Contents

High-Throughput Multi-Protein MD Pipeline – User Guide

1. Overview

This workflow provides a fully automated pipeline designed to prepare, equilibrate, simulate, and analyze multiple protein systems using GROMACS. The system is ideal for batch processing, large-scale datasets, or standardized production MD workflows. The pipeline automatically executes all stages of a classical molecular dynamics protocol—energy minimization, NVT equilibration, NPT equilibration, and short production replicas—followed by automated analysis and PDF report generation. The automation ensures reproducibility, eliminates manual setup errors, and reduces user workload to simply providing a folder containing PDB structures and running the main script.

2. Directory Structure and Workflow

The pipeline is organized to ensure clean, predictable, and modular data management. The user provides the **input directory**, and the pipeline creates the **output tree** automatically.

2.1 Required User-Provided Directories

You must create the following structure:

```
your_project/
│
├── proteins/      ← place all input .pdb files here
│       ├── protein1.pdb
│       ├── protein2.pdb
│       └── ...
├── multi_protein_pipeline.py
├── pre_analysis.py
├── replica_analysis.py
├── script_guide.pdf
├── pymol_guide.pdf
└── (output will be created automatically)
```

2.2 Pipeline-Generated Output Structure

Once the script runs, it produces:

```
output/
├── ProteinA/
│      ├── em.tpr, em.gro, em.edr, em.xvg
│      ├── nvt.tpr, nvt.gro, nvt.edr, nvt_temperature.xvg ...
│      ├── npt.tpr, npt.gro, npt.edr, npt_pressure.xvg ...
│      ├── replica1/
│      │      ├── mdrep1.tpr
│      │      ├── mdrep1.xtc
│      │      ├── mdrep1.gro
│      │      └── mdrep1_final.xtc
│      ├── replica2/
│      │      ├── mdrep2.tpr
│      │      ├── mdrep2.xtc
│      │      ├── mdrep2.gro
│      │      └── mdrep2_final.xtc
│      ├── pre_analysis.pdf    ← automatic report
│      ├── replicas_analysis.pdf    ← automatic report
│      └── logs/
│
├── ProteinB/
│      └── same structure ...
│
└── ...
```

3. How to Use the Pipeline

Running the workflow is extremely simple. Only three commands are required:

```
module load gromacs
module load python-data/3.12-25.09 ← must 3.12 or more
python3 multi_protein_pipeline.py proteins
```

Required Tools, Modules, and Python Libraries:

Category	*Tool / Library*	*Required For*	*Notes*
HPC Modules	**GROMACS 2024.x (gromacs / gmx_mpi)**	Running EM, NVT, NPT, replicas	Must be loaded on Puhti (module load gromacs)
	python-data/3.12-25.09	Python environment + scientific libraries	Includes NumPy, Matplotlib; stable on Puhti
Core Python Standard Libraries	**os**	Directory/path handling	Built-in
	subprocess	Executing GROMACS commands	Built-in
	sys	System-level operations	Built-in
	shutil	File handling for workflow	Built-in
Scientific Python Libraries	**NumPy**	Reading XVG, numerical analysis	Included in python-data module
	Matplotlib	Plotting EM/NVT/NPT/replica graphs	Included in python-data module
PDF Generation	**ReportLab**	Writing pre_analysis PDF report	Included in python-data or installed via pip
MD Trajectory Analysis	**MDAnalysis**	Replica structural analysis (RMSD, RMSF, Radius of Gyration)	Must be installed manually (pip or Conda)
Optional Visualization	**PyMOL**	Generating molecular images for documentation	Installed locally; not part of automated pipeline

The pipeline automatically:

1. Prepares each protein system
2. Solvates and ionizes it
3. Runs EM, NVT, NPT
4. Runs two short production replicas
5. Extracts .xvg analysis data
6. Generates a PDF analysis report for each protein

4. Script Descriptions

4.1 multi_protein_pipeline.py (Main Automation Engine)

This is the primary driver script. Its responsibilities include:

- Processing **all proteins** in the proteins/ folder
- Running GROMACS modules for:
    - Structure cleaning and topology generation
    - Solvation
    - Ion addition and neutralization
    - Energy minimization (EM)
    - NVT equilibration
    - NPT equilibration
- Automatically generating .tpr, .gro, .edr, .xvg, .xtc, and log files
- Creating replication subdirectories for two production MD runs
- Extracting energy and temperature data for analysis
- Automatically invoking **pre_analysis.py and replica_analysis.py** at the end of each protein run

4.2 pre_analysis.py (Automated Plotting + PDF Report Generator)

