Computational Structural and Dynamic Analysis of a Decapeptide via Molecular Dynamics Simulation

Nicolás Rojo Esteban

UAM

Universidad Autónoma
de Madrid

1 Introduction

Peptides play essential roles in biology, acting as hormones, signaling molecules, and structural components in many physiological processes. Understanding peptide behavior at the molecular level is important for gaining insights into protein folding, molecular recognition, and drug design. In structural biology, short peptides often serve as model systems to study the fundamental principles of protein structure and dynamics due to their smaller size and tractability.

Advances in computational chemistry now allow detailed simulations of peptide structure and dynamics. In particular, molecular dynamics (MD) simulations provide a powerful tool to investigate biomolecular behavior with atomic resolution. MD simulations can capture conformational changes over time, offering complementary information to experimental methods like NMR or X-ray crystallography. Software suites such as *Amber* are widely used for these purposes, enabling the setup, execution, and analysis of protein and peptide simulations. In this study, we employ all-atom MD simulations to analyze the structure and dynamics of a decapeptide with the sequence

```
NGLY TYR ASP PHE GLU THR GLY THR TRP CGLY
```

We focus on assessing its structural stability and exploring its conformational ensemble using computational tools in the *Amber* package.

2 Objectives

The primary objective of this work is to perform a simulation-based structural and dynamic analysis of the decapeptide. The specific goals are:

- To simulate the peptide in an explicit solvent environment and observe its behavior over time.
- To evaluate structural stability through the analysis of backbone RMSD.
- To identify predominant conformations via clustering analysis of the MD trajectory.
- To characterize the solvation of the peptide using radial distribution function (RDF) analysis.

3 Methodology

All simulations were performed with *Amber*, beginning with system preparation in `tleap`, where the peptide was built using the ff14SB force field and solvated with TIP3P water. Next, energy minimization in `pmemd` removed unfavorable contacts, followed by stepwise heating (NVT) and density equilibration (NPT). The production run was then conducted at 300 K and 1 atm, saving coordinates periodically for analysis. Post-simulation, `cpptraj` was used to compute RMSD, perform clustering, and evaluate solute–solvent interactions, yielding a comprehensive view of the peptide's structural and dynamic behavior.

4 Results

4.1 LEaP

The LEaP input file is designed to prepare the peptide system for molecular dynamics simulations with the Amber software suite. Initially, it loads the necessary force field parameters by sourcing libraries such as `leaprc.protein.ff14SB` for proteins and `leaprc.water.tip3p` for the TIP3P water model. This ensures that accurate parameters are available for both the peptide and its solvent. The script then constructs the peptide using its amino acid sequence ({ NGLY TYR ASP PHE GLU THR GLY THR TRP CGLY }), thereby generating the three-dimensional structure with appropriate backbone and side-chain conformations. Next, the peptide is solvated in a cubic water box with a defined padding (e.g., 15.0 Å and a buffer of 1.8 Å), ensuring that it is completely surrounded by water molecules, which mimics physiological conditions. To neutralize the system's overall charge, counter-ions (such as Na^+) are added, which stabilizes the electrostatic environment during simulations. Finally, the input file performs a system check and outputs the necessary files, including the Amber topology (`.top`) and coordinate (`.crd`) files, as well as a PDB file for visualization, before terminating the session with the `quit` command.

After preparing the peptide system, two PDB files were obtained: `1uao.pdb` and `chignolin_leap.pdb`. These files were visualized using PyMOL to inspect the final structures and ensure correctness before proceeding with further simulations or analyses.

Figure 1 illustrates the peptide (green ribbon) within a cubic box of explicit water molecules (red sticks) along with two counter-ions (purple spheres), as generated by `tleap`. This representation confirms that the peptide is fully solvated, which is crucial for realistic molecular dynamics simulations.

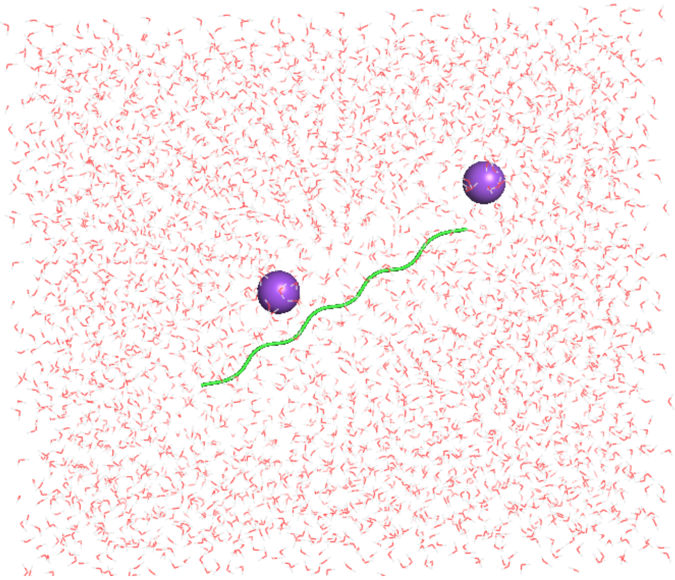

Figure 1: Visualization of `chignolin_leap.pdb` in PyMOL showing the peptide in explicit solvent (TIP3P) with added ions (purple spheres).

