Metehan Ilter
Y3190002

Proteins, which are comprised of amino acids, are the building blocks of living organisms and play a pivotal role in signaling cascades. They are, indeed, important and well-scrutinized targets for drug discovery studies. From this perspective, it is crucial to have a holistic understanding of the different states of the proteins. To this end, a static structure of a protein, whose coordinates in 3-dimensional space are deposited as a protein data bank (PDB) file (Berman et al., 2000; Burley et al., 2021) , is modeled and simulated under the physiological conditions to have a closer look into possible states of the protein. Each motion of the modeled protein during a simulation is saved as a frame and deposited in the form of the dcd, namely a trajectory file. Eventually, the produced trajectory file gets deeply analyzed to shed some light on the dynamic properties of the simulated protein as well as its structural properties. In this way, different states are deciphered and the propensity of the protein to adopt those explored states is calculated by taking the ratio of frames that represent a state over the total number of frames (Ilter & Sensoy, 2019; Lu, Jang, Nussinov, & Zhang, 2016). Consequently, the frame that represents the most probable state of the protein is utilized for drug discovery studies (Amaro & Li, 2010).

As the significance of uncovering the distinct states of the proteins are stressed in the above, over the years, more accurate, precise, and sharp trajectory analysis approaches have come into prominence. For this, some reaction coordinates, such as an angle/a distance between certain atoms of amino acids, have been calculated by means of both theoretical and experimental approaches. With this motivation, in the scope of the project, **a script**, which **measures the distance between the selected C alpha (Cα) atoms of amino acids as well as in the case of having a higher standard deviation than a user-defined cut-off value, converts the time-line measured distance data into normalized histograms**, was aimed to be developed (See the source code shown in below).

As example input files, **the PDB file** pertaining to the physiologically relevant model to H-RAS protein and **its trajectory** were utilized for compiling the scripts. The reliability of the measured distances was assessed by comparing the measured distance between the same reaction coordinates with GROningen MAchine for Chemical Simulations (GROMACS)(Abraham et al., 2015) that is well-established tool, but not capable of converting raw data into normalized histogram plots. Eventually, it was shown that the developed script gave almost the same result to the above-mentioned tool (See Fig 1.) .

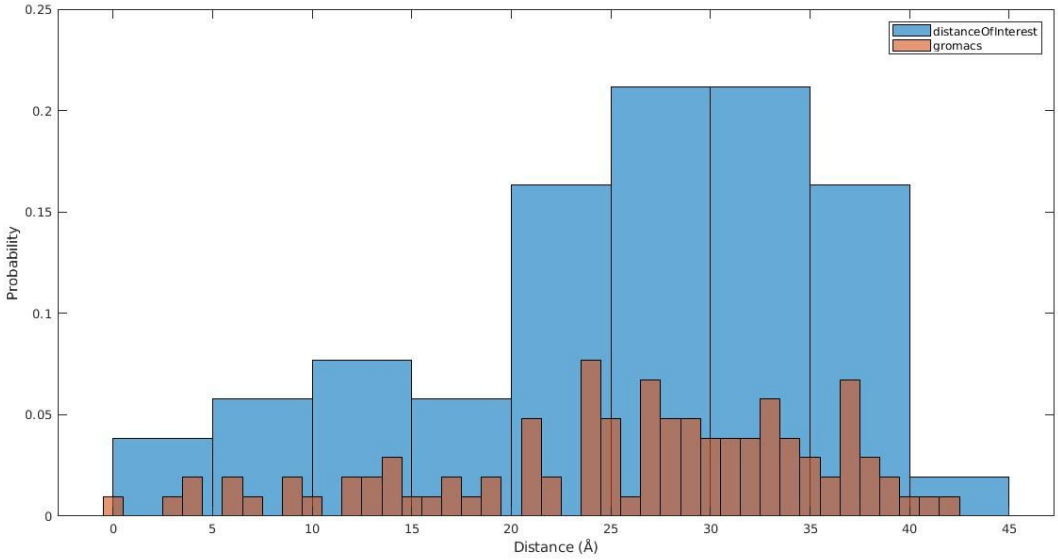

Figure 1. The distance measured between the 1st and 12th Cα atoms by the script and GROMACS were depicted as an superimposition of the obtained histograms.

