Luis Vollmers | **Dr. Maria Reif** | **Prof. Martin Zacharias**

Molecular Dynamics (SS2025)
Exercise 8

Self-Supervised Protein Simulation with Convergence and Principal Component Analysis

Report Tasks

1. Write a Methods section describing your MD simulation setup.
2. Monitor and report stability measures for one step of the simulation setup. Potential energy (EM), temperature (NVT), density and pressure (NPT).
3. Conduct a BSE analysis for R_{gyr} and RMSD(t) of your production run. Use the crystal structure as a reference and discuss the results concerning sampling quality.
4. Show the two extreme projections of the eigenvectors. Also, include a plot of the atomic contributions of eigenvectors 1-10 with a brief discussion.
5. Plot PC1 vs PC2, PC1 vs PC10 and PC9 vs PC10 and discuss the plots briefly.

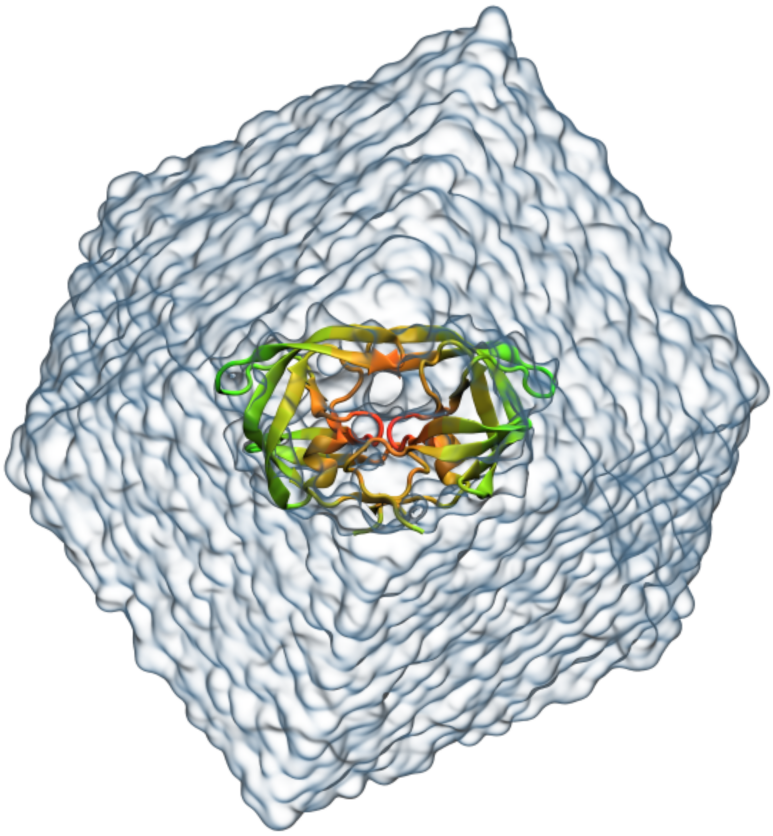

Figure 1: HIV-protease in a water box.

Introduction

This tutorial will conduct an equilibrium molecular dynamics (MD) simulation for the HIV-protease protein (PDB: 5YOK). Uniquely, the simulation setup will be self-supervised, utilising methods from prior exercises as a template library. This exercise focuses on the convergence of the simulation with respect to the protein conformation, for which a blocked standard error (BSE) analysis will be performed.[1] Understanding the movements of HIV-protease is crucial due to their impact on drug efficacy. Principal component analysis (PCA) will be used to elucidate the global molecular movements of this drug-resistant variant.[2]