In contrast, Figure 2 shows the peptide without explicit water molecules or ions. This view, derived from the same coordinate data but stripped of solvent, is often used to highlight the peptide's overall fold and secondary-structure elements more clearly.

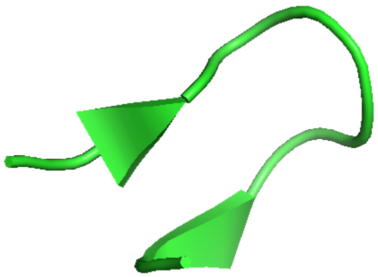

Figure 2: Visualization of the peptide without explicit water, allowing a clearer view of its backbone conformation.

Overall, examining both the solvated and non-solvated views in PyMOL helps verify that the system was built correctly and that the peptide is fully immersed in water, with ions placed to neutralize its net charge.

4.2 Min

Once the system was prepared and saved as `chignolin.top` and `chignolin.crd`, an energy minimization step was performed to relieve any unfavorable contacts and stabilize the structure prior to molecular dynamics. This step utilized the `pmemd20` engine of the AMBER suite.

The energy minimization phase is executed in two successive steps using the input files `chignolin_1_min.in` and `chignolin_2_min.in`. The first input file, `chignolin_1_min.in`, initiates the minimization process starting from the coordinates generated by LEaP. In this stage, the system is relaxed by employing a steepest descent method for the initial minimization cycles, followed by conjugate gradient optimization to efficiently reduce high-energy contacts and eliminate steric clashes, particularly within the solvent and near the peptide. Typical control parameters include setting `imin=1` to specify minimization, a total of `maxcyc=1000` cycles with a switch to conjugate gradient after `ncyc=500` cycles, and a non-bonded cutoff (e.g., `cut=10.0` Å) to balance computational efficiency and accuracy.

The second input file, `chignolin_2_min.in`, is used to continue the minimization in a restart mode (for instance, with `irest=1` and `ntx=5`), allowing the system to further refine its conformation by removing any remaining unfavorable interactions. This subsequent minimization step ensures that both the peptide and the surrounding solvent achieve a well-relaxed, low-energy state before proceeding to the heating and equilibration stages of the molecular dynamics simulation.

4.3 Dyn

For the dynamics phase, multiple input files were employed to guide the simulation through its various stages, including initial heating, density equilibration, and production dynamics. The first dynamics input `chignolin_01_dyn.in` is used for the heating phase, where the system is gradually brought from a low temperature to the target temperature (typically 300 K) under canonical ensemble conditions, using a time step of 2 fs and a Langevin thermostat (with parameters such as `ntt=3`, `gamma_ln=3.0`, and `ig=-1`). This phase allows the system to gently acquire kinetic energy while maintaining positional restraints to avoid large structural deviations. Then, the input file `chignolin_1_dyn.in` is utilized during the density equilibration phase under the isothermal–isobaric ensemble. Here, the simulation continues with the same time step but now allows the volume to fluctuate (using pressure coupling, `ntp=1`) so that the system can adjust to the target pressure and achieve a realistic solvent density. Finally, `chignolin_02_dyn.in` is used for the dynamics run, where the fully equilibrated system is simulated for an extended period of 1 ns to capture the peptide's conformational behavior. The entire MD simulation is executed using the `pmemd20` engine using the run script `run_pmemd20`, which is the responsible of the sequential use of these input files, reading the corresponding topology and coordinate files, and writing out trajectory and restart files.

4.4 Analysis

After completing the molecular dynamics simulation, the `cpptraj` module was used to analize the results. `cpptraj` is the primary trajectory analysis program included in the AmberTools suite, designed to handle and process molecular dynamics trajectories produced by simulation engines such as `pmemd` or `sander`. It is capable of reading various trajectory file formats along with the corresponding topology file and offers a wide range of commands

4.4.1 GetPDB

The final PDB file was generated using two `cpptraj` input scripts named `en_cpptraj_GetPDB_iqtc07` and `en_cpptraj_GetPDB_nc_iqtc07`. These scripts read the last frames from the molecular dynamics trajectory, apply coordinate centering through the `autoimage` command, and strip solvent or ions if necessary. The `trajout` directive then writes the processed coordinates to a PDB file, which can be visualized in PyMOL.

