p16_trajectory_analysis

***Tutorial 2***

## Comparative analysis of structural changes of the P16 protein (CDKN2A gene) wild type (WT) with mutants:

# 1. Analysis using basic MD metrics — RMSD, Rg, RMSF :

*After the initial relaxing 200 ns production run, we evaluate the protein trajectory using the basic MD analysis metrics:*

### *1.1 RMSD (оценка стабильности структуры во времени)*

```
gmx rms -s md_200ns.tpr -f md_200ns_fit.xtc -o rmsd.xvg -tu ns
```

Groups:
Fit group → 1(Protein)
RMSD group → 1(Protein)

**-s md_200ns.tpr** — *input tpr file (reference structure + parameters)*
**-f md_200ns_fit.xtc** — *trajectory after centering and PBC correction*
**-o rmsd.xvg** — *output file with RMSD over time*
**-tu ns** — *output time in nanoseconds (ns) instead of ps*

Output file:
**rmsd.xvg** — *RMSD(t) plot over the entire 200 ns trajectory.*

*Plot the .xvg graphs using any convenient tool (xmgrace, Python, Excel, etc.):*

*Output the statistics:*

```
gmx analyze -f rmsd.xvg
```

### *1.2 Radius of Gyration (Rg)*

```
gmx gyrate -s md_200ns.tpr -f md_200ns_fit.xtc -o gyrate.xvg
```

Group: Protein

Output file:
**gyrate.xvg** — *change in protein compactness (Rg) over time.*

*Statistics:*

```
gmx analyze -f gyrate.xvg
```

### *1.3 RMSF (per-residue fluctuations)*

```
gmx rmsf -s md_200ns.tpr -f md_200ns_fit.xtc -o rmsf.xvg -res
```

Select the group: Protein

Output file:
**rmsf.xvg** — *RMSF for each protein residue (156 aa).*

*Statistics:*

```
gmx analyze -f rmsf_WT.xvg
```

## 2. Comparative analysis

*As the wild-type (WT) template, we use the last frame (PDB) of the initial 200 ns production molecular dynamics run, taken from the centered trajectory(.xtc).*

*The resulting structure is cleaned of water molecules and ions, after which it is used as the starting structure for performing in silico mutagenesis (using third-party software, e.g., ChimeraX, Maestro Schrödinger).*
*(Files: **p16_p48l_h83y.pdb**, **p16_r131p_g55c.pdb**)*

*The mutations selected for comparison with WT correspond to real clinical cases and were chosen based on patient data available on cBioPortal — an international platform for the analysis of cancer genomics data.*

*Mutations*:
**P48L + H83Y:** *https://www.cbioportal.org/patient?sampleId=P-0006661-T01-IM5&studyId=pdac_msk_2024*
**R131P + G55C:** *https://www.cbioportal.org/patient?sampleId=P-0003222-T01-IM5&studyId=lung_msk_2017*

*Both the mutants and the WT undergo the standard pipeline followed by a 200 ns production run (as described in Tutorial 1).*

### *2.1 Comparison using basic metrics (RMSD, Rg, RMSF):*

*Use the commands described in the previous section* (`gmx rms, gmx gyrate, gmx rmsf`).
Combine the plots (the .xvg files) using any convenient method (xmgrace, Python, Excel, etc.).

*Using* `gmx analyze` *collect the data into a similar table for further comparison:*

| Белок | RMSD Avg | RMSD Std | RMSF Avg | RMSF Std | Rg Avg | Rg Std |
|---|---|---|---|---|---|---|
| WT | | | | | | |
| P48L/H83Y | | | | | | |
| R131P/G55C | | | | | | |

### *2.2 Protein Energy:*

```
gmx energy -f WT.edr -o WT_energy.xvg
gmx energy -f P48L_H83Y.edr -o P48L_H83Y_energy.xvg
gmx energy -f R131P_G55C.edr -o R131P_G55C_energy.xvg
```

- ***Temperature** should fluctuate around the target value (e.g., 300 K) with RMSD ~1–5 K and without noticeable drift.*
- ***Pressure in the NPT** ensemble exhibits large fluctuations (RMSD ~50–500 bar); evaluation is based on the average value, which should be close to 1 bar.*
- ***Potential energy** should be stable, fluctuating around a mean value with minimal drift relative to |E|.*
- ***Tot-Drift** should be small; significant values indicate numerical instability of the simulation.*

## *2.3 Intra-protein H-bonds:*

*Look at the average number of hydrogen bonds over the entire trajectory*

**Example command (WT):**

```
gmx hbond -s WT.tpr -f md_200ns_fit.xtc -num WT_hbnum_PP.xvg -WT_hbonds_PP.ndx
```

**-s WT.tpr** — *input tpr file with system topology and parameters*
**-f md_200ns_fit.xtc** — *trajectory*
**-num WT_hbnum_PP.xvg** — *output number of hydrogen bonds over time (Protein–Protein)*
**-o WT_hbonds_PP.**ndx — *save the list of all donor–acceptor pairs (index file)*

Select: `group "Protein"`

Output files:
**WT_hbnum_PP.xvg** — *time evolution of the number of Protein–Protein hydrogen bonds (time → H-bond count)*
**WT_hbonds_PP.ndx** — *complete list of donor–acceptor pairs within the protein*

