Molecular Modeling - Midterm project
MD simulation of KRAS

Katerina Mastoraki

University ID: 7115152100031

1 KRAS biology and importance

KRAS (Kirsten rat sarcoma virus) gene is responsible for the synthesis of K-RAS protein. The K-RAS protein is part of the RAS/MAPK pathway, which communicates a signal from a receptor of a cell to its nucleus. These signals can either direct the cell to proliferate (grow and divide) or to differentiate (do specific functions). The K-Ras protein is a GTPase, which means it converts a molecule called GTP into another molecule called GDP. Essentially, K-Ras protein acts like an on-off switch for cells' growth, according to whether it's bind to GTP or GDP. In healthy cells, K-Ras protein is activated by binding to a GTP molecule, and gets deactivated, once the GTP converges to GDP. However, when there is a mutation at KRAS gene, the protein can become stuck on on-position, thus letting the cells grow uncontrollably and activating the downstream pathways. This leads to multiplication of cells and cancer growth, which can then cause metastases [3]. As you can see on figure 1, KRAS has a protein core, consisting of six beta-sheets, surrounded by five alpha-helices. This area of the protein is known as G domain. In addition there is a flexible C-terminal structural element, named hypervariable region (HVR), which is responsible for fastening KRAS into the membrane, where the signaling takes place. Other important areas of KRAS are the two switches, switch I and switch II. These switches form the binding surface for the proteins, as well as the regulators of KRAS [6]. On figure 1 you can also see the Guanosine-5'-Diphospate (GDP 201) ligand and the MG 202 magnesium ion, both bound to chain A1 of the mutated protein [5]. The GTP ligand is the chemical signal that binds to the protein and regulates its function, e.g : proliferation, apoptosis. Ligand binding can cause fluctuations of the free energy of the protein, thus change its conformations and its frequency.

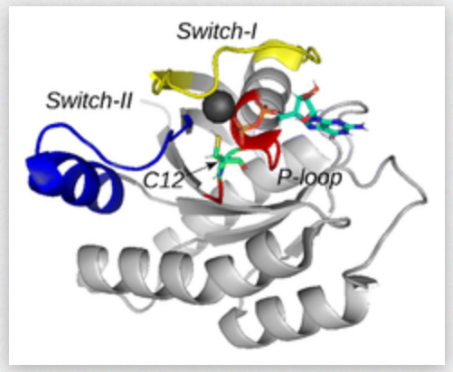


Figure 1: KRAS structure

KRAS mutations underlie the pathogenesis of up to 25% of human cancers [7]. KRAS usually becomes oncogenic by a mutation in codon 12 which is coding glycine (G12) and the most common KRAS mutants, caused by the previous codon mutation, are the G12C, G12D and G12R. KRAS drives 32% of lung cancers, 40% of colorectal cancers, and 85% to 90% (!) of pancreatic cancer cases This makes KRAS the most common gene that is linked to cancer, when mutations take place. As we can easily understand from the aforementioned evidence, KRAS is an extremely important gene and its study is crucial for making a big step closer to treat cancer.

2 KRAS description and simulation preperation

2.1 KRAS representation on VMD

Once the 4LDJ.pdb file is loaded to the VMD, you will get the representation shown on figure two, that depicts the G12C KRAS mutant protein. On the representation you can see the secondary structure of the mutant, as described on the previous section, as well as the ligand and the magnesium of the protein. As we already discussed on the previous section, the G12C KRAS mutant consists of:

- the protein core, which is created by 6 beta sheets
- 5 alpha helices that surround the protein core
- random structure elements, called loops
- a ligand - GDP GUANOSINE-5'-DIPHOSPHATE
- a magnesium - MG MAGNESIUM ION

Figure 2: G12C KRAS mutant representation on VMD

Also, by opening the 4ldj.pdb file, we can see that the file contains the "CRYSTAL STRUCTURE OF A GDP-BOUND G12C ONCOGENIC MUTANT OF HUMAN GTPASE" and the method which was used to export the data was X-ray diffraction. Moreover, there are 170 residues on this protein. The polypeptide chain forming this protein consists of all the amino acids, except two: tryptophan (TRP) and selenocysteine (SEC).