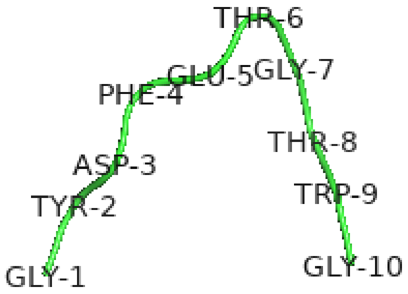

Figure 3: Visualization in PyMOL of the peptide's final conformation after the MD run, obtained via a `cpptraj` script that generates a PDB from the simulation trajectory. Residues are labeled to highlight the final arrangement of the peptide backbone.

The figure 3 displays the decapeptide after the molecular dynamics simulation,. The peptide's backbone adopts a bent conformation, with the central residues (including Phe-4, Glu-5, and Thr-6) forming the turn region. The final arrangement suggests that side-chain interactions, as well as backbone hydrogen bonds, may have contributed to maintaining this compact form.

4.4.2 RMSD

The RMSD was done using the `en_cpptraj_fullrmsd_iqtc07` .The `cpptraj` input file begins by reading the simulation trajectory using the `trajin` command. It then applies `autoimage` to center the peptide in the periodic box, removing discontinuities caused by periodic boundaries. Next, `strip :WAT,Na+` removes water and ions so that only the peptide coordinates are analyzed. The line `rmsd first out chignolin_rmsd.dat :1-10@CA` calculates the root-mean-square deviation (RMSD) of the Cα atoms (residues 1–10) relative to the first frame, storing the values in `chignolin_rmsd.dat`. Finally, `trajout` writes a new, solvent-free trajectory, and `go` executes all the specified commands.

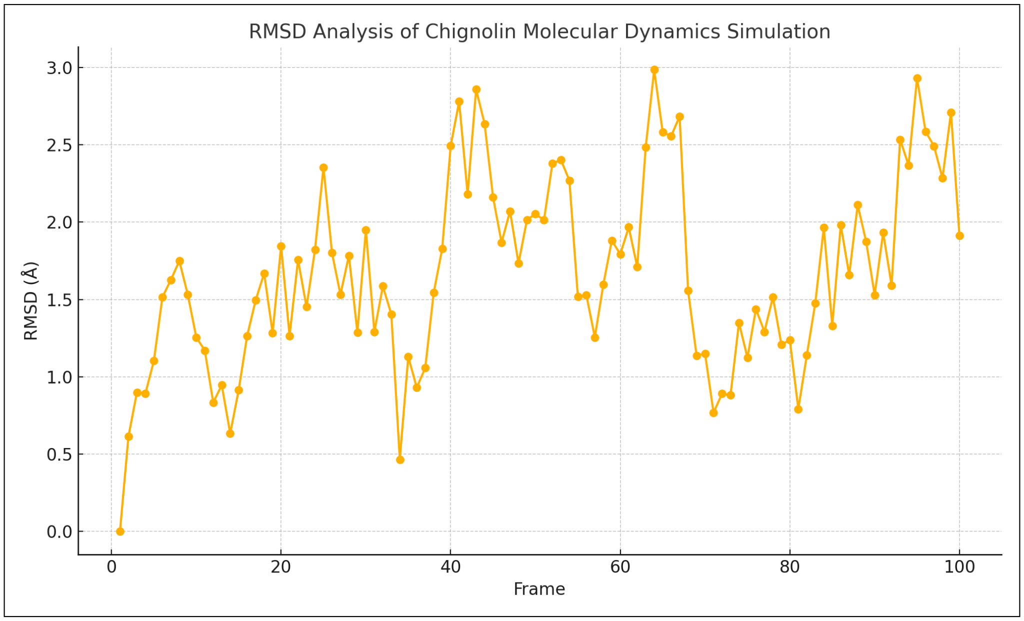

Figure 4: Root Mean Square Deviation (RMSD) of Chignolin over the Molecular Dynamics Simulation.

The plot 4 shows the RMSD values of the backbone atoms (Cα) of Chignolin throughout the molecular dynamics simulation. RMSD is a measure of the structural deviation of the protein from its reference structure over time, typically reflecting the stability and conformational changes in the system. At the beginning of the simulation, RMSD increases as the system relaxes from its initial configuration. Over time, the RMSD oscillates between 1.0 Å and 3.0 Å, suggesting that Chignolin undergoes structural fluctuations, which is expected due to its small size and inherent flexibility. The absence of a continuous increase in RMSD indicates that the system remains stable without undergoing denaturation or major unfolding events.

