Analysis of Residual Distance Matrices in Molecular Dynamics Simulations

Using AMBER Force Field

Molecular dynamics (MD) simulations of proteins facilitate the prediction of structural dynamics, potential energy surfaces, and the impact of various perturbations such as mutations, ligand binding, and post-translational modifications. One effective way to analyze the structural impact of such perturbations is by evaluating the distances between the amino acids.

Despite the importance of this analysis, no existing computational tools specifically address this issue for systems simulated using the AMBER force field. To address this, two python codes are developed, that operate in-memory: **"1_distance_analysis_CPU.py"** and **"2_difference_matrix.py"**. These scripts allow for the efficient calculation and comparison of residual distance matrices across different molecular systems.

"1_distance_analysis_CPU.py" script computes a matrix of inter-residue distances for a given molecular system simulated using the AMBER force field. The key computational steps include:

- Extracting atomic coordinates from MD trajectory PDB file.
- Computing the center of mass for each amino acid residue.
- Calculating pairwise distances between residue centers of mass.
- Storing the computed distances in a symmetric distance matrix.
- Averaging the distance matrices over the simulation trajectory to obtain a representative structural snapshot.

The script uses **NumPy** for matrix operations, **SciPy** for distance calculations, and **Matplotlib** for visualization of the computed matrices.

To quantify structural differences between two systems (e.g., mutant vs. wild-type, ligand-bound vs. apo form), the **"2_difference_matrix.py"** script is employed. The key steps include:

- Loading the averaged distance matrices of two systems (e.g., System A and System B).
- Computing the element-wise difference between the matrices (e.g., B - A).
- Calculating the standard deviation for statistical reliability.
- Visualizing the difference matrix to highlight structural changes induced by external factors.

Currently, the implemented scripts are designed for CPU-based computation. Future work will focus on optimizing the codes for GPU-based execution.