2.2 KRAS MD simulation preparation

Now we will proceed with the preparation of the protein for the MD simulation, using the NAMD tool.

- The first step is to remove the crystal waters. We can do that either with a notepad or through VMD, using the Tk console. First we should load the 4ldj.pdb file into VMD. Once the protein is loaded we should use the following command to create the 4ldj_noW.pdb file:

  set protein [atomselect top protein]

  $protein writepdb 4ldj_noW.pdb

- The second step on preparing the protein for minimization is to produce the .psf file. A PSF file, also called a protein structure file, contains all of the molecule-specific information needed to apply a particular force field to a molecular system. I created the file using the AutomaticPSFBuilder, a VMD tool. In order to do that, I provided the 4ldj_noW.pdb file and the topology files : top_all36_carb.rtf, top_all36_cgenff.rtf, top_all36_lipid.rtf, top_all36_na.rtf and top_all36_prot.rtf. As output I got the 4ldj_autopsf.pdb file and the 4ldj_autopsf.psf file.

- The third step is to solvate the protein. Solvating means that we add water, in order to resemble the cellular environment. I created a water box so that there is a layer of water 10 Å in each direction from the atom with the largest coordinate in that direction. To do that, I opened Tk console on VMD and I ran the following commands :

  package require solvate

  solvate 4ldj_autopsf.psf 4ldj_autopsf.pdb -t 10 -o lysozyme_water

  From the previous command I received 3 output files : kras_water.log, kras_water.pdb and kras_water.psf.

- The fourth step is to add ions to the system in order to neutralize it and make it more stable. To do that I ran on Tk console the following commands:

  package require autoionize

  autoionize -psf kras_water.psf -pdb kras_water.pdb -sc 0.15 -cation SOD

  From these commands I finally received the ionized.pdb and ionized.psf files that I used on NAMD simulation.

- It is also needed to calculate the coordinates of the centre of the water box as well as the maximum and minimum coordinates of the entire protein-water system. To do that I used the following commands on Tk console :

  set everyone [atomselect top all]

  measure center $everyone

  measure minmax $everyone

2.3 Minimization of KRAS

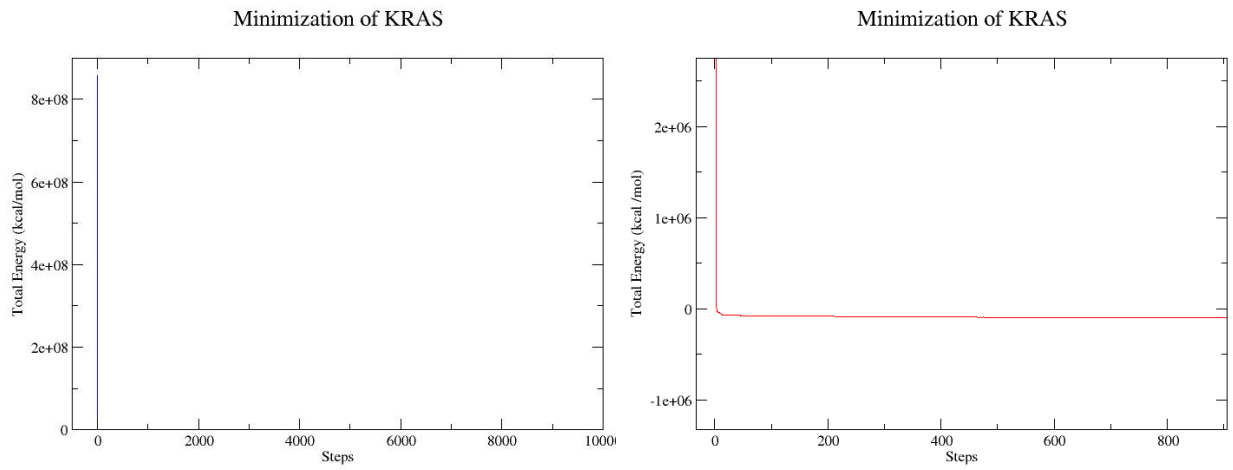
At this point I am ready to start the simulation. To do that first of all I loaded the topology and parameter files on the 1.Building_system folder inside NAMD tutorial directory and then I loaded the parameter files, the ionized.pdb and ionized.psf on 2.Minimization folder. I also created the configuration file for the minimization, that receives as input the previous files and outputs the various kras_wb_mini files, that contain the minimization. I used the coordinates of the centre of the water box, that I calculated through VMD as values for cellOrigin variable at the configure file and the min and max coordinates for the cellBasisVector1, cellBasisVector2 and cellBasisVector3. I used 10.000 steps for the minimization process. To run the simulation and create the log file, I used the following command:

namd2 kras_mini.conf > kras_mini.log

Once the simulation finished, I created the .dat file that contains the values for every different parameter for every simulation step with the following command :

grep 'ENERGY:' kras_mini.log > kras_mini.dat

The total energy versus minimization step are shown on figure 3.



(a) Total Energy - time step (b) A closer look at the total energy - time step PLOT

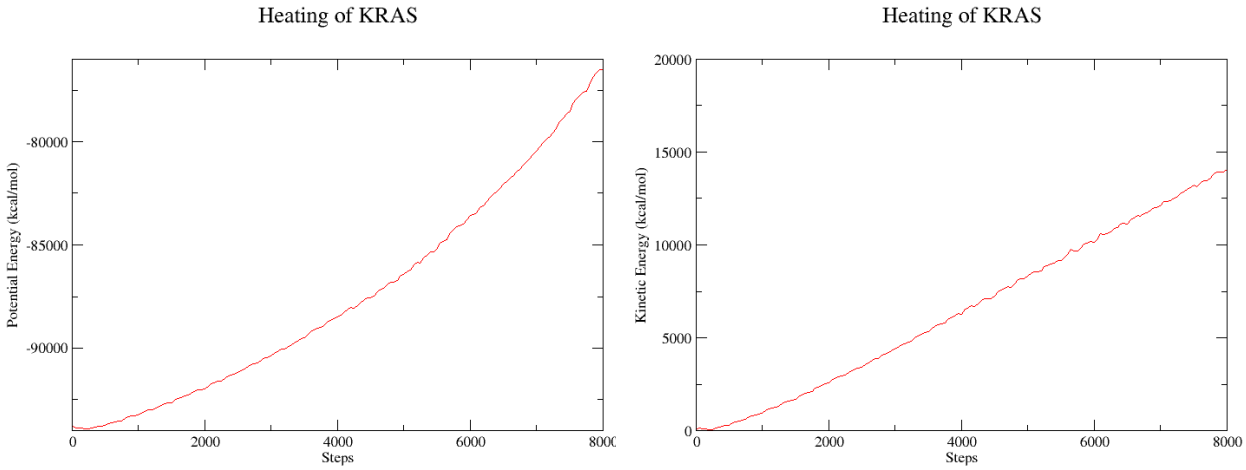
Figure 3: Minimization of KRAS

3 Heating and Equilibration of KRAS

3.1 Heating

First we have to create the configuration file for the heating phase. I created this file, modifying the configuration file of minimization phase. In the heating phase, the system starts with 0 kelvin temperature on the first step. Energy is given to the system gradually, with the type of heat. We can't heat the system at once, because that would cause to instability, something that would lead to the unsuccessful termination of the MD simulation. The system would gradually reach 310