Clustering Analysis

The cpptraj input file for clustering analysis first loads the trajectory with the `trajin` command and centers the system using `autoimage`. It then applies a clustering algorithm that groups peptide conformations based on the RMSD of backbone atoms using hierarchical agglomerative clustering with average linkage. The input specifies the clustering parameters and outputs a summary of cluster statistics along with representative structures in PDB format. This analysis identifies the major conformational states sampled during the simulation.

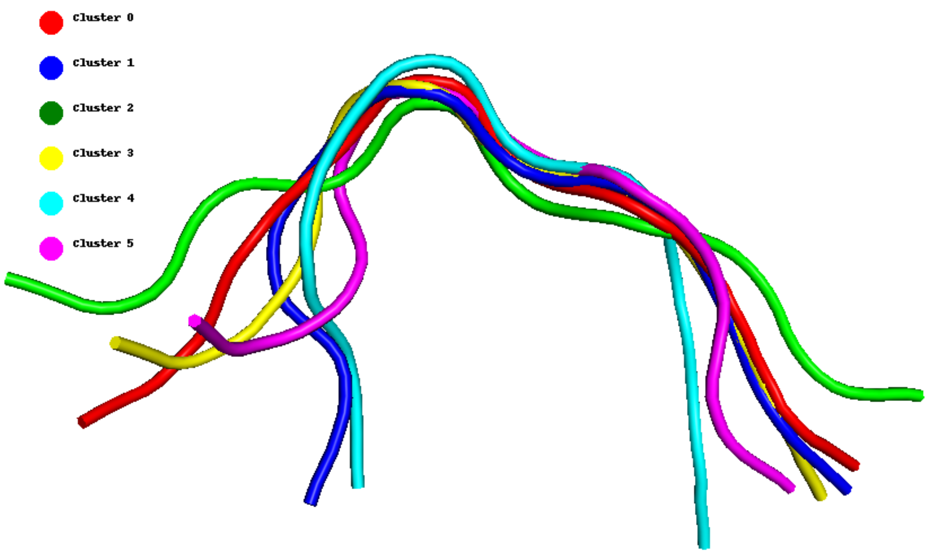

Figure 5: Superimposed representative structures of the six identified clusters, each colored according to its cluster ID. The legend on the left labels the clusters from 0 to 5

Figure 5 shows the superposition of the representative structures for each of the six clusters identified by the cpptraj analysis. Each cluster is colored differently, with the legend on the left indicating which color corresponds to which cluster. The structural alignment highlights the conformational diversity sampled by the peptide over the course of the simulation. The most populated cluster (Cluster 0, shown in red) appears to adopt a relatively stable, well-defined backbone conformation, while the other clusters (blue, green, yellow, magenta, and cyan) exhibit varying degrees of deviation from that dominant structure. These variations often reflect subtle differences in backbone torsion angles or side-chain orientations, underscoring the peptide's capacity to sample multiple conformational states in solution.

Table 1: Clustering results from the cpptraj analysis.

Cluster	Frames	Frac	AvgDist	Stdev	Centroid	AvgCDist
0	45	0.450	1.128	0.379	11	2.274
1	22	0.220	1.125	0.358	85	2.569
2	11	0.110	1.135	0.285	43	2.943
3	10	0.100	1.102	0.308	56	2.256
4	8	0.080	0.995	0.334	96	3.354
5	4	0.040	0.824	0.258	50	2.402

The table summarizes the clustering results obtained from the cpptraj analysis. Each row corresponds to a different cluster of similar conformations identified during the simulation. The ”Cluster” column labels the clusters, while ”Frames” indicates the number of trajectory frames assigned to each cluster and ”Frac” shows the fraction of the total frames that each cluster represents. ”AvgDist” represents the average distance among the structures within a cluster, and ”Stdev” is the standard deviation of these distances, which reflects the internal homogeneity of the cluster. The ”Centroid” column provides the index of the representative structure (centroid) for each cluster, and ”AvgCDist” gives the average distance from this centroid to the other structures in the cluster.

According to the results, Cluster 0 is the most populated, containing 45 frames, with an average internal distance of 1.128 Å and a standard deviation of 0.379 Å, suggesting that the conformations in this cluster are quite similar. Clusters 1 and 2 follow with 22 frames and 11 frames, respectively, indicating additional, less frequent conformational states with similar average distances but slight differences in structural variability. The remaining clusters (3, 4, and 5) contain fewer frames, with Cluster 5 being the least populated with 4 frames, 4, which suggests that these represent rare or transient conformations.