Metehan Ilter
Y3190002

```
% Removing the header of a PDB file and saving the edited version as another PDB file.
pdb = fopen('protein.pdb', 'r');
fgetl(pdb);
buffer = fread(pdb, Inf);
fclose(pdb);
pdb = fopen('fullmodelmin_woheaders.pdb', 'w');
fwrite(pdb, buffer);
fclose(pdb);

%Reading the edited PDB file
pdb_woheader = fopen('fullmodelmin_woheaders.pdb', 'r');

%Scanning strings that correspond to x-, y-, and z-coordinates found in the 7th, 8th, and 9th columns, respectively. For further information, please check out the
following website. https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html
pdbFormat='%4s  %5d %4s%4s%1s%4d    %8.3f%8.3f%8.3f%6.2f%6.2f      %4s%2s';
A=textscan(pdb_woheader,pdbFormat);
x=[A{1,7}];
y=[A{1,8}];
z=[A{1,9}];
atoms = [A{1,3}];

%The total number of x-,y-, and z-coordinates has to be equal to each other.
if length(x) == length(y) && length(y) == length(z)
    fprintf('The coordinates has been succesfully read!\n')
    totalNumberOfAtoms = length(x)-1;
else
    fprintf('A severe problem has raisen!\n')
end

%Finding indeces of Cα atoms and assigning them to a matrix.
Calpha_atoms_windex = [];
for i=1:length(atoms)
    if atoms(i) == "CA"
        Calpha_atoms_windex = [Calpha_atoms_windex, i];
    else
        continue
    end
end
Calpha_atoms = Calpha_atoms_windex';

%The Cartesian coordinates of all Cα atoms in the given PDB file.
coordinates_of_Calpha_atoms = [];
i = 1;
while i <= length(Calpha_atoms)
    coordinates_of_Calpha_atoms = [coordinates_of_Calpha_atoms; x(Calpha_atoms(i)), y(Calpha_atoms(i)), z(Calpha_atoms(i))];
    i = i + 1;
end

%Measuring pair-wise distances of all Cα atoms within the read PDB file.
i = 1;
distance_matrix = [];
for i=1:length(coordinates_of_Calpha_atoms)
    for j=1:length(coordinates_of_Calpha_atoms)
    distance_matrix = [distance_matrix;sqrt(power((coordinates_of_Calpha_atoms(i,1) - coordinates_of_Calpha_atoms(j,1)),2) +
power((coordinates_of_Calpha_atoms(i,2)- coordinates_of_Calpha_atoms(j,2)),2) + power((coordinates_of_Calpha_atoms(i,3) -
coordinates_of_Calpha_atoms(j,3)),2))];
    end
end

%Making the matrix that deposits the measured distances n-by-n.
nbyn_distance_matrix = reshape(distance_matrix,[length(Calpha_atoms), length(Calpha_atoms)]);

%Illustrating the pair-wise distance as a heat map (See Figure 2).
h_distance = heatmap(nbyn_distance_matrix);
xlabel('Residue Number')
ylabel('Residue Number')
XYLabels = 1:length(Calpha_atoms);
string_XYLabels = string(XYLabels);
string_XYLabels(mod(XYLabels,10) ~= 0 ) = " ";
h_distance.XDisplayLabels = string_XYLabels;
h_distance.YDisplayLabels = string_XYLabels;

%Reading a trajectory file by using the readdcd function embedded in the MDToolBox package
trajectory = readdcd('theshortest.dcd');
sizeOftrajectory = size(trajectory);
numberOfcoordinates = sizeOftrajectory(2);
numberOfframes = sizeOftrajectory(1);

%Retrieving x-, y-, and z-coordinates of each frame allocated in the trajectory file and writing the coordinates into another matrix.
x_trajectory = []; y_trajectory = [] ; z_trajectory = []; organized_trajectory = [];
nframes = 1;
while nframes <= numberOfframes
    for n=1:1:numberOfcoordinates/3
        x_trajectory = [x_trajectory;trajectory(nframes,3*n-2)];
        y_trajectory = [y_trajectory;trajectory(nframes,3*n-1)];
        z_trajectory = [z_trajectory;trajectory(nframes,3*n)];
    end
    fprintf('Progress:%f\n', 100*nframes/numberOfframes)
    nframes = nframes+1;
end
organized_trajectory = [organized_trajectory;x_trajectory, y_trajectory, z_trajectory];

% Determining the indeces of each Cα atom per frame.
Calpha_atoms_trajindex = []; Calpha_traj = [];
for k = 0:numberOfframes-1
    [Calpha_atoms_trajindex] = [Calpha_atoms_trajindex; Calpha_atoms + k*totalNumberOfAtoms];
end
for i = 1:length(Calpha_atoms_trajindex)
```

