



# New clustering method to infer prototypes covering the 3D structures of nucleic acid fragments

Antoine Moniot, Isaure Chauvot de Beauchêne, Yann Guermeur

## ► To cite this version:

Antoine Moniot, Isaure Chauvot de Beauchêne, Yann Guermeur. New clustering method to infer prototypes covering the 3D structures of nucleic acid fragments. RECOMB 2021 - 25th International conference on research in computational molecular biology, Aug 2021, Padova/Virtual, Italy. hal-03432671

HAL Id: hal-03432671

<https://hal.science/hal-03432671v1>

Submitted on 17 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# New clustering method to infer prototypes covering the 3D structures of nucleic acid fragments

Antoine Moniot, Isaure Chauvet de Beauchene and Yann Guermeur

University of Lorraine, CNRS, INRIA, LORIA, Nancy, France  
antoine.moniot@loria.fr

## Introduction:

In structural biology, fragment-based 3D modeling methods make use of fragment libraries. The library associated with one fragment represents the whole set of possible 3D structures (conformations) that it can adopt (with a chosen precision). Given the computational constraints, deriving libraries of minimal cardinality appears as a strong requirement. This amounts to deriving  $\varepsilon$ -nets whose cardinalities are as close as possible to the corresponding *covering numbers*. A heuristic to derive such  $\varepsilon$ -nets is to cluster the observed conformations under appropriate constraints, and keep only the *representatives/prototypes*. In the framework of interest, the main difficulty encountered is to implement the clustering with the RMSD as dissimilarity measure. Indeed, the computation of this measure follows a superimposition, the nature of which has both biological and mathematical consequences.

We introduce such a method as a variant of the Hierarchical Agglomerative Clustering (HAC) algorithm. Compared to HAC, it makes it possible to reduce the number of prototypes, while maintaining an acceptable computation time.