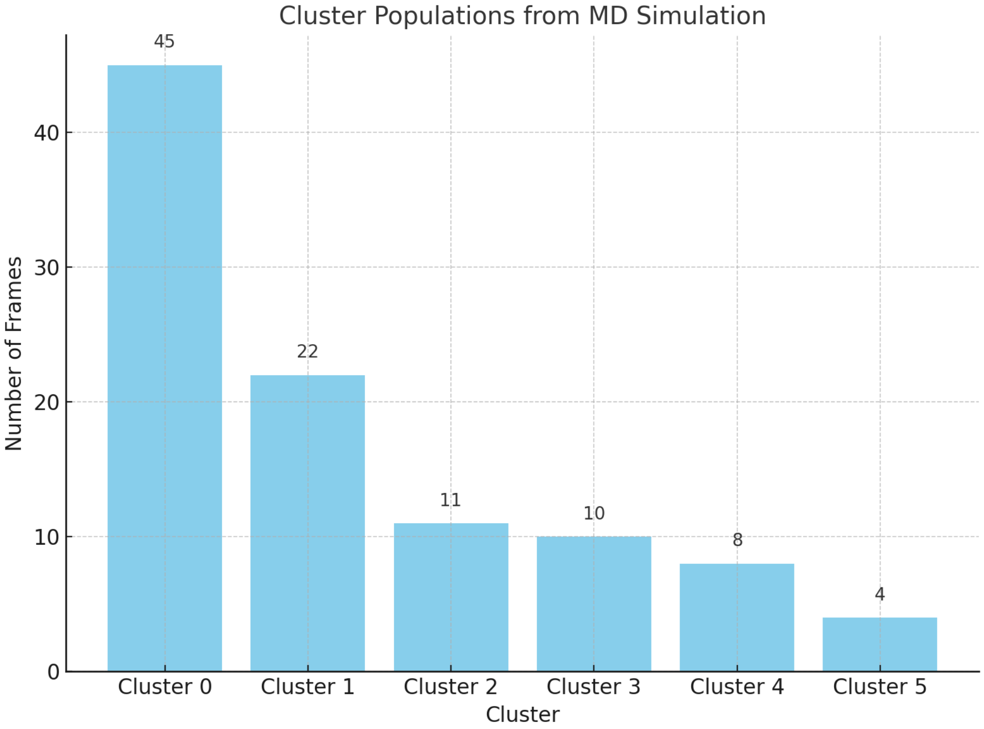

Figure 6: Cluster populations from the MD simulation.

Figure 6 illustrates the population of each cluster in terms of the number of trajectory frames. Cluster 0 has the largest population with 45 frames, indicating that it represents the most frequently sampled conformation during the simulation. The remaining clusters each have fewer frames, suggesting that they correspond to alternative, less frequently visited states of the peptide.