This script performs post-processing and produces a comprehensive PDF report for each protein. Its functions include:

- Reading .xvg files from EM, NVT, NPT, and replicas
- Generating high-resolution plots for:
    - Temperature
    - Pressure
    - Density
    - Potential energy
- Computing averages, fluctuations, and drifts
- Evaluating stability (pass/fail) for each stage
- Writing comments and recommendations for the user
- Exporting all graphs + comments into **pre_analysis.pdf** in each protein's output directory

4.3 replica_analysis.py (Replica Trajectory Analysis + PDF Report Generator)

This script performs detailed post-processing of replica MD simulations (replica1 and replica2) for each protein and generates a consolidated PDF report. Its main functions include:

- Reading replica trajectory files (.xtc) and topologies (.tpr) for each replica.
- Aligning the trajectories to remove translational/rotational motion.
- Computing key structural metrics:
    - **RMSD (Root Mean Square Deviation)** – monitors structural stability over time.
    - **RMSF (Root Mean Square Fluctuation)** – identifies flexible residues and potential loop motions.
    - **Radius of Gyration (Rg)** – evaluates compactness of the protein structure.
- Automatically generating high-resolution plots for each metric per replica.
- Assessing stability based on predefined thresholds and adding **conditional comments** under each plot, indicating whether the replica is stable or suggesting further improvements (e.g., longer equilibration, additional restraints, or adjusted parameters).
- Exporting all plots and comments into a single PDF file (replicas_analysis.pdf) in the corresponding protein's output directory, providing a concise, visual, and interpretable report of replica behavior.

This script complements pre_analysis.py by focusing specifically on **replica trajectories**, enabling users to verify convergence and structural stability before production runs or further analysis.

5. Standard Expected Results

5.1 for Each Stage

The table below summarizes the typical indicators of a successful MD preparation pipeline:

Stage	*Expected Behavior*	*What Stability Looks Like*	*Common Problems*
Energy Minimization (EM)	System seeks lowest-energy structure	Potential energy decreases smoothly and reaches a plateau	EM not converging, large initial steric clashes
NVT Equilibration	Temperature coupling stabilizes kinetic energy	Temperature ~300 K with small fluctuations (±1–2 K)	Temperature drift, poor thermostat coupling
NPT Equilibration	System adjusts volume/pressure	Pressure fluctuates but density stabilizes; temperature remains constant	Box collapse/expansion, unstable pressure
Replica 1 & 2	Short production runs	Stable temperature & potential energy; trajectories wrap and unwrap cleanly	Large drift, unstable potential energy

5.2 for Each Replicas

Here's a standard reference table for RMSD, RMSF, and radius of gyration (Rg)

Metric	*Typical Range*	*Interpretation / Comment*
RMSD (Å)	1–3 Å (after equilibration)	Indicates overall structural stability. Values <3 Å usually mean the protein is stable. Large jumps or drift may suggest unfolding or instability.
RMSF (Å)	0.5–2 Å (core regions)	Measures flexibility per residue. Higher values (2–5 Å) often appear in loops or termini. Peaks indicate flexible regions.
Radius of Gyration (Rg, Å)	±1–2% around initial value	Reflects protein compactness. Large changes (>5%) indicate significant conformational change or unfolding.

6. Customizing the Pipeline for Your Own Needs

This pipeline is designed to be fully editable so that any user can adapt the MD protocol to their scientific requirements. Below is a guide to all customizable components and how to safely modify them.

6.1 Modifying MDP Parameters (EM, NVT, NPT, Replica Runs)

They are generated by the write_mdp_files() function inside **multi_protein_pipeline.py**.
You can freely edit any of the following MDP sections:

Energy Minimization (em.mdp)

- emtol – convergence threshold
- nsteps – number of minimization steps
- integrator – usually *steep* or *cg*
- cutoff-scheme, PME settings

NVT Equilibration (nvt.mdp)

- ref_t = 300 → change temperature
- Temperature groups (tc-grps = Protein Non-Protein)
- Velocity generation settings.
- constraint scheme
- timestep (dt)
- output frequency (nstenergy, nstxout)

NPT Equilibration (npt.mdp)

- Barostat type (Parrinello-Rahman, Berendsen, C-rescale)
- Pressure reference (ref_p)
- Compressibility
- Coupling time constant (tau_p)
- Duration of equilibration (edit nsteps)

Replica Production Runs (mdrep1.mdp & mdrep2.mdp)

- Replica length (nsteps)
- Temperature (ref-t)
- Whether to generate new velocities
- Random seeds (gen-seed)
- Output trajectory frequency (nstxtcout)