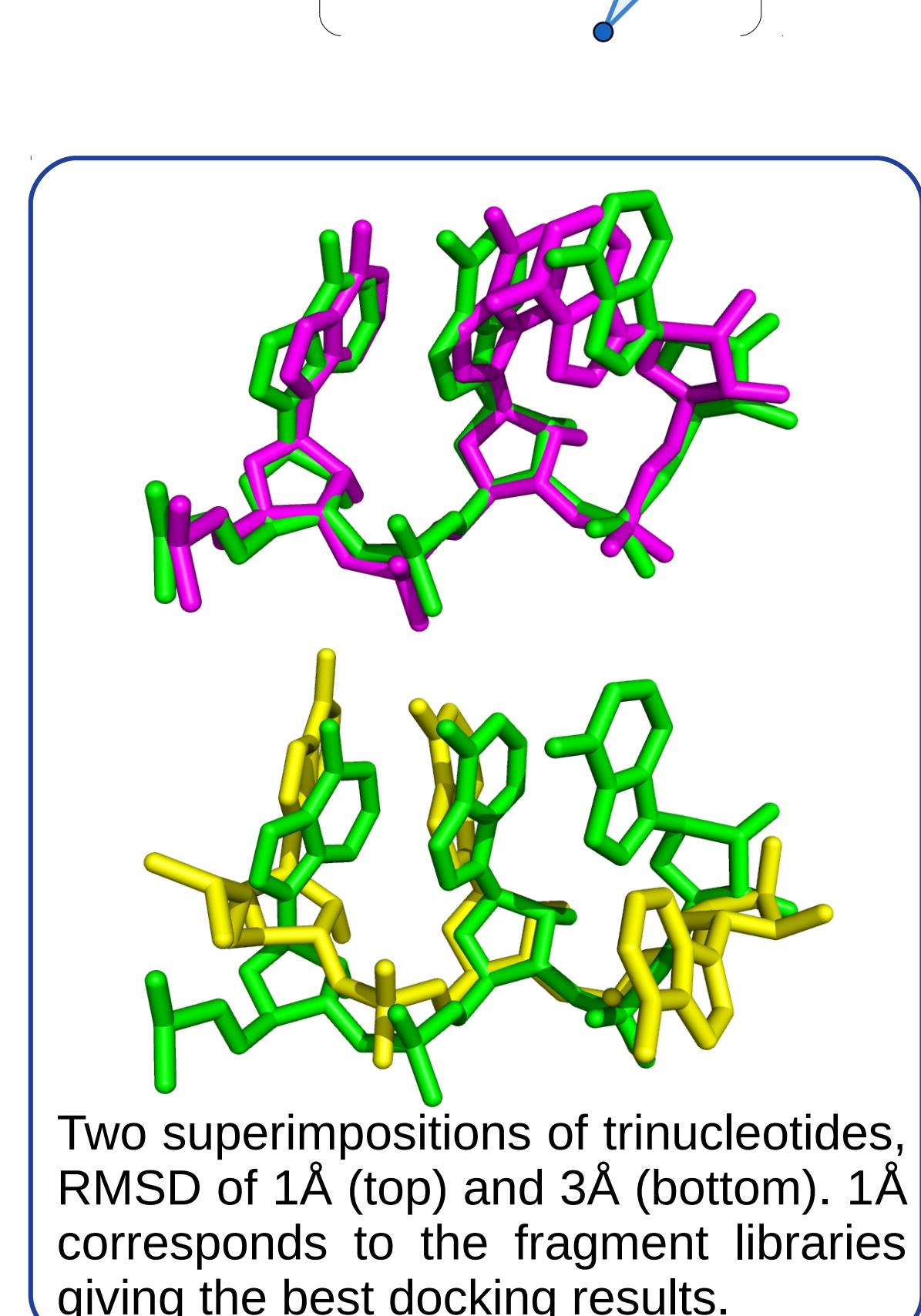
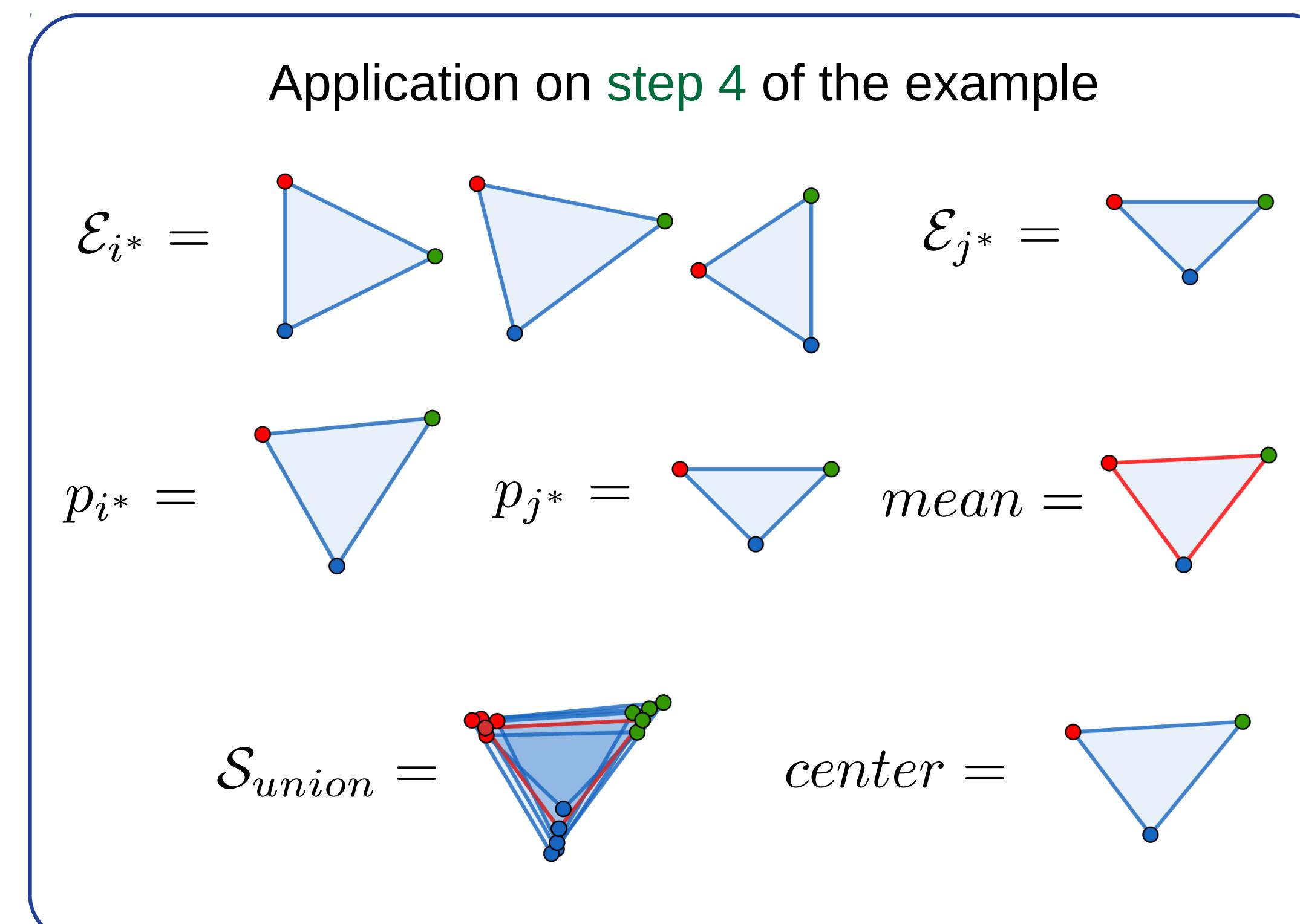
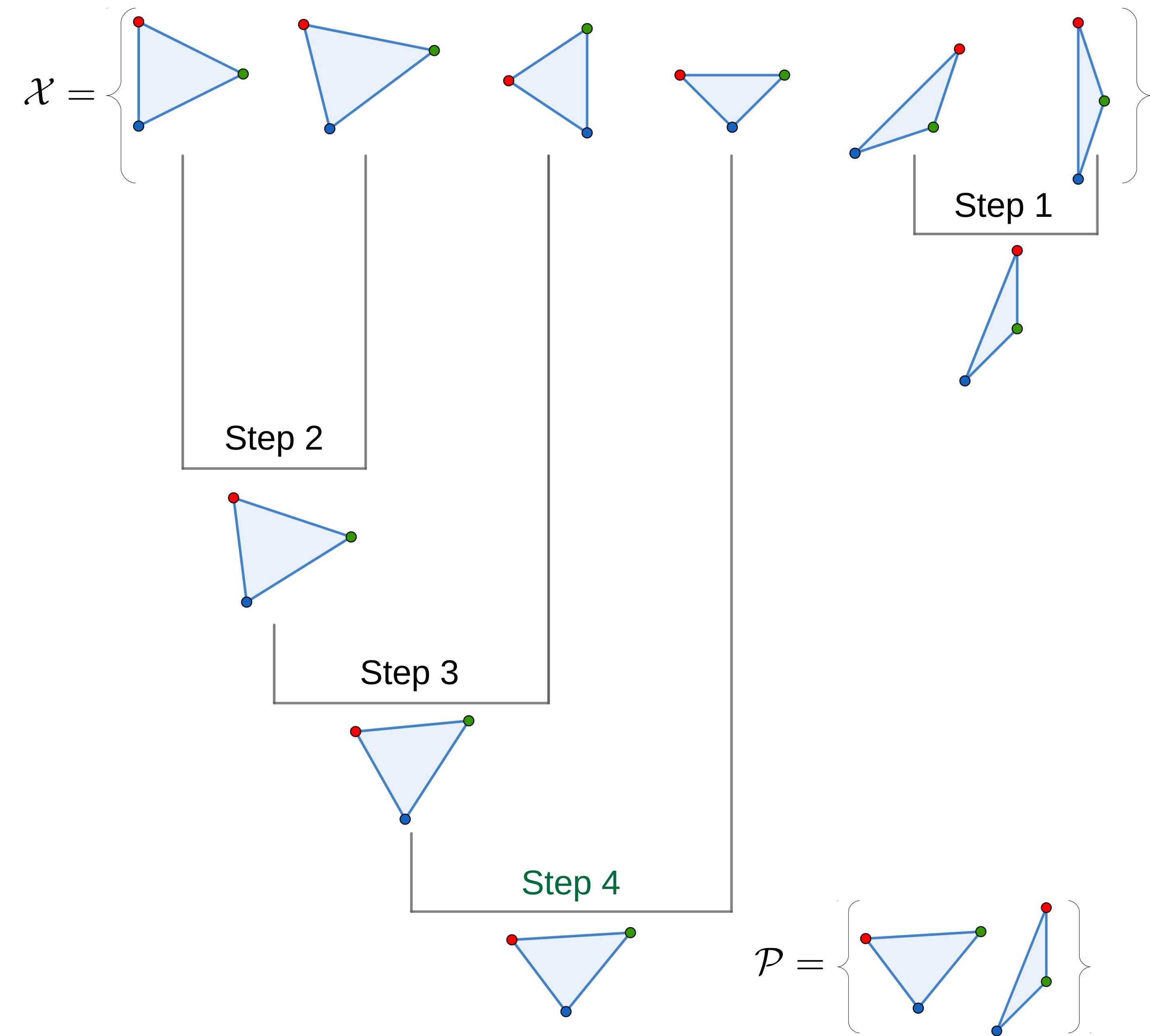
## Algorithm:

```

Input :  $\mathcal{X} := \{x_i : 1 \leq i \leq n\}$  # set of conformers
        $t > 0$  # threshold value
Output:  $\mathcal{P} := \{p_i : 1 \leq i \leq s\}$  # set of prototypes
Initialisation:
 $s := n$  # number of prototypes
 $\mathcal{P} := \mathcal{X}$  # set of prototypes
 $R := (r_{i,j}) \in \mathbb{R}^{s \times s}$  # matrix of RMSD of prototypes after
# one-against-one superimposition
 $R^{\text{mask}} := (r_{i,j}^{\text{mask}}) \in \{0, 1\}^{s \times s}$ 
 $\mathcal{E} := \{\{x_i\}_{1 \leq j \leq s} : 1 \leq i \leq n\}$  # set of clusters
 $\text{fusion} := \text{True};$ 
while  $\text{fusion}$  do
     $\text{fusion} := \text{False};$ 
     $R^{\text{mask}} := \text{False};$  # initialized at False
     $\text{tag} := \text{False};$ 
    while (not tag and not fusion) do
