

# Curso práctico de Dinámica Molecular y Análisis de Trayectorias

## Episodio 6: Modelos de Markov con PyEMMA

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 6: Modelos de Markov con PyEMMA
  - Preparando trayectorias
  - Discretización y clustering
  - MSM y validación
  - Transiciones y rutas
  - Resumen

# Transformar datos

- 'simulateAmber.py' y 'simulateCharmm.py' producen trayectorias DCD que alimentamos a PyEMMA.
- Alinear, recortar solvente y guardar en HDF5 para economía de almacenamiento.
- Reportes de energía y RMSD se usan como checks previos a la discretización.

# Extracción de características

- Distancias, dihedros y contactos entre ligando y proteína forman el vector  $\mathbf{x}$ .
- La matriz de covarianza  $C = \langle (\mathbf{x} - \langle \mathbf{x} \rangle)(\mathbf{x} - \langle \mathbf{x} \rangle)^T \rangle$  se proyecta.
- Todos los scripts de la guía (simulateGromacs, simulateTinker) comparten esta salida para alimentar PyEMMA.

$$C_{\tau} \mathbf{w} = \lambda C_0 \mathbf{w}.$$

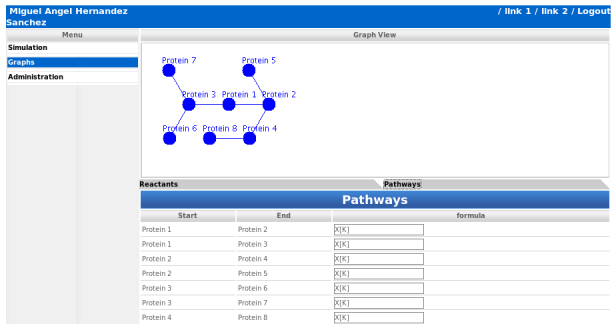
- Seleccionamos  $\tau$  comparando mesetas en los tiempos implícitos  $t_i = -\frac{\tau}{\ln \lambda_i}$ .
- tlCA captura los modos lentos que nos interesan para MSM y Deeptime.

# Clustering de microestados

- k-means/minibatch generan microestados; silhouette ayuda a escoger  $k$ .
- Preferimos estados densos para evitar microestados vacíos (OpenMM User Guide §3.7).

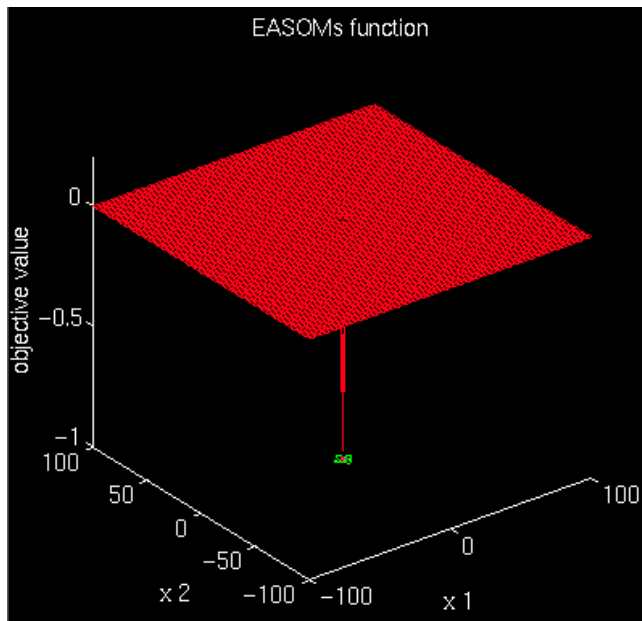
# Mapa de estados

## BioBridge



Grafo de transiciones influenciado por los resultados del tutorial REST. [1]

# Paisaje energético y particiones





$$T_{ij}(\tau) = \frac{C_{ij}(\tau)}{\sum_k C_{ik}(\tau)}.$$

- Las filas suman 1 y la matriz es estocástica reversible cuando se impone detailed balance.
- Utilizamos estimadores bayesianos con pseudoconteos para evitar ceros.

$$P(X_{t+\tau} \mid X_t, \dots) = P(X_{t+\tau} \mid X_t).$$

- Prueba Chapman-Kolmogorov y tests de pseudo-time (PyEMMA) validan la Markovianidad.

$$\pi^T T = \pi^T.$$

- La distribución estacionaria  $\pi$  define poblaciones relativas.
- Asociamos con energías libres  $F_i = -k_B T \ln \pi_i$  y comparamos con  $\Delta G$  alquímicos.

$$t_i = -\frac{\tau}{\ln \lambda_i}.$$

- Los autovalores cercanos a 1 representan procesos lentos (Markovianidad).
- Las diferencias grandes entre  $t_i$  respaldan la separación de escalas.

# Incertidumbre y regularización

- Remuestrear trayectorias discretizadas para intervalos confiables.
- PyEMMA guarda pesos y permite reproducir con semilla distinta.
- Imponemos reversibilidad para reducir ruido.
- Ajustamos priors a partir de las fuerzas detectadas en 'simulateAmber.py'.

# TPT, committor y flujos

- Flujos  $f_{ij} = \pi_i T_{ij} q_i (1 - q_j)$  identifican rutas reactivas.
- El committor  $q_i = P(\text{llegar a B antes que a A} \mid X_0 = i)$  define el progreso.
- Grafo de transiciones coloreado por flujo destaca cuellos de botella cinéticos.
- 'simulateGromacs.py' proporciona las trayectorias de referencia.

# Checklist MSM

- Features informativos.
- $\tau$  validado y autovalores con gap claro.
- Bootstrap de errores y reporte con referencias.