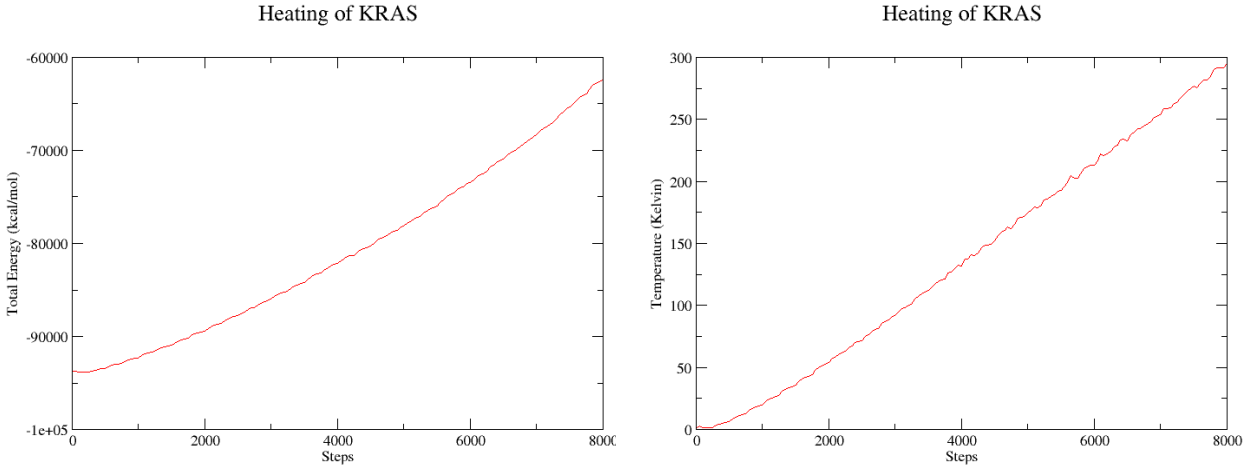
Kelvin degrees. More specifically, the total steps for this simulation are 8.000 and energy is given to the system every 50 steps (0.1ps). The plots of the kinetic and potential energy are shown above (figure 4). Also figure 5 shows how the total energy and the temperature fluctuate through heating process.