6.2 Changing Output Directory Organization

The current structure:

```
output/
    └── ProteinName/
            ├── em.*, nvt.*, npt.*
            ├── replica1/
            └── replica2/
```

You may reorganize by:

- Adding new subfolders
- Moving replicas
- Naming the outputs differently
- Storing plots or PDF reports in additional directories

Changes can be made where the script uses:

```
os.makedirs(outdir)
os.path.join(outdir, ...)
```

6.3 Adding or Removing Simulation Stages

You can disable stages or add more stages by editing the main function:

Examples:

Remove NVT stage

Comment out:

```
# run NVT
```

Add a second NPT or additional temperature steps

You can duplicate the entire NPT section and modify its MDP.

6.4 Modifying Protein Preparation Parameters

Inside the GROMACS commands:

- Change the water model (tip3p, spc, spce)
- Change the force field (charmm27, amber99sb, etc.)
- Modify the solvent box size in editconf
- Change ion concentration (replace -neutral with -conc 0.15)

6.5 Extending the Pipeline to More Advanced Features

The script can be expanded to include:

- Production MD runs (ns-scale)
- Automated RMSD clustering
- Binding pocket tracking
- Adaptive sampling
- Multiple replicas with different seeds
- GPU-based mdrun
- Amber or OPLS force fields

These modifications mainly affect:

- MDP templates
- GROMACS command-line arguments
- Directory naming conventions

6.7 Safety Notes for Customization

When modifying the script:

- Always check that **file paths** are correct
- Keep consistent TPR–GRO file names
- Ensure restraints match the topology
- Always validate MDP syntax (GROMACS rejects invalid lines)

7. Simulation Output Images and Automated PDF Report

The pipeline also produces a set of visualization images and an automatically assembled PDF report that summarizes the entire workflow for each processed protein. This allows the user to quickly evaluate system stability and verify that equilibration proceeded correctly.

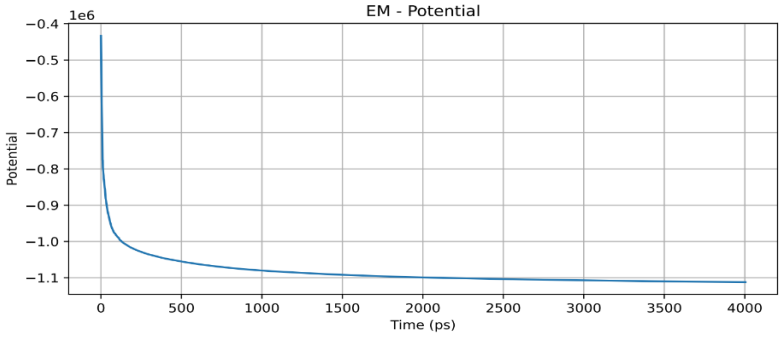
7.1 Images Automatically Generated by the pre_analysis Script

During execution of **pre_analysis.py**, the script analyzes the .xvg output files from each stage (EM, NVT, NPT, Replica 1, and Replica 2). For each property, the script automatically generates high-quality PNG images using Matplotlib. These plots are saved directly inside the protein's output directory.

The images typically include:

Energy Minimization (EM)

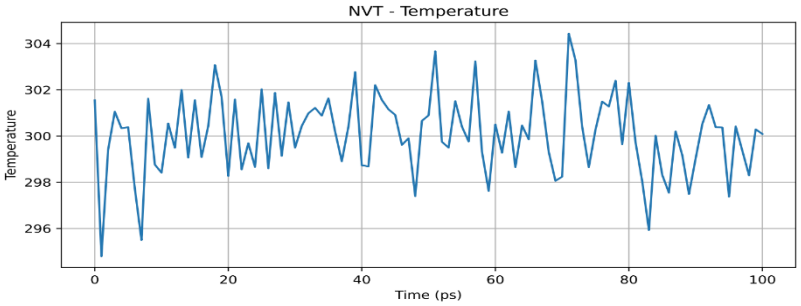
- **Potential Energy vs. Minimization Steps**
  Shows whether the energy smoothly decreases during minimization. A monotonic decay indicates proper removal of steric clashes.



Potential energy shows drift of -678883.84 units.