         $i^*, j^* := \underset{1 \leq i < j \leq s}{\operatorname{argmin}}(r_{i,j} : r_{i,j}^{\text{mask}} == \text{False});$ 
         $\text{mean} := (p_{i^*} + p_{j^*})/2;$ 
         $\mathcal{S}_{\text{union}} := \mathcal{E}_{i^*} \cup \mathcal{E}_{j^*};$  # superimposed on mean
        Compute the center of the smallest englobing ball associated
        with  $\mathcal{S}_{\text{union}}$ ;
        if ( $\max_{x_i \in \mathcal{S}_{\text{union}}} (d_{\text{superimposed}}(x_i, \text{center})) \leq t$ ) then
             $\mathcal{P} \setminus \{p_{i^*}, p_{j^*}\} \cup \{\text{center}\};$ 
            Update  $R$ ;
            Update  $\mathcal{E}$ ;
             $s := s - 1;$ 
             $\text{fusion} := \text{True};$ 
        else
             $r_{i_{\min}, j_{\min}}^{\text{mask}} := \text{True};$ 
            if ( $\forall 1 \leq i < j \leq s : r_{i,j}^{\text{mask}} == \text{True}$ ) then
                |  $\text{tag} := \text{True};$ 
            end if
        end if
    end while
end while
return  $\mathcal{P};$ 

