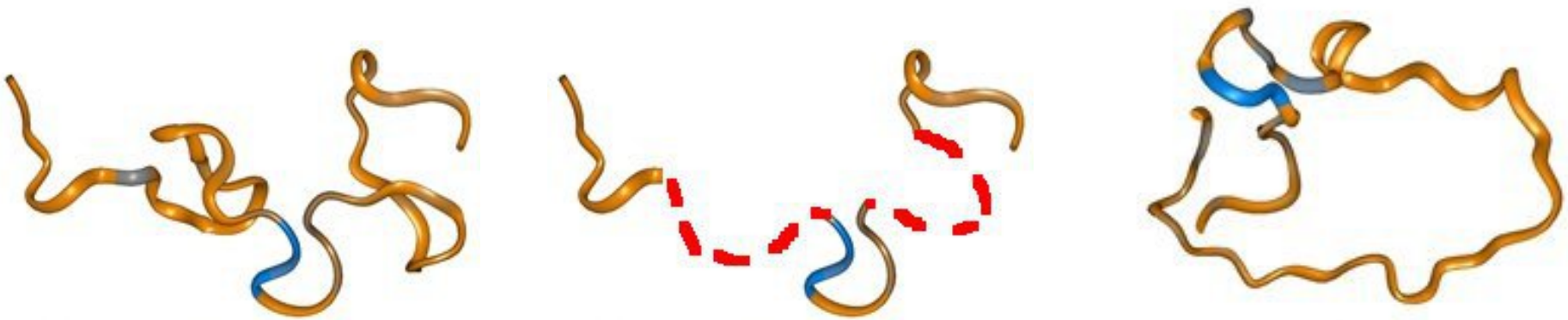
# Large Neighboring Search

- This approach is inherently approximated.
- This justifies using approximated search methods.
- We employed Large Neighboring Search
- The search for next solutions is performed by exploring a *large* neighborhood