(a) Potential energy - time step

(b) Kinetic energy - time step

Figure 4: Heating of KRAS



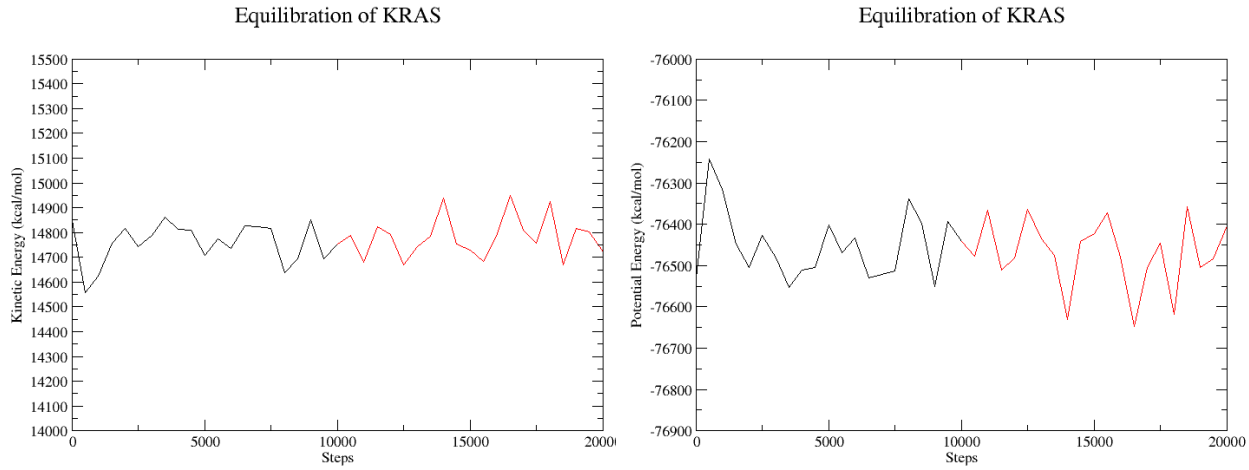
(a) Total energy - time step

(b) Temperature - time step

Figure 5: Heating of KRAS

3.2 Equilibration

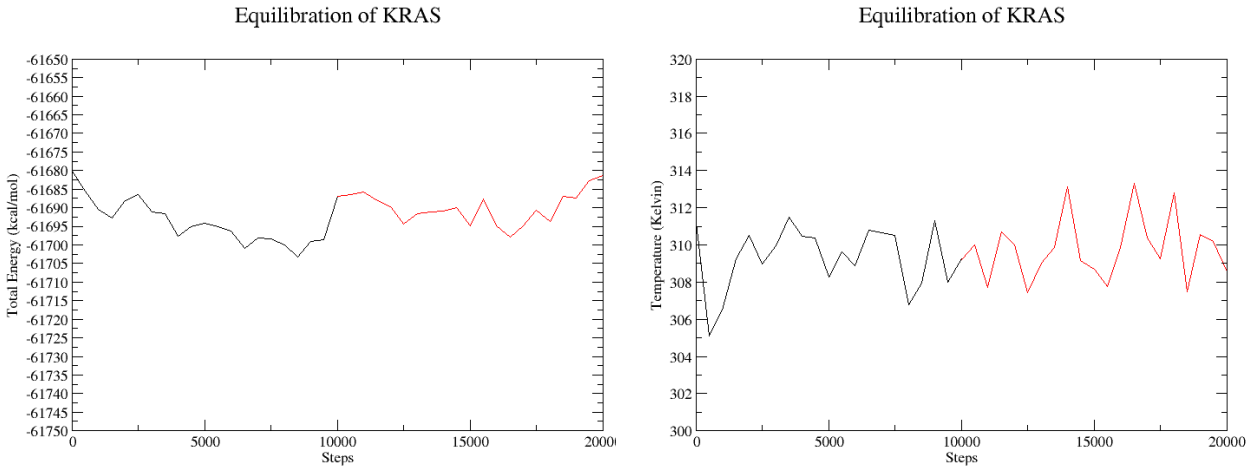
Now, we are ready to proceed to the equilibration phase of our simulation. The equilibration phase is mandatory, due to the fact that we heated the system in a constant manner, thus we have to stabilize it. We asses that the system has reached thermodynamic equilibrium, once its energy has finally reached a plateau. At the first round of equilibration I ran the simulation for 10.000 steps. I continued with another 10.000 steps on the second round of equilibration (20.000 steps in total), despite the fact that the system was already pretty stable. The plots of the two different stages of the equilibration are shown on figure 6 and 7.



(a) Kinetic energy - time step

(b) Potential energy - time step

Figure 6: Equilibration of KRAS



(a) Total energy - time step

(b) Temperature - time step

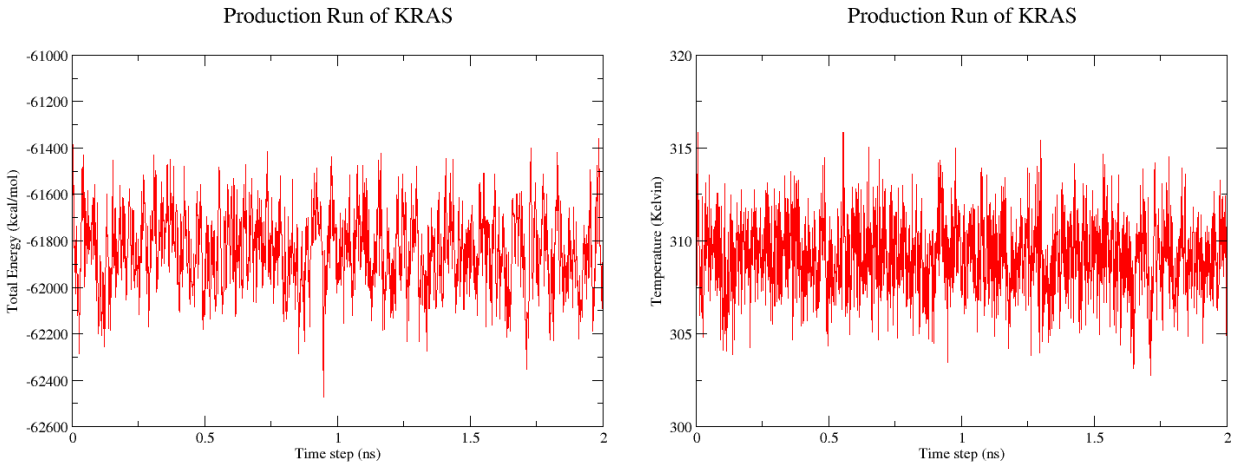
Figure 7: Equilibration of KRAS

4 Equilibrium assessment

As you can see on both figure 6 and figure 7 the system has reached thermodynamic equilibrium already by the first 10.000 steps of the simulation. However for the sake of the project I did a second equilibration run with another 10.000 steps. The system now is stable, with some expected fluctuations around the mean value for the temperature and almost linear function for the total energy.