NVT Equilibration

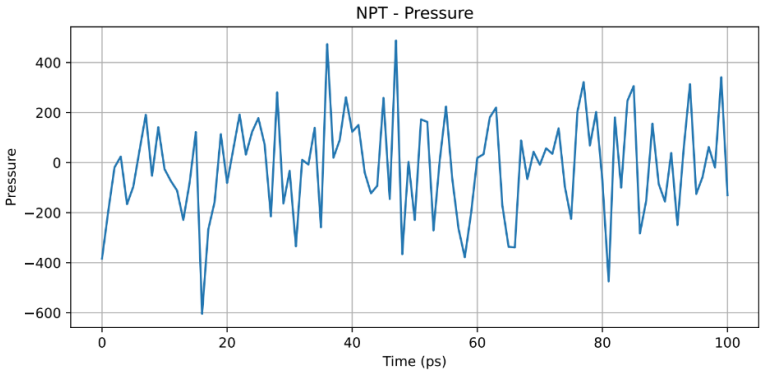
- **Temperature vs. Time**
  Checked for stability around the target (usually 300 K).
- **Potential Energy vs. Time**
  Should remain relatively consistent after thermal equilibration.



Temperature is stable: mean=300.08 ± 1.71 K.

NPT Equilibration

- **Temperature vs. Time**
- **Pressure vs. Time**
  Natural fluctuations are expected, but values should be centered near 1 bar.
- **Density vs. Time**
  Ideally approaches ~1000 kg/m³ for TIP3P water.



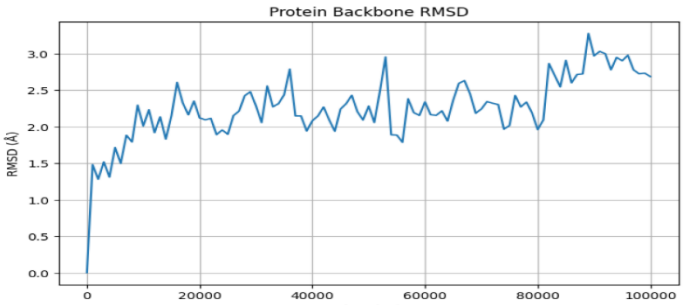
Pressure fluctuations are large: mean=-13.82 ± 202.30 bar.

7.2 Images Automatically Generated from Replica_analysis script

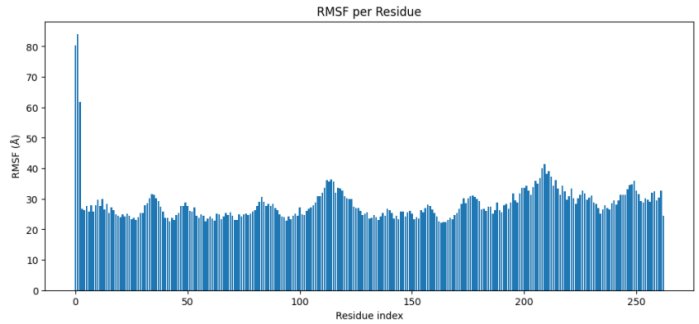
After the main MD simulations are complete, the replica analysis script processes the trajectories of Replica 1 and Replica 2 to evaluate the structural stability of the protein. Using the simulation outputs (.xtc and .tpr), the script automatically calculates key structural metrics: **RMSD**, **RMSF**, and **Radius of Gyration (Rg)**.

For each replica, the script generates high-resolution images using Matplotlib:

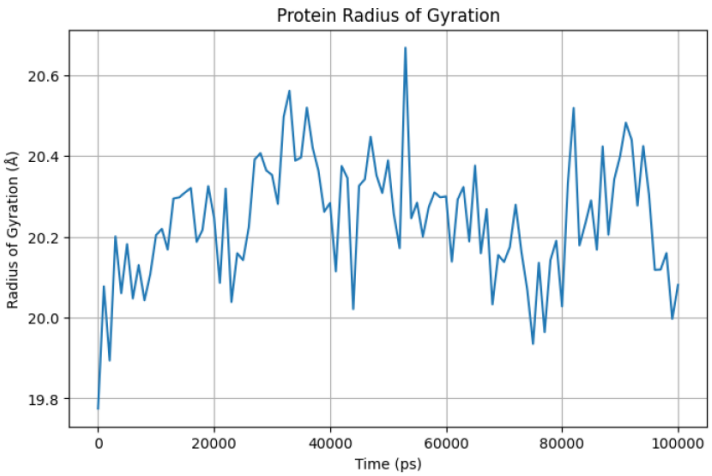
- **RMSD plots:** Show the time evolution of backbone RMSD for each replica, providing insight into the overall structural stability and equilibration.



- **RMSF plots:** Depict per-residue fluctuations to identify flexible regions (loops, termini) and rigid cores of the protein.