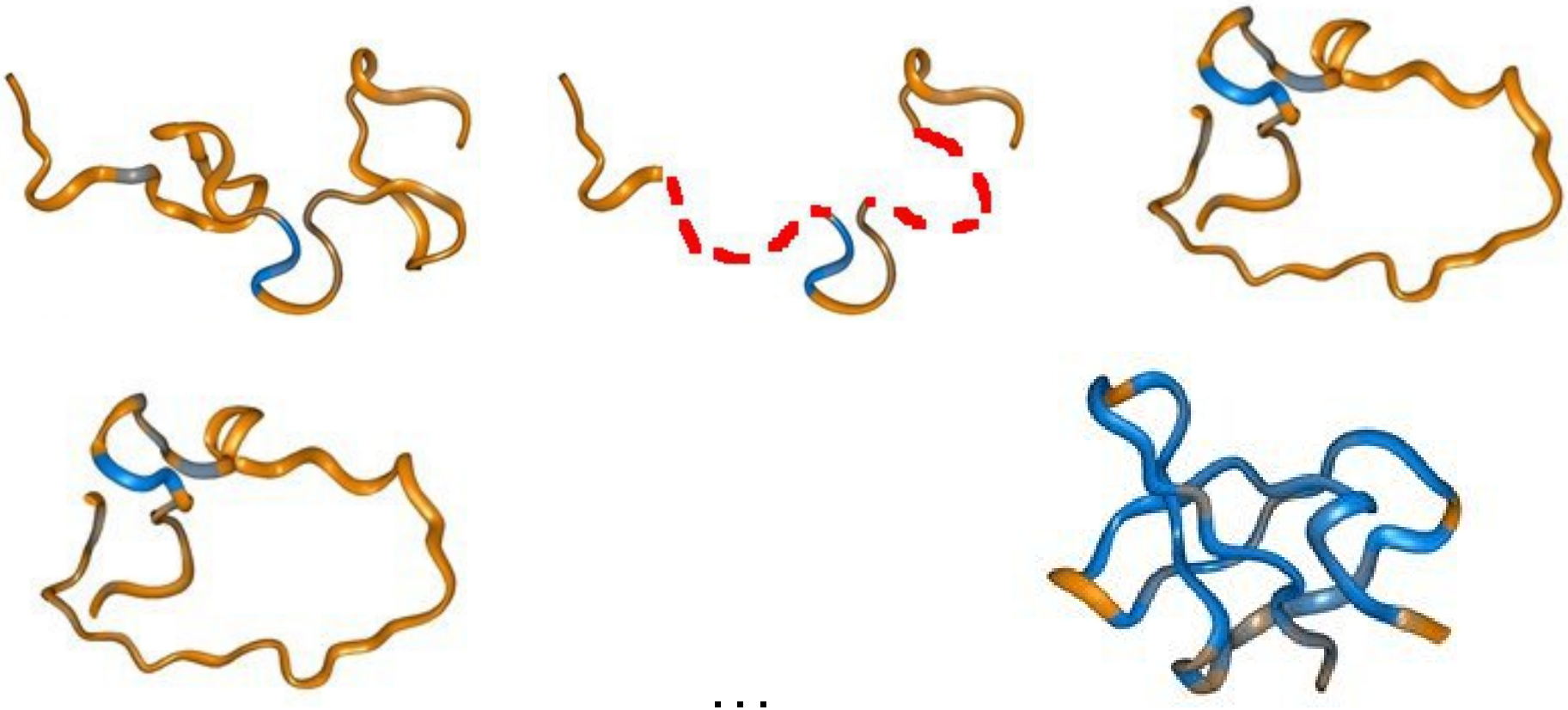
# Large Neighboring Search

- This approach is inherently approximated.
- This justifies using approximated search methods.
- We employed Large Neighboring Search
- The search for next solutions is performed by exploring a *large* neighborhood
- Find a feasible solution.
- Repeat until timeout:
  - ✓ A (almost random) large number of variables (subject to constraints) is allowed to change.
  - ✓ A new (best or equal) solution is found (some variants here)

# Large Neighboring Search



# Large Neighboring Search



# The Joined-Multibody Constraint

- We can generalize the approach. We could have some known parts (rigid blocks) and some unknowns parts (with a set of fragments).
- We might have some spatial constraints on various atoms/amino acids (and the usual non-overlapping constraints)
- We have put all this into a global constraint, proved that it is untractable and developed approximated approximation algorithms.
- This is the joint-multibody constraint
- We have used this codes to predict loops namely the form of a subset of a protein connecting two knows proteins (this is a typical problem for biologists)

# The Joined-Multibody Constraint

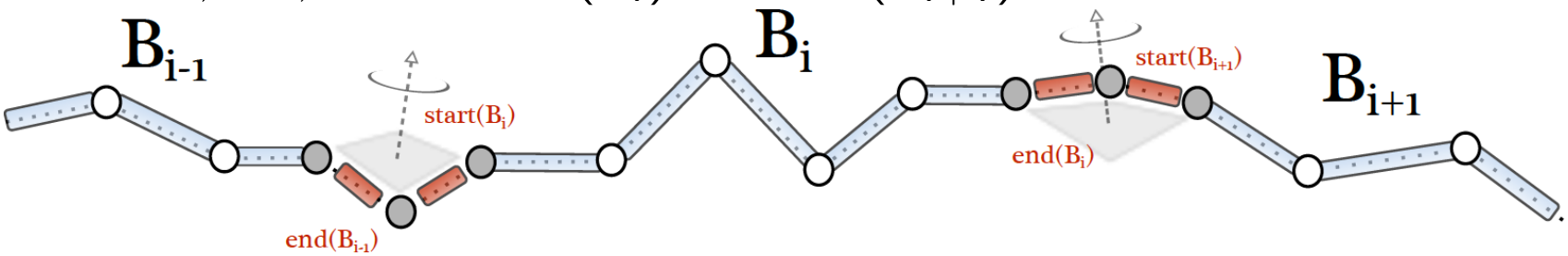
- A *rigid block* $B$ is an ordered list of at least three (distinct) 3D points, denoted by points($B$). start($B$) and end($B$) are the lists of the first three and the last three points of points($B$).
  For two lists of points $\vec{p}$ and $\vec{q}$, we write $\vec{p} \frown \vec{q}$ if they can be perfectly overlapped by a *roto-translation*.

# The Joined-Multibody Constraint

- A *rigid block* $B$ is an ordered list of at least three (distinct) 3D points, denoted by points($B$). start($B$) and end($B$) are the lists of the first three and the last three points of points($B$).
  For two lists of points $\vec{p}$ and $\vec{q}$, we write $\vec{p} \frown \vec{q}$ if they can be perfectly overlapped by a *roto-translation*.
- A *multi-body* is a sequence $S_1, \ldots, S_n$ of non-empty sets of rigid blocks.

