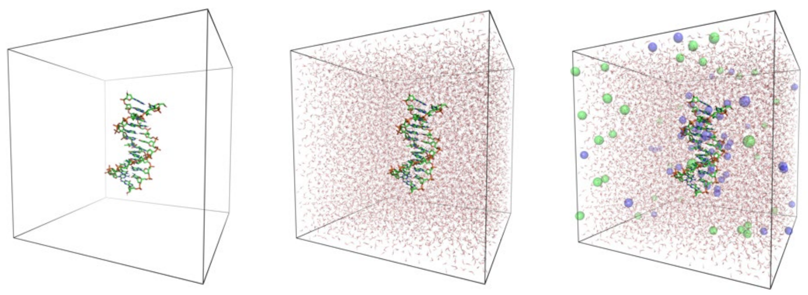
Project: Atomic Root-mean-square Deviation (RMSD) Calculation for Assessing Structural Integrity in Molecular Dynamics Simulations of DNA

Objective:

This project involved developing a simple script to analyze RMSD data from Molecular Dynamics (MD) simulations of DNA. The script reads RMSD data from plain text files using pandas. The primary output is a composite plot showcasing time series and distribution data for all five replicate simulations.

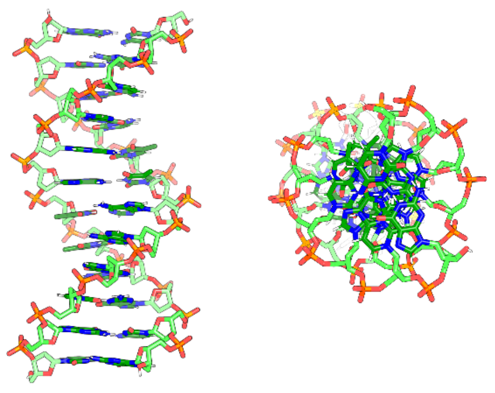
A Brief Overview of MD Simulations:

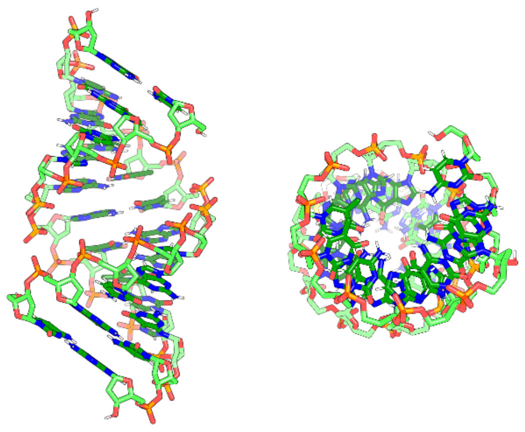
MD simulations typically involve a single solute, such as a protein or DNA, in a periodic box. This setup is analogous to the "Snake 2" game on a Nokia 3310, where crossing one edge of the screen leads to re-entry from the opposite side. The box size is critical to ensure that the DNA does not interact with its own ends. In these simulations, the solute is surrounded by water, either implicitly (simulating average water effects) or explicitly (adding thousands of water molecules). Modern simulations predominantly use the explicit method. The images below show (1) a DNA positioned in a box with a minimum distance to the wall of 1.4 nm, (2) the box filled with explicit water molecules, (3) added ions (Na+ and Cl-).



Data Description:

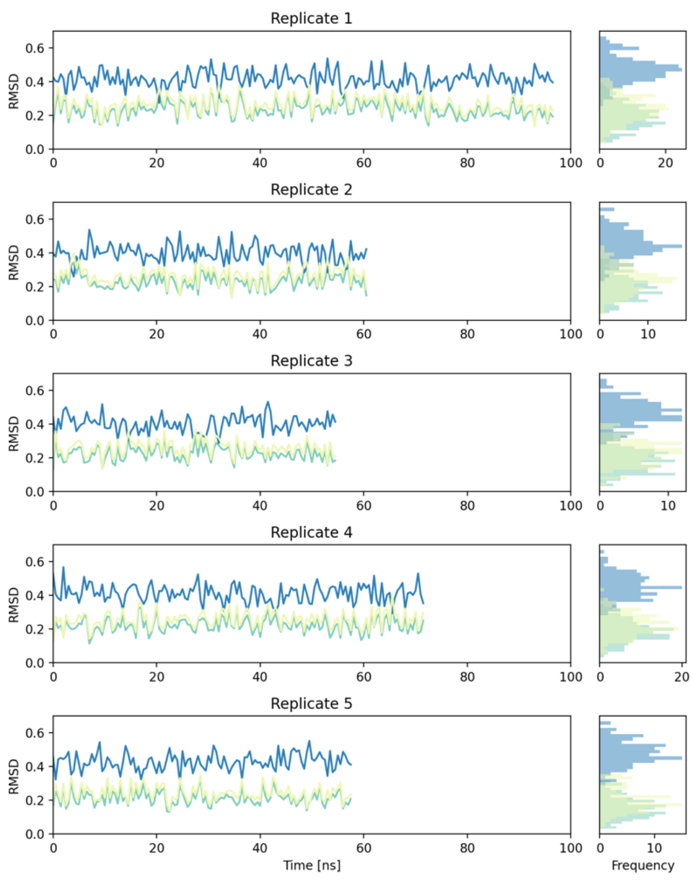
The focus of the simulation was a 12 basepair (bp) segment of double-stranded DNA, replicated in five separate simulations with varying initial velocities. Note: the simulations were not done at the time of finishing this assignment, therefore none of the time series actually reached the end of the simulation (100 ns). A key analysis post-simulation is the calculation of RMSD for each backbone atom, which quantifies the deviation from the initial DNA structure. RMSD was computed relative to three structures: the original structure at the start of the simulation, and two common helical DNA structures, A-DNA and B-DNA. While both A- and B-helical structures are prevalent, B-DNA is more commonly observed under physiological conditions. The images below show a side and topview for B-DNA on the left and for A-DNA on the right.





Results:

The resulting plot shows the RMSD values with respect to the three reference structures: the initial structure in lime green, the B-DNA structure in light blue, and the A-DNA structure in a deeper shade of blue. This visualization reveals that the initial structure (PDB code: 1NAJ) closely resembles a 'perfect' B-DNA helix. Furthermore, it's evident that the simulated DNA maintains a closer resemblance to the B-DNA helix than to the A-DNA helix. Notably, the DNA structure remains consistently intact throughout the simulation process. This is indicated by the flat trajectory of the time series plot, which shows no significant deviation from the initial structure over the course of the simulation. While the histograms on the right side of the plot are not particularly revealing in this instance, they could be instrumental in cases where simulations exhibit significant structural deviations or disintegration, potentially revealing multi-peaked distributions.



Sorry, I did not fully understand the testing / validation part, I believe that is more appropriate for scripts focused on calculations rather than for plotting scripts. That said, I have added a few try/except statements, if that counts for anything. My usual approach to coding, especially when working in Jupyter Notebooks, involves an iterative process where I frequently test and validate parts of the code in separate cells. This method allows me to continuously refine and verify the functionality of the code as I develop it, though it may not be as formalized as proper testing and validation protocols.

One last thing: I'd like to offer a suggestion for Module 2 of the course, specifically regarding the Bash and Python segments. In my role as a tutor for a Python course at BOKU, we've integrated JupyterHub into our teaching. Hosted on a Linux server at our institute, JupyterHub provides a standard terminal interface and facilitates the easy creation of Jupyter Notebooks. This setup significantly reduces both time and technical challenges, allowing students and instructors to concentrate more on the subject matter itself. While I acknowledge the value in learning to install and configure software on personal computers, it seemed that a handful of students did not have a personal computer or were reluctant on using platforms offered by big tech companies (e.g. Google Colab). Implementing a solution like JupyterHub could address these concerns and enhance the learning experience.