MD Simulation

This time, the setup of the MD simulations is self-designed. Temperature, pressure, concentration, and other conditions should be reasonably close to physiological or laboratory standards, as in preceding exercises. If parameters are borrowed from previous `MDP` files, they should be validated against an additional source, such as the `GROMACS` manual or another reliable `MDP` file. The protein should not interact with its nearest mirror image; however, enormous box dimensions will result in longer simulations due to excess water molecules. Thus, careful planning of the simulation design is essential. The selection of the water model and force field is also required. While empirical confidence in the successful interplay of the force field and water model is ideal, this tutorial allows for any combinations, with the guideline that newer models are generally better.
The report should include a section describing the critical choices and parameters for the MD simulation, similar to the methodology in scientific publications. There are no strict conventions, but a good approach is to describe the design process similarly to setting up the system: initial coordinates, force field, solvent and box, equilibration, and production. Each stage should detail the thermostat, barostat, electrostatics treatment, and simulation time for equilibration and production. Any constraints applied to bonds should be mentioned, as well as any additional information that seems relevant to the user. For guidance, refer to the methods section of any recent publication utilising MD simulations.
Ensuring the stability of the minimisation and equilibrations is crucial and should be included in the report. Energy minimisation optimises the potential energy, and plotting it against the minimisation steps should result in a curve steeply converging to a minimal potential energy value. The canonical ensemble (NVT) keeps the temperature constant with certain fluctuations, so the temperature should be measured during this equilibration step. The final part of the equilibration uses the isothermal-isobaric ensemble (NPT), which keeps the pressure constant. Due to significant pressure fluctuations, the time evolution of density is of interest. Including one of these plots in the report can indicate simulation stability. Additionally, note that the `PDB` file contains non-standard compounds not parameterised in the current force field; these must be deleted before building the simulation system (`pymol` is recommended to conduct the deletion).

Accessing Sampling Quality and Convergence

Convergence can be assumed when additional sampling does not significantly change the obtained result. This definition aligns with the concept of relative convergence, which describes

[1] `https://doi.org/10.1016/S1574-1400(09)00502-7`
[2] `https://doi.org/10.1016/j.str.2007.12.011`

statistical uncertainty concerning an observable. While absolute convergence cannot be ensured or estimated in an MD simulation, relative convergence has been extensively studied. One method to assess the quality of relative sampling is to conduct a blocked standard error analysis for an observable.[3]

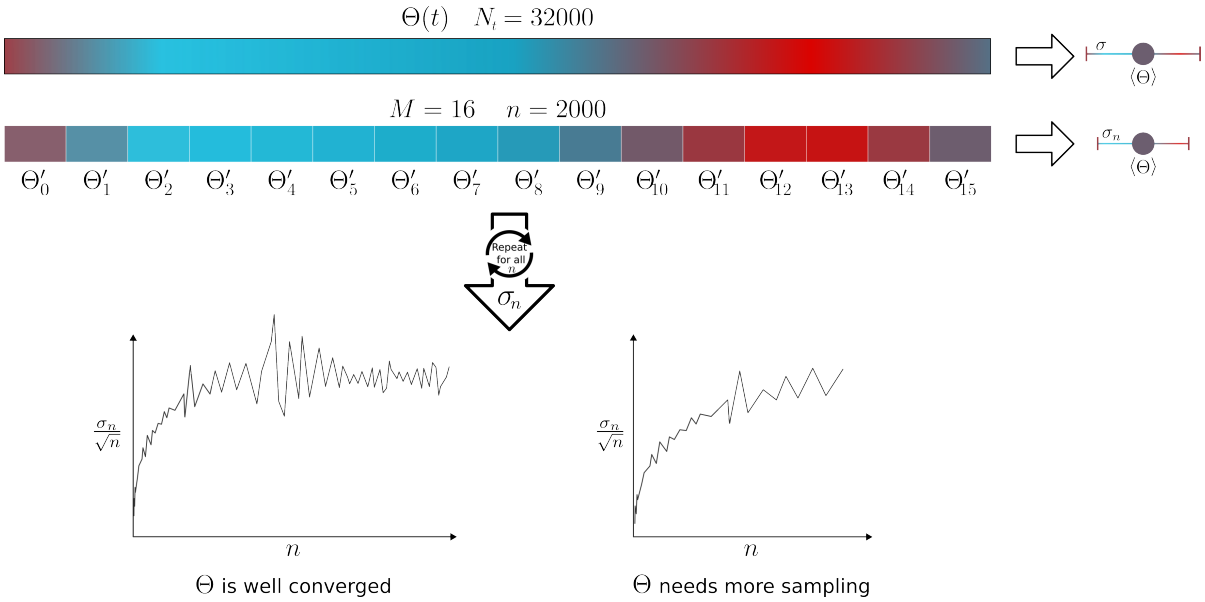

Figure 2: An observable Θ yields an average $\langle\Theta\rangle$ and standard deviation σ when evaluated over an entire trajectory. The colour indicates some arbitrary magnitude of Θ measurements. The total sample size is 32,000, meaning 32,000 instantaneous values of Θ have been collected. We obtain the same overall average by splitting the trajectory into M blocks, each containing n data points, and then calculating each block's average. However, the standard deviation of the block averages, σ_n, differs from the standard deviation of the entire dataset, σ. Repeating this process for all possible (M, n) pairs produces a curve that indicates convergence concerning Θ.