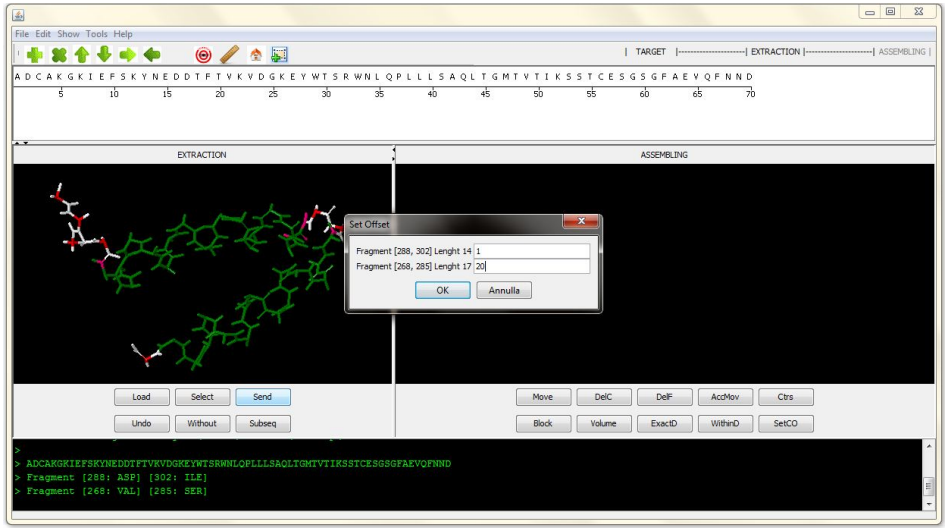
# The Joined-Multibody Constraint

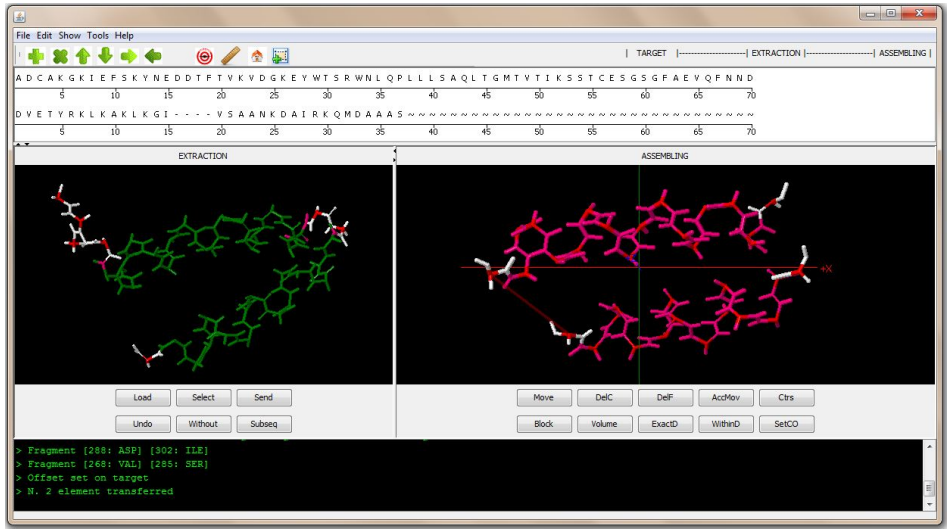
- A *rigid block* $B$ is an ordered list of at least three (distinct) 3D points, denoted by points($B$). start($B$) and end($B$) are the lists of the first three and the last three points of points($B$).
  For two lists of points $\vec{p}$ and $\vec{q}$, we write $\vec{p} \frown \vec{q}$ if they can be perfectly overlapped by a *roto-translation*.
- A *multi-body* is a sequence $S_1, \ldots, S_n$ of non-empty sets of rigid blocks.
- A sequence of rigid blocks $B_1, \ldots, B_n$, is called a *rigid body* if, for all $i = 1, \ldots, n-1$, end($B_i$) $\frown$ start($B_{i+1}$).




- Basically, the JM constraint is the formalization of the problem of finding a rigid body from a multi body that fulfills a set of spatial constraints.

# FIASCO: Fragment-based Interactive Assembly for protein Structure prediction with COnstraints



# References


- A. Dal Palù, A. Dovier, F. Fogolari, and E. Pontelli. CLP-based protein fragment assembly. TPLP 10(4–6):709–724, July 2010,
- A. Dal Palù, A. Dovier, F. Fogolari, and E. Pontelli. Exploring Protein Fragment Assembly Using CLP. In IJCAI 2011, pp. 2590-2595.
- F. Campeotto, A. Dal Palù, A. Dovier, F. Fioretto, and E. Pontelli. A Filtering Technique for Fragment Assembly-based Proteins Loop Modeling with Constraints. In M. Milano ed., Proc of 18th International Conference on Principles and Practice of Constraint Programming, Quebec City, Canada, 8-12 October 2012. LNCS. (also in WCB 12)
- F. Campeotto, A. Dal Palù, A. Dovier, F. Fioretto, F. Fogolari, E. Pontelli, et al. Introducing FIASCO: Fragment-based Interactive Assembly for protein Structure prediction with COnstraints. WCB 11

- To conclude, I suggest to: Play with Foldit `http://fold.it/portal/` It is funny and (it can be) useful.