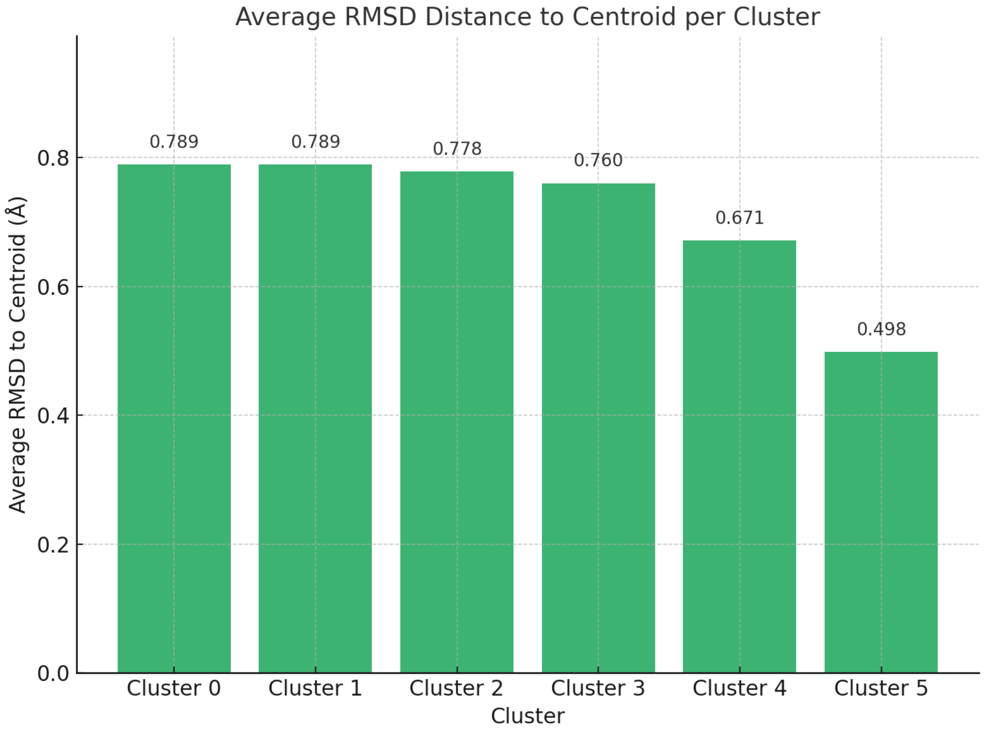

Figure 7: Average RMSD distance to centroid per cluster.

Figure 7 shows the average RMSD distance of each cluster to its respective centroid. Lower values indicate that a cluster's conformations are tightly grouped around its representative structure, whereas higher values suggest greater structural diversity within that cluster. Notably, Cluster 5 exhibits the smallest average RMSD, implying a closely packed set of conformations, even though its population is relatively small.

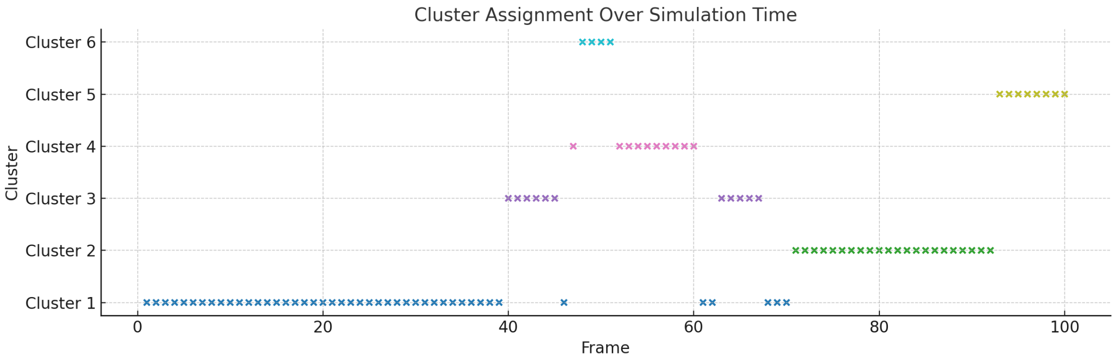

Figure 8: Temporal evolution of cluster assignments during the molecular dynamics simulation of Chignolin. Each point represents the cluster assignment of the system at a specific frame. This timeline illustrates the conformational transitions and stability of distinct clusters over time.

Figure 8 shows the temporal evolution of the cluster assignments during the molecular dynamics (MD) simulation of Chignolin.

The timeline demonstrates how the system transitions between different conformational states over the course of the simulation. Notably, the system predominantly occupies **Cluster 1** during the initial phase of the simulation, suggesting that this conformational state is highly populated and potentially represents a stable folded structure.

As the simulation progresses, transitions to other clusters (e.g., **Clusters 2, 3, and 4**) are observed, indicating the exploration of alternative conformational states. These transitions reflect the intrinsic flexibility of Chignolin and its ability to sample various structural basins in the conformational landscape.

The presence of periods where the system remains within a single cluster for an extended number of frames suggests **metastable states**, where the molecule is kinetically trapped in a specific conformation. In contrast, rapid transitions between clusters highlight regions of **higher conformational mobility**.