5 Production Run

First of all we have to create the configure file with the same way as before, in order to run the production run and be able to start the system's analysis. I wasn't able to run the simulation for 10ns, as that would require 5.000.000 steps and almost a week on my computer. Instead, I ran the production run for 1.000.000 steps (2ns), which is a sufficient time period, as stated on lecture. Once the simulation is over, the file kras_wb_prod.dcd is produced. This is the file that we will use for the analysis. The total energy and temperature versus step plots are shown on figure 8.



(a) Total Energy - time step

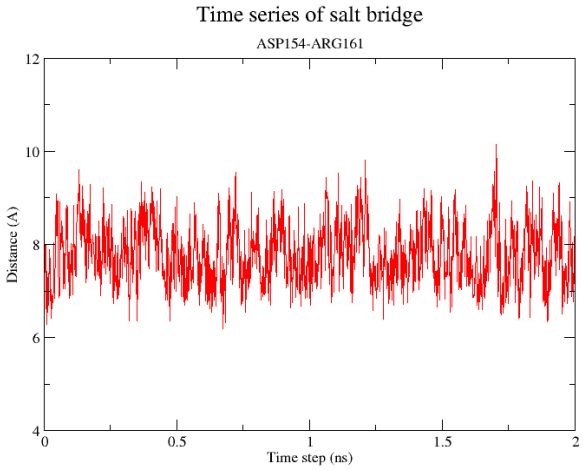
(b) Temperature - time step

Figure 8: Production run of KRAS

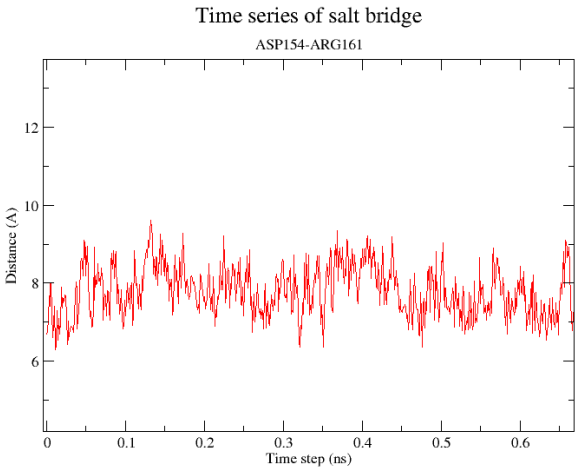
6 Trajectory analysis

6.1 Salt bridge

A salt bridge is a bond that is formed between amino acids with opposite charges, when at least two heavy atoms lie in a hydrogen bond distance and they are primarily found in solvated proteins [8]. This bond is usually formed between the anionic carboxylate (RCOO) of either aspartic acid or glutamic acid and the cationic ammonium (RNH3+) of lysine or the RNHC(NH2)2+ of arginine [4]. To find the salt bridges that are formed at the protein, I used the Salt Bridges tool of VMD. This tool produces .dat files for every salt bridge on the molecule with distance below a specific threshold. I loaded to VMD the kras_wb_prod.dcd file and the ionized.psf file and I set the distance cutoff to 5A. The salt bridge of ASP154 and ARG161 is formed between the OD2 oxygen of aspartic acid and the NH1 of arginine, both of which are part of the side chains of each amino acid. The time series plot of the specific salt bridge, is shown at figure 9.



(a) Plot of salt bridge for the whole production run (2ns)



(b) A closer look at the salt bridge plot

Figure 9: Time series plot of ASP154 and ARG161 salt bridge

6.2 Movie of the simulation

The movie of the simulation is the kras.mpg, which is enclosed in the project folder.