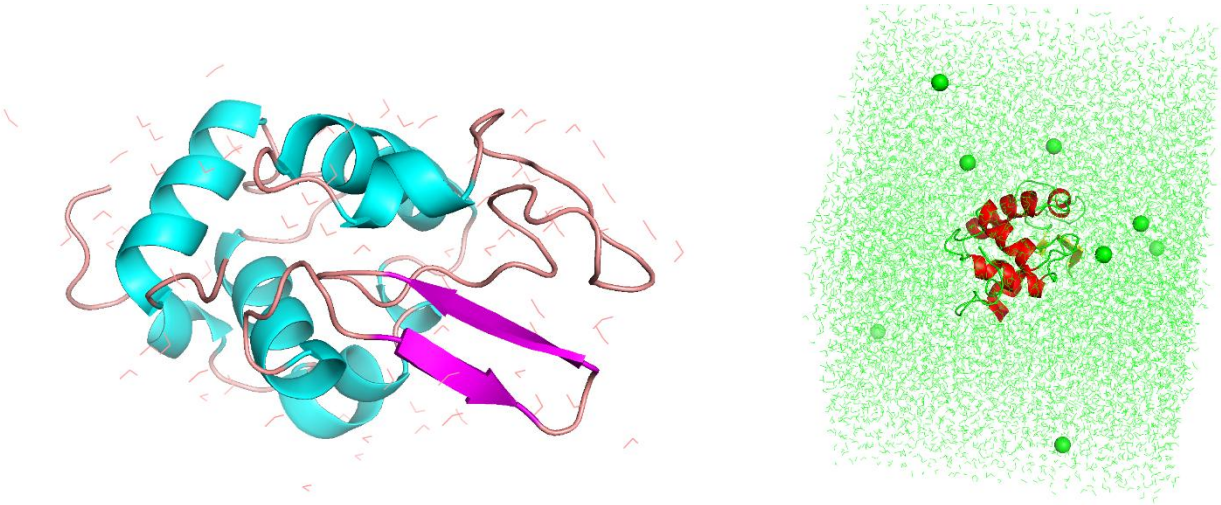
- **Radius of Gyration plots (Rg):** Visualize changes in the protein's compactness throughout the simulation, helping detect potential unfolding events.



Each plot includes a **conditional comment** beneath it, indicating whether the metric is within acceptable thresholds (e.g., RMSD <3 Å, Rg stable within ±2%, RMSF within typical loop/core ranges). The comments also provide recommendations for further improvement if instabilities are detected, such as extending the equilibration, adjusting restraints, or reviewing simulation parameters.

7.3 Structural Visualization Images Generated from the Simulation (Using PyMOL)

In addition to the numerical plots and PDF reports, the workflow can also incorporate structural images generated directly from the simulation trajectory using PyMOL. These images provide a visual confirmation that the system remains stable throughout equilibration and during the replica MD runs. The user can load any of the produced trajectory files (e.g., nvt.xtc, npt.xtc, mdrep1_final.xtc, mdrep2_final.xtc) together with the corresponding structure files (*.tpr, *.gro, or *.pdb) into PyMOL to create high-quality molecular snapshots that visually document the behavior of the system at different stages.



Typically, these PyMOL images include:

1. Initial Solvated Structure

A snapshot of the fully solvated protein after the solvation and ionization steps.
This image shows:

- the protein inside its periodic box
- uniform water distribution
- ions placed in electrostatically reasonable positions

This serves as a reference before minimization and equilibration.

2. Post-Energy Minimization Structure

A PyMOL capture of the minimized system to confirm that:

- steric clashes have been resolved
- the protein retains a physically reasonable conformation

- no distortions or collapsed regions occurred during EM

This image visually complements the potential energy plot.

3. NVT Equilibrated Structure

A snapshot from the end of the NVT equilibration.
Expected features include:

- stable protein geometry
- well-packed solvent
- no abnormal deformations or vacuum bubbles
- box shape preserved

Users typically verify that the system visually appears stable at the target temperature.

4. NPT Equilibrated Structure

The NPT snapshot reflects the system after pressure adjustment.
It typically shows:

- slight box size changes
- uniform water density
- stable protein orientation

Any unusual resizing, vacuum spaces, collapsed protein regions, or water inhomogeneities can be detected visually in this step.

5. Replica MD Snapshots (Replica 1 & 2)

These images demonstrate how the protein behaves under short production runs with different random seeds.
The images commonly include:

- backbone alignment between frames
- overall structural stability
- absence of unfolding or unusual deviations
- solvent still evenly distributed

These final frames act as visual confirmation that the system is ready for long-term production MD.