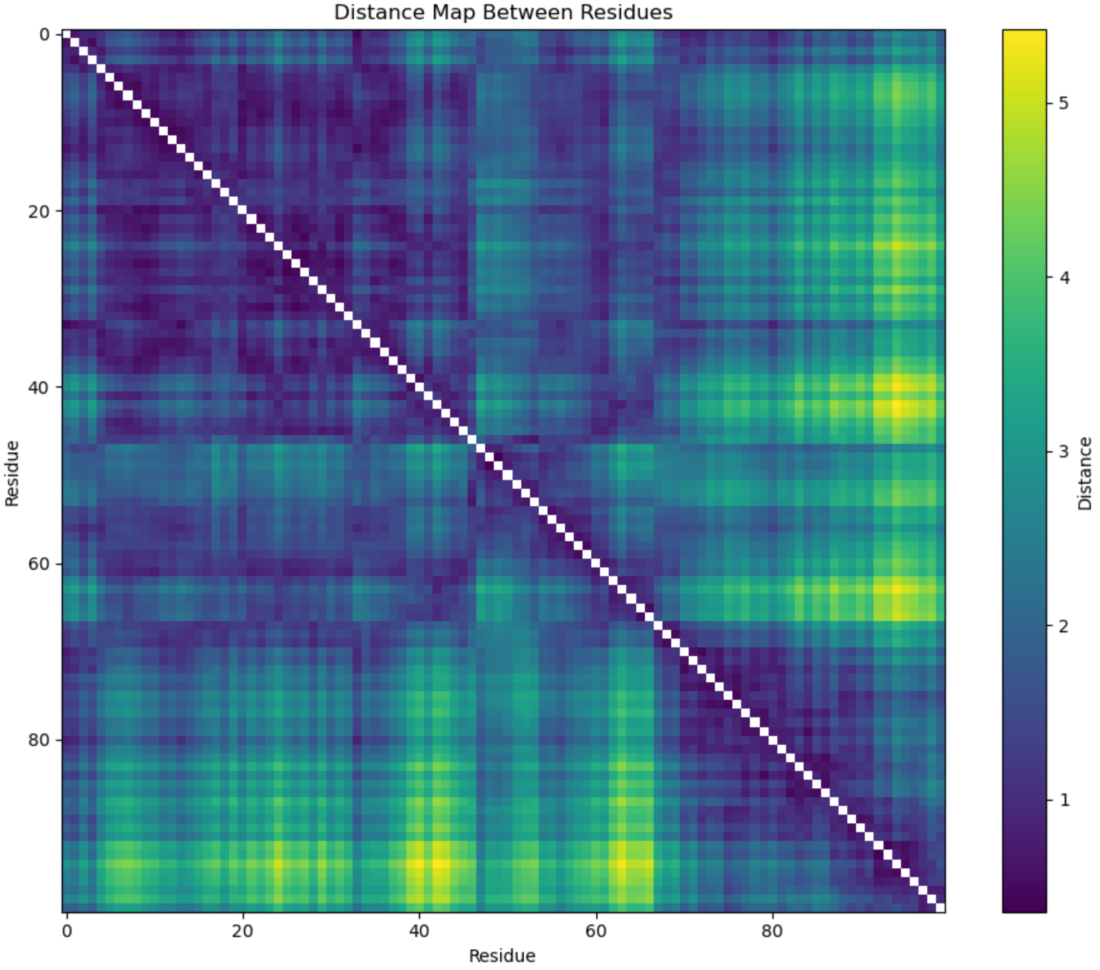

Figure 9: Pairwise RMSD distance matrix calculated from the molecular dynamics simulation of Chignolin. Each element represents the RMSD (in Å) between two frames of the trajectory. Darker regions indicate higher structural similarity (lower RMSD), while lighter regions indicate greater structural differences. The presence of distinct blocks along the diagonal suggests the existence of stable conformational clusters during the simulation.

Each cell (i, j) in the 100×100 matrix represents the distance between residue i and residue j. Because the distance from i to j is the same as from j to i, the matrix is symmetric.

Darker (purple/blue) areas correspond to shorter distances, whereas lighter (yellow/green) regions indicate larger distances. This coloring scheme allows for a quick visual identification of residue clusters that are close in space versus those that are farther apart.

The main diagonal shows the distance of each residue to itself. Often, these values are zero or undefined (NaN), so the diagonal can appear as a distinct line. Since self-distances are not typically informative for structural analysis, they can be ignored in many cases.

Blocks or bands of similar color may reveal domains or secondary structures that are close together. Likewise, residues with consistently larger distance values are likely found in different regions of the protein fold. This distance map can help in understanding protein folding patterns, inter-residue interactions, or identifying functionally important clusters.

