**Exercise 1**: **Just to get started**  
Write a python script that:  
\_ calculates the center of mass (**COM**) of a protein (PDB file) and writes its 3 coordinates in the terminal  
\_ writes an output file in pdb format containing a single atom placed at the center of mass.  
\_ Calculates the radius of gyration of the protein.

Feel free to use libraries, but you don’t have to.

**Exercise 2**: **Manipulate coordinates, compare 3D structures**  
We did a docking of RNA on a symmetric dimer of protein, with the software ATTRACT. To keep only the most representative positions, we clustered the poses based on their pairwise distance ([RMSD](https://bioinfo-fr.net/comparaison-de-structures-le-rmsd)) and kept the centroid of each cluster.

Each monomer is a chain (G or H) in receptor.pdb.

Among the resulting docking poses, some are symmetrical, and therefore redundant. We want to eliminate these redundancies. We will consider that a pose p1 within 2 angstrom RMSD from either another pose p2 or its symmetric p2’ is redundant, and we keep only one of (p1, p2). You can visualise examples of redundant poses by opening pose1-pose2.pse with pymol (free software).

Download ATTRACT from <http://www.attract.ph.tum.de/services/ATTRACT/attract.html> >menu >help and documentation  
  
The poses are written in result.dat, in a “degrees of freedom” DOF format described in the ATTRACT manual ($ATTRACTDIR/manual.txt, but this manual is work-in-progress). Each pose is defined by 2 lines, each line containing the 3 angles of rotation and 3 translation distances to apply to the original position of the receptor (1st line) and ligand (2nd line). In this example, the receptor was fixed.

The 3 angles of rotation are Euler angles. See the documentation in $ATTRACTTOOLS/euler2rotmat.f and $ATTRACTTOOLS/euler2rotmat.py on the Euler angle convention used in ATTRACT.

ATTRACT is a package containing docking scripts, but also many tool scripts for coordinates processing.

To get a bit familiar with ATTRACT, you can read the web interface paper, fill in a default example (the xylanase demo) in the web-interface, then download and study the generated bash script. But this is not strictly required by this exercise. You can also look at the docking script docking.sh

*!! The link above is for the simplified version of the web-interface that docks only proteins. Don’t test RNA on it.*

The most important command line tool is $ATTRACTDIR/collect, which converts a DOF file + PDB templates (1 per molecule) into a PDB containing the docking result.

$ATTRACTDIR/collect poses.dat receptor.pdb ligand.pdb > poses.pdb

*To write only the position of the ligand and not of the receptor, you can replace receptor.pdb by /dev/null. This is useful when the receptor is fixed, to avoid writing huge files that pymol could have difficulties to read.*

It is important to understand what happens exactly (in *collect* or during docking) when the DOFs are applied to the coordinates of a molecule. In the first lines of the DOF file, the "pivot" parameter defines the rotation pivot (rotation center):

\_ "pivot 0 0 0" for the global origin

\_ "pivot auto" for the center-of-mass (COM) of the molecule.

In the next lines, "centered false" means that all translations are relative to the original template coordinates.

The DOFs are applied as follows: the pivot is subtracted from the coordinates, the coordinates are rotated using the Euler angles, then the pivot is added back (=> same as rotating the molecule around its COM, but easier for computations), and finally the translation is applied.

You can use the following tools to apply operations to DOF files. Documentation is scarce, you may have to study the source code.

*$ATTRACTTOOLS/euler2rotmat.py*. Takes three Euler angles (as they are written in a DOF file) and returns the corresponding 3x3 rotation matrix.

*$ATTRACTTOOLS/euler.py* and *$ATTRACTTOOLS/detect-symmetry-axis.py*. These can be used to fit one monomer onto the other, and get either the corresponding rotation-translation numbers, or the symmetry axis.

*$ATTRACTDIR/axsymmetry* : You can give a symmetry axis and apply the symmetry on one molecule to generate the other(s). See usage in *axsymmetry.cpp*. The "symmetry" parameter should be 2.

*$ATTRACTDIR/fastcluster.py file.dat ligand.pdb x > clustx*

This will cluster the poses of file.dat based on their pairwise RMSD with an x angstrom cutoff. All poses in a cluster are at less than x A from the center of the cluster (and less than 2x A from each other).

Each line of *clustx* contains 1 cluster and the list of indices of the poses in that cluster.

**Exercise 3: Graph analysis**

The .npz file contains all connections between connected poses of RNA trinucleotides, i.e. all pairs of trinucleotides with a low overlap RMSD between nucleotides 2-3 of trinucleotide 1 and nucleotides 1-2 of trinucleotide 2 (See schema.pdf). Here the RNA is a homopolymer, so all trinucleotides are actually equivalent.

The exercise is to count all possible RNA chains of 7 nucleotides, i.e. that consist of 5 overlapping trinucleotides. You only need to count, not to enumerate every individual chain. To avoid the enumeration of all chains, use dynamic programming.

I added the beginning of a python script to read the .npz file. It is a dictionary. “interface-x” contains the list of all duplets of connected poses (their index, not their coordinates, which you don’t need) at (x+1)th and (x+2)th position in the RNA sequence (= in the chain).

Bonus question: Think of an algorithm to enumerate all chains that have a average rank of the poses under a certain threshold.