Result check:

```
gmx analyze -f WT_hbnum_PP.xvg
```

## *2.4 DSSP analysis (secondary structure):*

** **mkdssp** must be pre-installed*

**Example command (WT):**

```
gmx dssp -s WT.tpr -f md_200ns_fit.xtc \
  -o WT_dssp.dat -num WT_dssp_num.xvg \
  -sel 'group "Protein"'
```

**gmx dssp** — *calculation of protein secondary structure (DSSP)*
**-s WT.tpr** — *system topology*
**-f md_200ns_fit.xtc** — *centered trajectory without PBC artifacts*
**-o WT_dssp.dat** — *DSSP annotation by frames and residues*
**-num WT_dssp_num.xvg** — *numerical statistics of H/E/C over time*
**-sel 'group "Protein"'** — *analyze protein only*

Output:
**WT_dssp.dat** — *secondary structure assignment for analysis and plotting*
**WT_dssp_num.xvg** — *control statistics*

**For analysis and visualization of the protein secondary structure over time, you can use the script* **sse_P16.ipynb**

## *2.5 PCA (Principal Component Analysis) = analysis of the principal components of motion.*

*Compute the covariance matrix of atomic positional fluctuations → calculate its eigenvectors (PC1, PC2, PC3 …).*

**Example command (WT):**

```
gmx covar \
  -s WT.tpr \
  -f md_200ns_fit.xtc \
  -o WT_eigenval.xvg \
  -v WT_eigenvec.trr \
  -av WT_average.pdb \
  -b 100000
```

Select a group: 4
Selected 4: 'Backbone'

**-s WT.tpr** — *system topology*
**-f md_200ns_fit.xtc** — *centered trajectory without PBC artifacts*
**-o WT_eigenval.xvg** — *eigenvalues (contribution of each mode)*
**-v WT_eigenvec.trr** — *eigenvectors (principal motions)*
**-av WT_average.pdb** — *average protein structure*
**-b 100000** — *start analysis from 100 ns (initial relaxation is discarded)*

Output:
**WT_eigenval.xvg** — *mode spectrum (PCA)*
**WT_eigenvec.trr** — *trajectory of principal components*
**WT_average.pdb** — *average structure*

***PCA 1 vs 2 :***

*This is the projection of the trajectory onto the first and second principal components (PC1 and PC2). It shows the main conformational landscape.*

**Example command (WT):**

```
gmx anaeig \
  -v WT_eigenvec.trr \
  -s WT.tpr \
  -f md_200ns_fit.xtc \
  -2d WT_2D_PC1_PC2.xvg \
  -b 100000 \
  -first 1 \
  -last 2
```

```
Select a group: 4/4
Selected 4: 'Backbone'
```

**-v WT_eigenvec.trr** — *eigenvectors from gmx covar*
**-s WT.tpr** — *system topology*
**-f md_200ns_fit.xtc** — *centered trajectory*
**-2d WT_2D_PC1_PC2.xvg** — *projection of the trajectory onto PC1–PC2*
**-b 100000** — *start analysis from 100 ns*
**-first 1** — *first principal component (PC1)*
**-last 2** — *second principal component (PC2)*

Output:
**WT_2D_PC1_PC2.xvg — *2D conformational landscape (PC1 vs PC2)***

***PCA 1 vs 3 :***

*Reveals additional / secondary motions that may not be visible in PC1–PC2 (alternative PCA projection).*

**Example command (WT):**

```
gmx anaeig \
  -v WT_eigenvec.trr \
  -s WT.tpr \
  -f md_200ns_fit.xtc \
  -2d WT_2D_PC1_PC3.xvg \
  -b 100000 \
  -first 1 \
  -last 3
```

```
Select a group: 4/4
Selected 4: 'Backbone'
```

*Plot the graphs. They can be overlaid on top of each other*

### *2.6 ProLif is a library for analyzing interactions in MD trajectories.*

*Analysis of MD trajectories that shows which amino acids interact with each other, what types of contacts they form, and how often over time; it is needed to quantitatively compare interactions in WT and mutants and understand which structural contacts are preserved, lost, or newly formed.*

*Detailed tutorial here:* *https://prolif.readthedocs.io/en/stable/source/tutorials.html*
*Example calculation for our trajectories -* **P16_prolif.ipynb**

### *2.7 Visual analysis of trajectories*

*For visualization, we will use* ***VMD.***

*After completing the MD simulation, first load the structure, then overlay the trajectory. To do this, launch VMD, open* ***File → New Molecule****, and load the structure file (.gro). Then, in the same window, click* ***Load Files*** *and add the trajectory file (.xtc). The trajectory will automatically play as an animation over the original structure.*

*In VMD, the display style is configured via* ***Graphics → Representations****: typically, a representation is created with the selection* `protein` *and the* ***NewCartoon*** *drawing method enabled, colored by secondary structure. Water can either be completely hidden by setting the selection to* `not water`*, or displayed separately in a simplified form (e.g.,* ***Lines*** *or* ***Points****) to avoid cluttering the visualization.*

*The visualization can be recorded using standard screen recording tools.*

***Official VMD tutorials:***

*https://www.ks.uiuc.edu/Training/Tutorials/vmd/tutorial-html/*
*https://www.ks.uiuc.edu/Training/Tutorials/vmd-imgmv/imgmv/imgmv-tutorial.pdf*