Farhad Ramezanghorbani

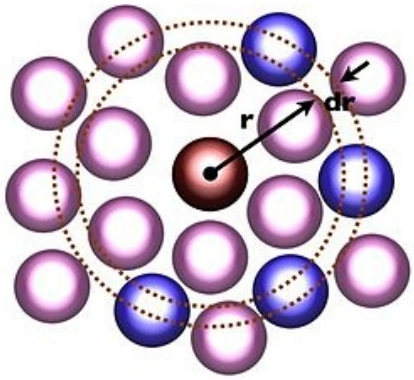
Exploring a given Molecular Trajectory:

Lets say that we have two input files of a molecular system containing N molecules:

1. A .pdb or .gro (structure file of Gromacs) file which is a structure file containing positions and names of atoms and residues and in some cases their velocities (.gro).

2. A trajectory file of a simulation containing time and positions of each step. This trajectory is actually an output of A simulation in which these N molecules will make some aggregate during the whole system equilibration.

We can see this equilibration procedure by these inputs if we use any visualizer tool like VMD. Now with this given information we want to investigate two main purposes:



First, for a given structure file we want to obtain its Radial Distribution Function. The radial distribution function (or RDF) is an example of a pair correlation function, which describes how, on average, the atoms in a system are radially packed around each other (how density varies as a function of distance from a reference particle). This proves to be a particularly effective way of describing the average structure of disordered molecular systems such as liquids.

Second part of the project is a survey to find out about the clusters which are results of the aggregations. In this case we should be able to compute some variables like average size of clusters (number of molecules in each aggregate), cluster size distribution and origin of each molecule in each aggregate. We can see these properties in VMD.

For example by assigning different colors to each molecule we can see their behavior during the simulation, and final location of them (in which cluster). For this part of the project we need to use MDAnalysis library of python.

MDAnalysis Library:

MDA can read molecular dynamic trajectories and access the atomics coordinates generated by Gromacs, NAMD, CHARMM, LAMMPS or Amber.

MDAnalysis is a Python package that provides classes to access data in molecular dynamics trajectories. It is object oriented so it treats atoms, groups of atoms, trajectories, etc as different objects. Each object has a number of operations defined on itself (also known as “methods”) and also contains values describing the object (“attributes”)

For example an AtomGroup object has a centerOfMass() method that returns the center of mass of the group of atoms. For example I am looking for center of mass of molecules for the first part to calculate radial distribution function.