Given an observable Θ, statistical relevance is only achieved once enough independent samples have been measured. Measuring 30 consecutive frames of a slow movement will result in a sharply defined average value; however, this value is not significant because the measurements are likely to be highly correlated. Correlated data is invalid for applying the central limit theorem, which states that a normal distribution emerges for infinitely many independent random variables. The blocked standard error (BSE) analysis addresses this issue by transforming the initial, correlated observations of Θ. This transformation is done by splitting the trajectory into M blocks, each of length n, and calculating the average of each block, Θ'_i. From the M obtained values Θ'_i, the n-specific standard deviation is calculated.[4]

$$\sigma_n = \sqrt{\frac{1}{M-1}\sum_{i=0}^{M}\left(\Theta'_i - \langle\Theta\rangle\right)^2} \tag{1}$$

At some point, by increasing n, the correlation time for Θ is exceeded, rendering Θ'_i effectively uncorrelated. Given a large sampling size, this results in a normal distribution $\mathcal{N}$ with a fixed

[3] https://doi.org/10.1016/S1574-1400(09)00502-7
[4] https://youtu.be/0KqCK0yG9T0

mean and variance.[5]

$$\lim_{M\to\infty} \Theta_i' \sim \mathcal{N}\left(\mu, \frac{\sigma_n}{\sqrt{n}}\right) \tag{2}$$

Since $\sigma_n/\sqrt{n}$ resembles σ of the stable normal distribution, plotting it against the "time period" (i.e., block size n) used for block averaging results in a horizontal line, but only if the samples are truly uncorrelated. For smaller values of n where Θ_i' is still correlated, the corresponding σ will steeply approach the stable standard error (see figure 2).
Creating this plot with GROMACS is relatively easy. When using `gmx analyse`, the flag for error estimation, `-ee`, needs to be specified. The command takes an XVG file as input, such as a file created by `gmx rms`. The commands to generate the BSE analysis for the RMSD and R_{gyr} could look like the following:

```
gmx trjconv -s prod.tpr -f prod.xtc -o prod_center.xtc -pbc mol -center
gmx gyrate -f prod_center.xtc -s prod.tpr -o rgyr.xvg
gmx rms -s em.tpr -f prod_center.xtc -o rmsd.xvg
gmx analyse -f rmsd.xvg -ee bse_rmsd.xvg
gmx analyse -f rgyr.xvg -ee bse_rgyr.xvg
```

The two plots should be part of the report. In order to quantify statistical significance from the BSE plot, some intuition is necessary since the convergence onto a horizontal line is only achieved in an ideal setup. Therefore, the plot needs to be discussed to determine whether the amount of sampling is enough.

Principal Component Analysis of Atomic Movements

Principal Component Analysis (PCA) is a two-step process that begins by calculating the covariance matrix of the atomic movements in the trajectory. An eigenvector decomposition is then performed on this matrix, resulting in $3N$ eigenvectors (principal components) and associated eigenvalues. The eigenvectors define the principal motions of the protein. These eigenvectors are typically sorted from highest to lowest eigenvalue, representing the most significant and most negligible protein motions, respectively. The higher principal components exhibit a higher degree of collectivity, meaning that a small number of high-level principal components account for the essential protein movement, which can provide insights into crucial functionalities.[6]
Mathematically, principal components are a set of new variables that are linear combinations of the initial input variables. In this context, the input variables are the atom coordinates. How are the coefficients of the linear combination chosen? They are chosen so that the resulting principal components are uncorrelated, meaning they carry independent information. The eigenvectors denote the axes along which there is the most variance (i.e., "information") in the data, and the eigenvalues denote the amount of variance in each principal component.
GROMACS provides powerful tools to analyse a trajectory concerning its principal components, which can aid in understanding the functioning of HIV-protease. The following lines of code can be used to calculate the eigenvectors from the trajectories:

```
gmx covar -f prod_center.xtc -s prod.tpr -xpma
```

This command produces the covariance matrix, eigenvectors, and eigenvalues by default. Thus, information about the principal motions is already present. However, to visualise different properties concerning atomic motion and convergence, `gmx anaeig` must be used. The atom-wise

