

Practical Course on Molecular Dynamics and Trajectory Analysis

Episode 6: Markov models with PyEMMA

Jordi Villà i Freixa

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

jordi.villa@uvic.cat

MD Course and Trajectory Analysis
Concepcion, January 2026

- 1 Episode 6: Markov models with PyEMMA
 - Preparing trajectories
 - Discretization and clustering
 - MSM and validation
 - Transitions and pathways
 - Summary

Transform data

- `simulateAmber.py` and `simulateCharmm.py` produce DCD trajectories that feed into PyEMMA.
- Align, trim solvent, and save to HDF5 for storage efficiency.
- Energy and RMSD reports serve as checks before discretization.

Feature extraction

- Distances, dihedrals, and ligand–protein contacts form the vector \mathbf{x} .
- The covariance matrix $C = \langle (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T \rangle$ gets projected.
- All guide scripts (simulateGromacs, simulateTinker) share this output to feed PyEMMA.

tlCA and time-lagged reduction

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

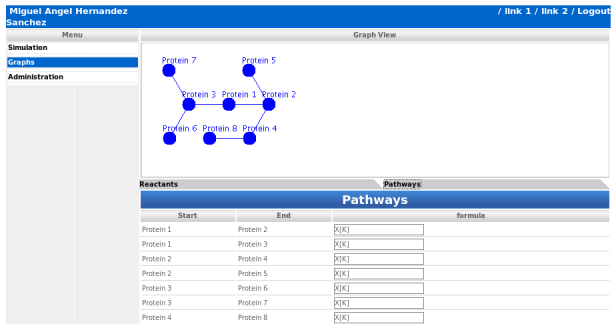
- Choose τ by comparing plateaus in implicit times $t_i = -\frac{\tau}{\ln \lambda_i}$.
- tlCA captures the slow modes relevant for MSM and Deeptime.

Microstate clustering

- k-means/minibatch create microstates; silhouette helps choose k .
- Favor dense states to avoid empty microstates (OpenMM User Guide §3.7).

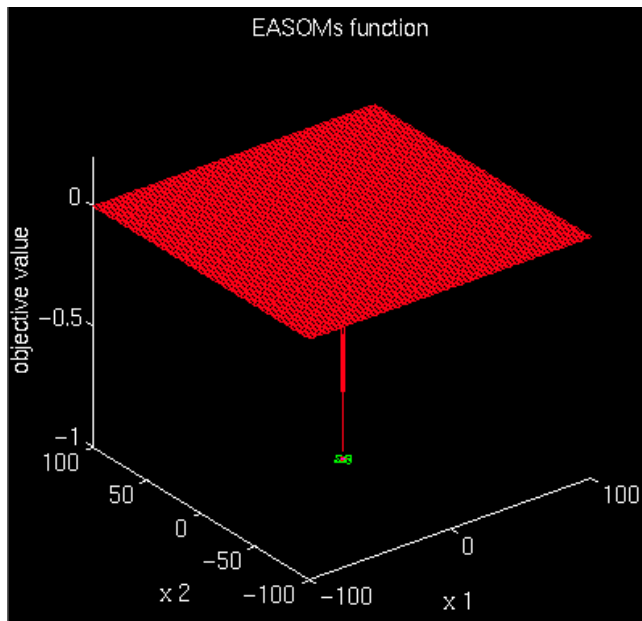
State map

BioBridge



Transition graph influenced by the REST tutorial results. [1]

Energy landscape and partitions



Transition matrix

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

- Rows sum to 1 and the matrix becomes reversible stochastic when detailed balance is enforced.
- Use Bayesian estimators with pseudocounts to avoid zeros.

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

- Chapman-Kolmogorov tests and pseudo-time diagnostics (PyEMMA) validate Markovianity.

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

- The stationary distribution π defines relative populations.
- Relate to free energies $F_i = -k_B T \ln \pi_i$ and compare with alchemical ΔG .

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

- Eigenvalues near 1 represent slow processes (Markovianity).
- Large gaps between t_i support scale separation.

Uncertainty and regularization

- Resample discretized trajectories to obtain reliable intervals.
- PyEMMA stores weights and allows reruns with a different seed.
- Enforce reversibility to reduce noise.
- Tune priors using forces detected in `simulateAmber.py`.

TPT, committor, and fluxes

- Flows $f_{ij} = \pi_i T_{ij} q_i (1 - q_j)$ identify reactive pathways.
- The committor $q_i = P(\text{reaching B before A} \mid X_0 = i)$ gauges progress.
- A transition graph colored by flux highlights kinetic bottlenecks.
- `simulateGromacs.py` provides the reference trajectories.

MSM checklist

- Informative features.
- Validated τ and eigenvalues with a clear gap.
- Error bootstrap and referenced reporting.

References I

- [1] OpenMM Cookbook. *Grafos de estados en Replica Exchange Solute Tempering*. CC BY-SA 4.0. URL: https://openmm.github.io/openmm-cookbook/latest/notebooks/tutorials/Running_a_REST_simulation.html (visited on 01/12/2026).
- [2] Visualización interna del curso. *Embudo energético de muestreo*. Figura adaptada del archivo `function_funnel.png` del repositorio. 2024.