```

## Example:



## Analysis:

The algorithm is based on the HAC [1], the main difference is the linkage method. We defined a linkage method which is the “smallest enclosing balls”. The computation of those balls is a quadratic problem and to solve it we are using the Frank-Wolfe algorithm [3].

The algorithm was applied to trinucleotides of RNA. We compared the results obtained, for the sequence AAA, to a classic HAC with the “complete” linkage method. For the “complete” linkage we obtain 2817 clusters, and with our “smallest enclosing balls” we obtain 2421 prototypes (coming from a set of 11,813 conformations). The difference is that the “complete” linkage is not adapted to calculate  $\varepsilon$ -nets, it is merging clusters when all members are at maximum 1 Å from other members, and it does not give any prototypes.

In the fragment-based method for the docking of RNA on protein [2], the docking of the conformers is proportional to the number of prototypes.

## Conclusion and ongoing research:

- Our method performs better than the state-of-the-art one;
- On the application of trinucleotides, the cardinality directly translate into a gain in computation time;
- We are planning to apply our method to other biological problems;
- Statistical analysis is in progress to derive generalization error bounds and excess risk bounds.

## References:

1. AK. Jain, and RC. Dubes, Algorithms for clustering data. Prentice Hall, Englewood Cliffs, 1998.
2. I. Chauvet de Beauchene, S. de Vries, and M. Zacharias. Fragment-based modelling of single stranded RNA bound to RNA recognition motif containing proteins. NucleicAcidsResearch, 44(10):4565-4580, 2016.
3. M. Frank, and P. Wolfe. An algorithm for quadratic programming. Naval Res. Logist. Quart., 3(1):95-110, 1956.