[5] https://aip.scitation.org/doi/pdf/10.1063/1.457480
[6] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4676806/

component for each principal component is specified via the `-comp` flag, and the eigenvector projections on the trajectory are obtained with the `-proj` flag.

```
gmx_mpi anaeig -v eigenvec.trr -f prod_center.xtc -s prod.tpr -proj eigproj1_10.xvg -comp
    eigcomp1_10.xvg -last 10
```

The atomic components are difficult to rationalise, as each atom contributes only a small amount to each principal component. Nevertheless, important information can be deduced, especially for the collective high-level principal components. The projection on the trajectory can provide clues concerning the convergence of the trajectory, albeit the information is qualitative. Given the first two principal components, which are highly collective but decorrelated motions, plotting their trajectory projections against each other can help identify different sampling spaces or clusters. In a well-sampled simulation, each of these clusters should show a variety of transitions. Conversely, in an insufficiently sampled simulation, each cluster is sparsely populated with only a few transitions.
Since most of the atomic motion is captured in the higher principal components, it should be sufficient to calculate the projections and atomic components for the first ten eigenvectors. The plots of the atomic components of all ten eigenvectors should be included in the report, and a brief discussion should be given. This plot is generated directly from the `gmx anaeig` output as an `XVG` file. The trajectory projections, however, require manual post-processing, as the corresponding `XVG` files list principal components against time information and not against each other. The report should include a plot of PC1 against PC2 to assess the sampling quality. Additionally, plots of PC1 against PC10 and PC9 against PC10 are necessary to visualise the decreasing importance of lower eigenvectors. Discuss these plots concerning convergence, and rationalise the usage of PC1 vs. PC2 for quality assessment over any other PC pair. Exemplary plots can be seen in figure 3.
Another useful flag of `gmx anaeig` is `-extr`, which yields the two extreme states of the largest eigenvector. Visualising them conveys an idea of the largest collective motion of the system, and the `PDB` structures that correspond to the end-states can be generated with the following command.

```
gmx anaeig -v eigenvec.trr -f prod_center.xtc -s prod.tpr -extr
```

It is recommended to visualise the end-states in `VMD` instead of `pymol` and to include snapshots in the report. Choosing a representation and a view that supplements a brief discussion is beneficial. Knowledge about binding sites and other structural features should be incorporated.

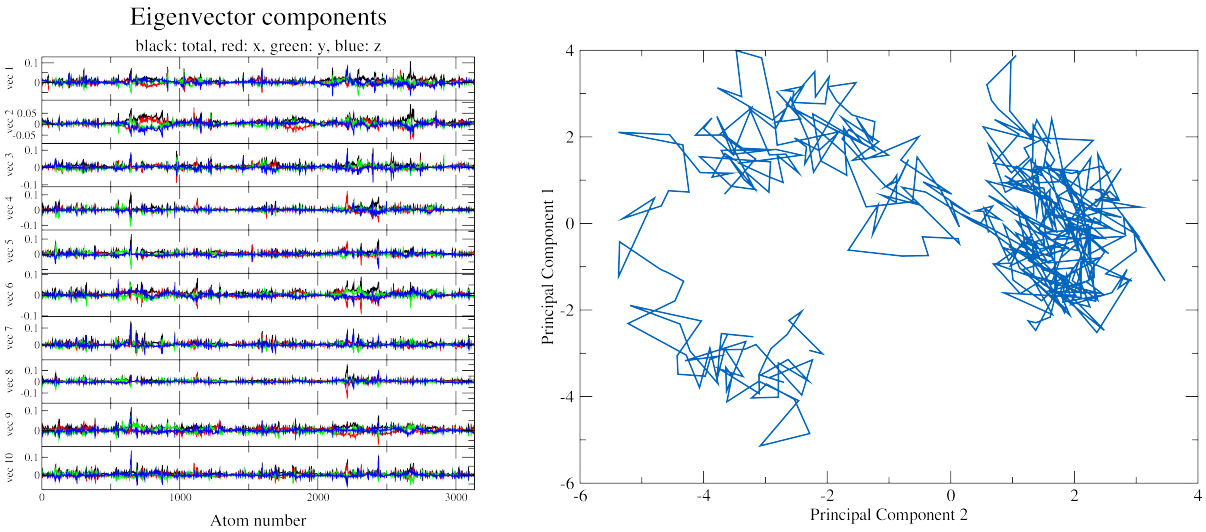

Figure 3: Exemplary results of the atomic components of the first 10 eigenvectors and the trajectory projections.