4.4.3 RDF

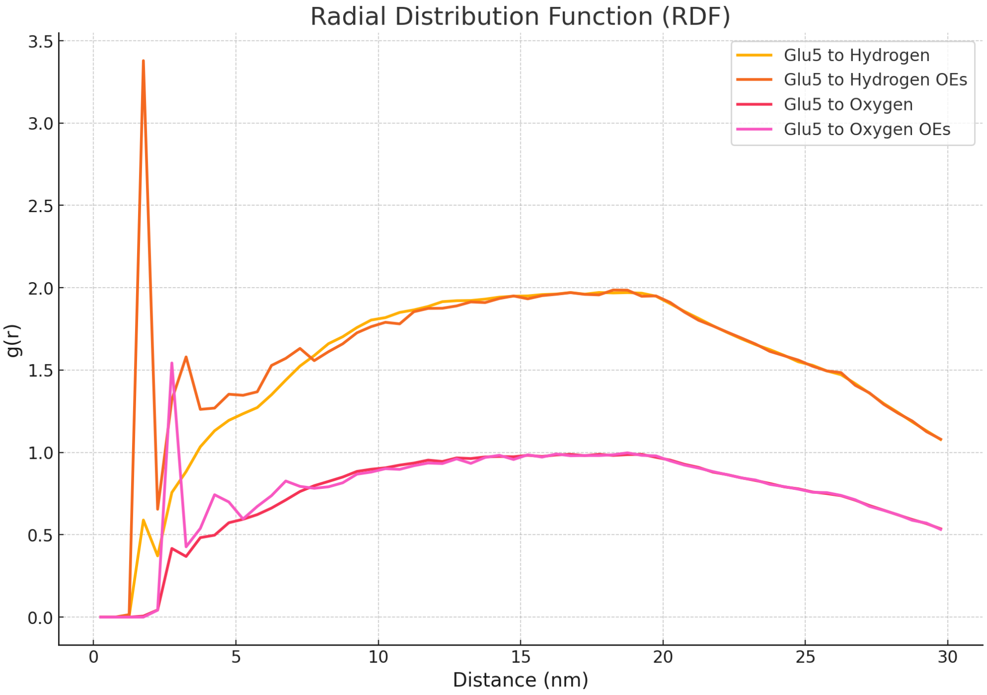

Figure 10: Radial Distribution Function (RDF) of Glu5 with respect to different atomic groups: Hydrogen (blue), Hydrogen OEs (orange), Oxygen (green), and Oxygen OEs (red).

Figure 10 shows the Radial Distribution Functions (RDFs) calculated between the residue **Glu5** and different atomic groups in the system under study. The RDF, denoted as $g(r)$, describes how atomic density varies as a function of distance from a reference particle. In this case, **Glu5** serves as the reference residue, and we investigate its spatial correlations with hydrogen and oxygen atoms, as well as specific subgroups labeled **OEs** (oxygen epsilon atoms typically found in carboxylate groups).

- **Glu5 to Hydrogen (blue line):** This RDF represents the probability distribution of finding hydrogen atoms at a given distance from Glu5. A prominent peak indicates regions of preferred hydrogen atom proximity, typically associated with hydrogen bonding or local structural arrangements.

- **Glu5 to Hydrogen OEs (orange line):** Similar to the previous curve, but focused on hydrogen atoms associated specifically with oxygen epsilon atoms. The presence of distinct peaks suggests specific interactions, possibly hydrogen bonds with the carboxylate oxygens of Glu5.

- **Glu5 to Oxygen (green line):** This curve illustrates the distribution of oxygen atoms around Glu5. Peaks at shorter distances may imply electrostatic interactions or direct hydrogen bonding.

- **Glu5 to Oxygen OEs (red line):** RDF for oxygen epsilon atoms in proximity to Glu5. Given their potential for negative charge, peaks at certain distances can reflect favorable electrostatic interactions or structured solvation shells.

The analysis of RDFs provides insight into the structural organization and interaction preferences of Glu5 within its molecular environment. The locations of the first peaks indicate the most probable distances for close contacts between Glu5 and the different atomic groups. Peaks beyond the first coordination shell may represent more long-range ordering effects or secondary interaction zones.

5 Conclusions

The molecular dynamics simulations carried out with Amber have provided significant insights into the behavior of the decapeptide in an explicit solvent environment. Analysis of the backbone RMSD showed that the peptide quickly reached and maintained a stable conformation over time. Clustering analysis identified a predominant structural state that dominates the trajectory, while also revealing minor conformational variations that underscore the inherent flexibility of the system. Moreover, the radial distribution function analysis confirmed effective solvation, particularly around the polar and charged residues, which is essential for stabilizing the peptide structure. Overall, these results validate the computational protocol used in this study, demonstrating its efficacy in capturing both the structural stability and dynamic behavior of the peptide.

6 Bibliography

- Case, D. A., Cheatham, T. E., Darden, T., Gohlke, H., Luo, R., Merz, K. M., *et al.* (2005). *Journal of Computational Chemistry, 26*(16), 1668–1688.

- Civera, M., Sattin, S., & Kumar, K. (2023). *Exploration of Drug Science, 1*, 140–171.

- Daura, X., van Gunsteren, W. F., & Mark, A. E. (1999). *Proteins: Structure, Function, and Genetics, 34*(3), 269–280.

- Karplus, M., & McCammon, J. A. (2002). *Nature Structural Biology, 9*(9), 646–652.
- Roe, D. R., & Cheatham, T. E. III. (2013). *Journal of Chemical Theory and Computation, 9*(7), 3084–3095.
- Hospital, A., Goni, J. R., Orozco, M., & Gelpí, J. L. (2015). *Advances and Applications in Bioinformatics and Chemistry, 8*, 37–47.