Metehan Ilter
Y3190002

```
    [Calpha_traj] = [Calpha_traj;organized_trajectory(Calpha_atoms_trajindex(i),1),organized_trajectory(Calpha_atoms_trajindex(i),2),
organized_trajectory(Calpha_atoms_trajindex(i),3)];
end

%Inserting the indeces column next to the z-coordinates, where each index corresponds to amino acid number.
row_res_number=[];
for k = 1:numberOfframes
    row_res_number = [row_res_number,1:length(Calpha_atoms)*k/k];
end
column_res_number = (row_res_number)';
Calpha_traj_wresnumber = [Calpha_traj column_res_number];

flag = -1;
while flag < 0
%In order for a user to start/continue or terminate the search, one of the specified entries have to be given as an input.
    command = input('If you would like to start/contiue your search, please enter sc!. If not, please enter t\n', 's');
    if command == 'sc'

%If an user enters an invalid atom number, that user will be warned.
     atomnumber1 = input('Enter the first atom:\n');
     atomnumber2 = input('Enter the second atom:\n');
     if atomnumber1 > length(Calpha_atoms) || atomnumber2 > length(Calpha_atoms)
         disp('Invalid atom number')
         break
     end

%The entered atom number has to an integer.
     if isnan(atomnumber1) || fix(atomnumber1) ~= atomnumber1 && isnan(atomnumer2) || fix(atomnumber2) ~= atomnumber2
        disp('Please enter an integer')
     end

%Measuring the distance between the given atoms during the read trajectory.
     distanceOfInterest = [];
     coordinates_atomnumber1=[];
     coordinates_atomnumber2=[];
     for i = 1 : length(Calpha_traj_wresnumber)
        if Calpha_traj_wresnumber(i,4) == atomnumber1
            coordinates_atomnumber1 = [coordinates_atomnumber1; Calpha_traj_wresnumber(i,:)];
        end
        Calpha_traj_wresnumber(i,4) == atomnumber2;
        coordinates_atomnumber2 = [coordinates_atomnumber2; Calpha_traj_wresnumber(i,:)];
     end
     i = 1;
     uperdeterminants = size(coordinates_atomnumber1);
     upperboundary = uperdeterminants(1);
     while i <= upperboundary
         distanceOfInterest = [distanceOfInterest, sqrt(power((coordinates_atomnumber1(i,1) - coordinates_atomnumber2(i,1)),2) +
power((coordinates_atomnumber1(i,2)- coordinates_atomnumber2(i,2)),2) + power((coordinates_atomnumber1(i,3)- coordinates_atomnumber2(i,3)),2))];
         i = i +1;
    end

%Normalized histogram
%The cut-off entry has to be numerical.
    cutoff = input('Please enter the cut-off value:\n');
     if isnan(cutoff) || fix(cutoff) ~= cutoff
     else
         fprintf('Please, enter a numerical value');
         continue
     end

%If the standard deviation of the measured distance is above the entered cut-off value by an user, generate the histogram plot (See Figure 3).
     if std(distanceOfInterest) > cutoff
        figure(2);histogram(distanceOfInterest, 'Normalization', 'probability');
        xlabel('Distance (Å)')
        ylabel('Probability')
        title(['The distance between Calpha no: ',num2str(atomnumber1), ' and ', num2str(atomnumber2), ', where the cut-off=', num2str(cutoff)])

%If the standard deviation of the measured distance is below the cut-off, do not plot a normalized histogram
     else
         continue
     end

%In order to terminate the search, t has to be entered as an input.
    elseif command == 't'
        display('The search is terminated')
        break

%In the case of entering an undefined command, the search gets terminated.
    elseif command ~= 't' || command ~= 'sc'
        display('Please, enter a suitable command')
        break
    end
end
```

