

Curso práctico de Dinámica Molecular y Análisis de Trayectorias

Episodio 5: Análisis de trayectorias

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya
Facultat de Ciències, Tecnologia i Enginyeries (FCTE)

jordi.villa@uvic.cat

Curso MD y Análisis de Trayectorias
Concepción, enero 2026

1 Episodio 5: Análisis de trayectorias

- Preprocesado y limpieza
- Velocidades e inercia
- Reducción y clustering
- Energías libres y superficies
- Estadística y confianza
- Resumen

Pipeline previo al análisis

- ① Convertir y limpiar trayectorias con `simulatePdb.py` o `simulateGromacs.py`.
- ② Alinear estructuras y eliminar solvente irrelevante.
- ③ Definir subconjuntos de átomos y escribir DCD/NetCDF para análisis posteriores.

Scripts del Application Layer de OpenMM. [2]

Velocidades y distribuciones

- 'StateDataReporter' y 'DCDReporter' construyen histogramas de v_i y energías.
- Por el teorema de equipartición: $\langle K \rangle = \frac{3N}{2} k_B T$ para sistemas clásicos.
- Controlar $C(t) = \langle A(0)A(t) \rangle$ arroja las escalas relevantes.

RMSD y alineamientos

$$\text{RMSD}(t) = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\mathbf{r}_i(t) - \mathbf{r}_i^{\text{ref}}\|^2}.$$

- Usamos el algoritmo de Kabsch para rotaciones óptimas.
- Documentamos con `simulateAmber.py` y la salida en CSV para cada réplica.

Radio de giro y tensor de inercia

$$I_{ab} = \sum_i m_i (r_i^2 \delta_{ab} - r_{ia} r_{ib}) .$$

- Autovalores de I describen la anisotropía global.
- Monitorear $R_g^2 = \frac{1}{N} \sum_i m_i \|\mathbf{r}_i - \mathbf{r}_{CM}\|^2$ por réplica.

Extremos y contactos

- Mapas binarios con umbral de 0.45 nm generan matrices de contacto.
- Las columnas representan cambios de dominios detectados por `simulateCharmm.py`.

Covarianza y PCA

$$C = \langle (\mathbf{x} - \langle \mathbf{x} \rangle)(\mathbf{x} - \langle \mathbf{x} \rangle)^T \rangle.$$

- Proyectamos sobre los primeros autovectores y definimos ejes de evento.
- Relacionamos con `coarse_grained_polymer.py` para comparar modos lentos de polímeros y proteínas.

Agrupamiento jerárquico

- Usamos enlaces promedio e índice silhouette para decidir k .
- Una mala elección produce microestados vacíos; preferimos ensayos progresivos.

Transiciones y grafos

- Grafos de estados (ver 'graph.png') representan canales cinéticos.
- Los bordes anotados reflejan flujos medidos por scripts de OpenMMTools.

Autovalores y modos

$$C\mathbf{w} = \lambda\mathbf{w}, \quad \lambda \in \mathbb{R}_+.$$

- Los modos dominantes identifican movimientos colectivos.
- Guardamos proyecciones para alimentar modelos de Markov.

Funciones de densidad

$$g(r) = \frac{1}{4\pi r^2 \rho} \left\langle \sum_{i \neq j} \delta(r - r_{ij}) \right\rangle.$$

- Comparar con `argon-chemical-potential.py` para validar.
- Usa histogramas reponderados para estimar $F(r) = -k_B T \ln g(r)$.

Superficies de energía libre y barreras

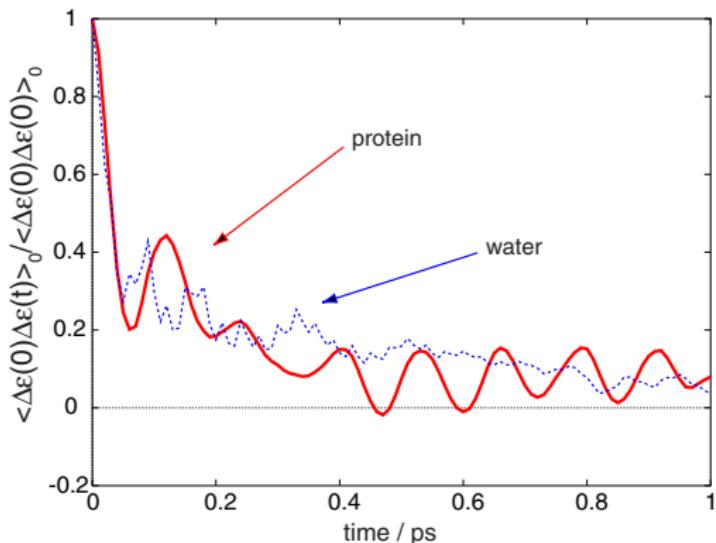
- La proyección $F(s) = -k_B T \ln P(s)$ requiere CVs robustos.
- Integrar con ventanas de ‘umbrella sampling’ y combinar con MBAR.
- Comparar perfiles unidimensionales con `simulateAmber.py`.
- Validar con histogramas solapados manualmente.

Bootstrap y bloqueos

- Re-muestrear bloques para estimar varianzas.
- Las autocorrelaciones (ver `bs_autocorrelation.pdf`) determinan el tamaño del bloque.

Autocorrelaciones de bloques

Autocorrelation of the energy gap
in water and protein for the LADH system



Serie de errores

$$\sigma_{\bar{A}} = \frac{\sigma_A}{\sqrt{N_{\text{eff}}}}, \quad N_{\text{eff}} = \frac{N}{g}, \quad g = 1 + 2 \sum_t \rho(t).$$

Reportes finales

- Generar tablas de RMSD/RMSF/radio de giro con `simulateCharmm.py`.
- Exportar CSV/HDF para facilitar el paso a PyEMMA/Deeptime.

Resumen y siguiente paso

- Preprocesado sin PBC artefactual.
- CVs estables y histogramas convergentes.
- Reportes con errores y trayectorias documentadas.
- Usar los features preparados como entrada para PyEMMA/Deeptime.

Referencias I

- [1] OpenMM Cookbook. *Autocorrelation example from Selecting Values for Simulation Parameters.* CC BY-SA 4.0. URL: https://openmm.github.io/openmm-cookbook/latest/notebooks/tutorials/simulation_parameters.html (visited on 01/12/2026).
- [2] OpenMM developers. *OpenMM application guide.* Referencia para scripting y ejemplos de workflow. 2024. URL: <https://docs.openmm.org/latest/userguide/application.html>.