6.3 RMSD

The RMSD can be calculated using mdtrajanalysis package, in the same way as we did on lecture. The RMSD for KRAS is shown on the figure 10. (Code on jupiter notebook, enclosed to the folder)

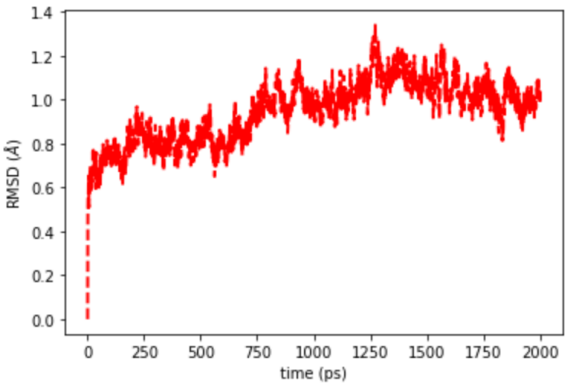


Figure 10: RMSD of KRAS

6.4 Flexible parts

As we already know, proteins are created with peptide bonds, that are formed between amino acids. The peptide bond is rigid, thus the backbone of a protein has not much flexibility, in contrast with its outer part ,that is formed by the side chains of the various amino acids. So, the side chains of the amino acids, form the most flexible part of the protein. Moreover, as we already discussed, there is the hypervariable region (HVR), which is a flexible C-terminal structural element of KRAS mutant and plays a key role on stabilizing KRAS on the membrane [6]. Finally, based on root-mean-square fluctuation (RMSF) values, switch regions of KRAS seem to have high flexibility [7].

6.5 First cluster representative

First and foremost, we should align the dcd file, using the VMD tool. Once the file is aligned, it is time to perform hierarchical clustering, in the same way as we did on lecture (using the average linkage algorithm). For 2000 frames (2ns) of production run, the dendogram of the clusters is shown on figure 11.

From the above plot is easy to infer that there are 4 clusters on our system, while the most populated one is the second cluster. The next step is to calculate the conformations, which are included to each cluster. To do that I used the fcluster function [2]. This function allow us to form flat clusters from the hierarchical clustering, with a specific maximum distance. In the specific case, in order to receive 4 clusters, I used 0.7 A cutoff. Then, the final step is to calculate the centroid of the second cluster (most representative structure of the cluster). To do that, I used the sample code for find centroids [1]. Finally, the most representative structure on my simulation is the one with index 729. You can see the code used for calculating the first cluster representative, on the jupiter notebook that I am sending you.

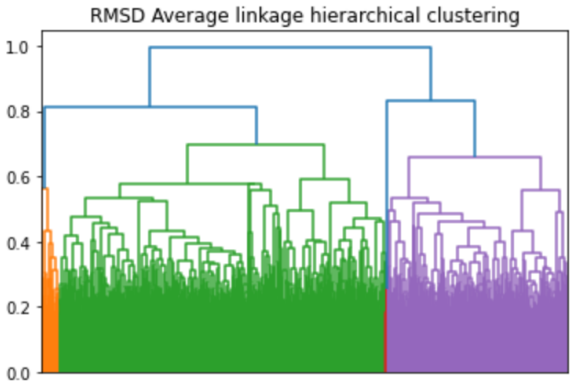


Figure 11: Clusters of KRAS

References

[1] Finding centroids. https://www.mdtraj.org/1.9.5/examples/centroids.html.

[2] Flcuster. https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.fcluster.html.

[3] KRAS-Wikipedia. https://en.wikipedia.org/wiki/KRAS.

[4] Salt bridge-Wikipedia. https://en.wikipedia.org/wiki/Salt_bridge_(protein_and_supramolecular)#Salt_bridges_

[5] John C Hunter, Deepak Gurbani, Scott B Ficarro, Martin A Carrasco, Sang Min Lim, Hwan Geun Choi, Ting Xie, Jarrod A Marto, Zhe Chen, Nathanael S Gray, et al. In situ selectivity profiling and crystal structure of sml-8-73-1, an active site inhibitor of oncogenic k-ras g12c. *Proceedings of the National Academy of Sciences*, 111(24):8895–8900, 2014.

[6] Tatu Pantsar. The current understanding of kras protein structure and dynamics. *Computational and structural biotechnology journal*, 18:189–198, 2020.

[7] Tatu Pantsar. Kras (g12c)–amg 510 interaction dynamics revealed by all-atom molecular dynamics simulations. *Scientific reports*, 10(1):1–9, 2020.

[8] Svetlana Pylaeva, Martin Brehm, and Daniel Sebastiani. Salt bridge in aqueous solution: Strong structural motifs but weak enthalpic effect. *Scientific reports*, 8(1):1–7, 2018.