Metehan Ilter
Y3190002

Example outputs of the code

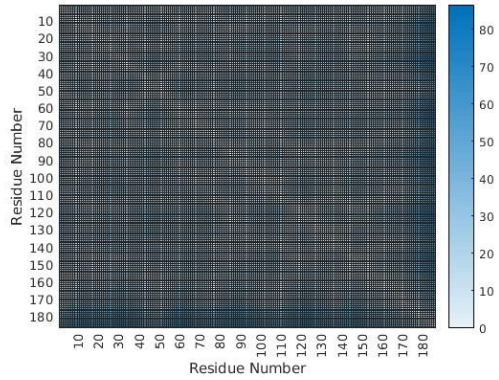

Figure 2. The measured pair-wise distance of the Cα atoms deposited in the PDB file.

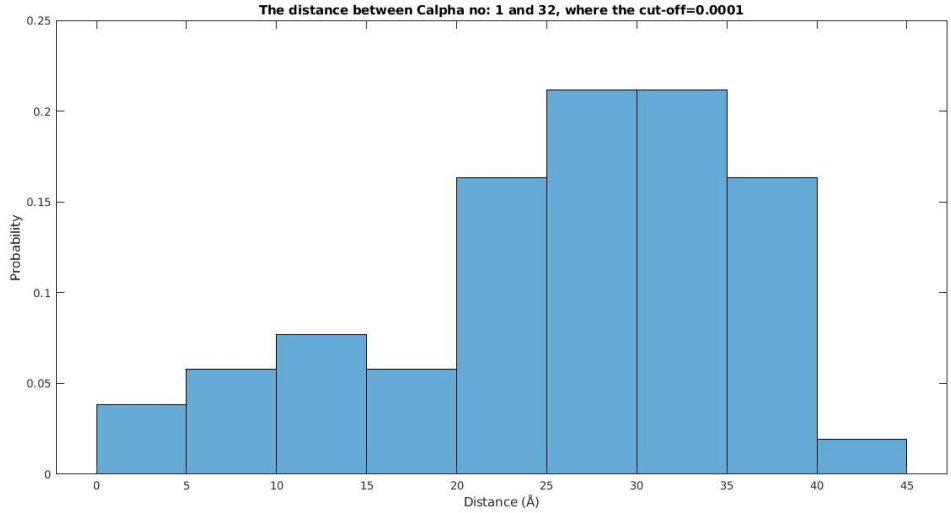

Figure 3. The normalized histogram that shows the probability distribution of the measured distance between the 1st and 32nd Cα atoms, where the user-defined cut-off was given as 0.0001.

References

Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., & Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, *1–2*, 19–25. https://doi.org/10.1016/J.SOFTX.2015.06.001

Amaro, R., & Li, W. (2010). Emerging Methods for Ensemble-Based Virtual Screening. *Current Topics in Medicinal Chemistry*. https://doi.org/10.2174/156802610790232279

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., … Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*. https://doi.org/10.1093/nar/28.1.235

Burley, S. K., Bhikadiya, C., Bi, C., Bittrich, S., Chen, L., Crichlow, G. V, … Zhuravleva, M. (2021). RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological

macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research*. https://doi.org/10.1093/nar/gkaa1038

Ilter, M., & Sensoy, O. (2019). Catalytically Competent Non-transforming H-RASG12P Mutant Provides Insight into Molecular Switch Function and GAP-independent GTPase Activity of RAS. *Scientific Reports*. https://doi.org/10.1038/s41598-019-47481-1

Lu, S., Jang, H., Nussinov, R., & Zhang, J. (2016). The Structural Basis of Oncogenic Mutations G12, G13 and Q61 in Small GTPase K-Ras4B. *Scientific Reports*, *6*(February), 1–15. https://doi.org/10.1038/srep21949