Markov State Model Review

H.Koh

2024.11.14.

Overview

Non – Markovian Dynamical Approaches

Molecular Dynamics → Markov State Model → qMSM → IGME

Molecular Dynamics

+ Complementary to experimental approaches
+ Provide atomistic details of protein dynamics

- Required timescale often too large

Markov State Model

+ Long timescale dynamics
+ A large ensemble of short MD

- Hard to meet Memoryless conditions

qMSM

+ Introduce Memory Kernel K
+ Generalized Master Equation

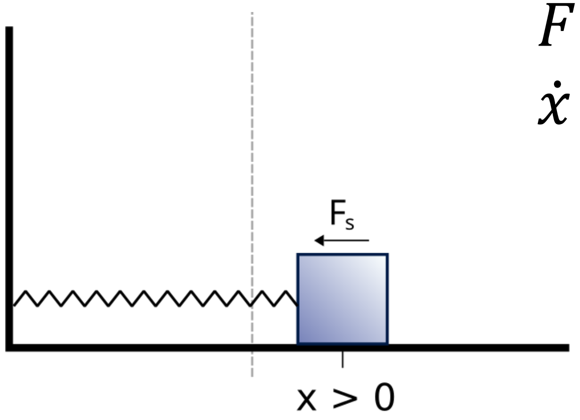
- Issue of robustness
- Numerical instabilities in K

IGME

+ Integrate Memory Kernel first
+ Fit analytic results with MD
+ Enhanced numerical stability

Markov State Model

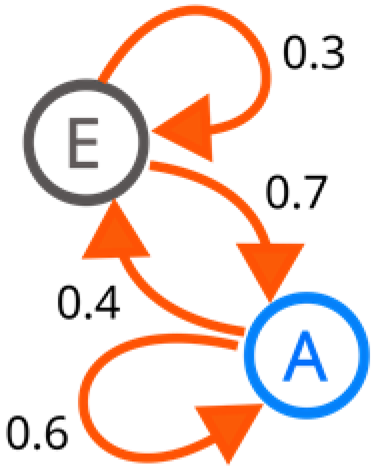
Wikipedia : Harmonic Oscillator



$$F = \dot{p} = -kx \ ,$$
$$\dot{x} = v = p/m$$

$$\begin{bmatrix} \dot{p} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & -k \\ 1/m & 0 \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix}$$

Wikipedia : Markov Chain



$$\begin{bmatrix} E_{i+1} \\ A_{i+1} \end{bmatrix} = \begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix} \begin{bmatrix} E_i \\ A_i \end{bmatrix}$$

$$x_{i+1} = \mathbb{T} \, x_i$$

Newtonian Physics

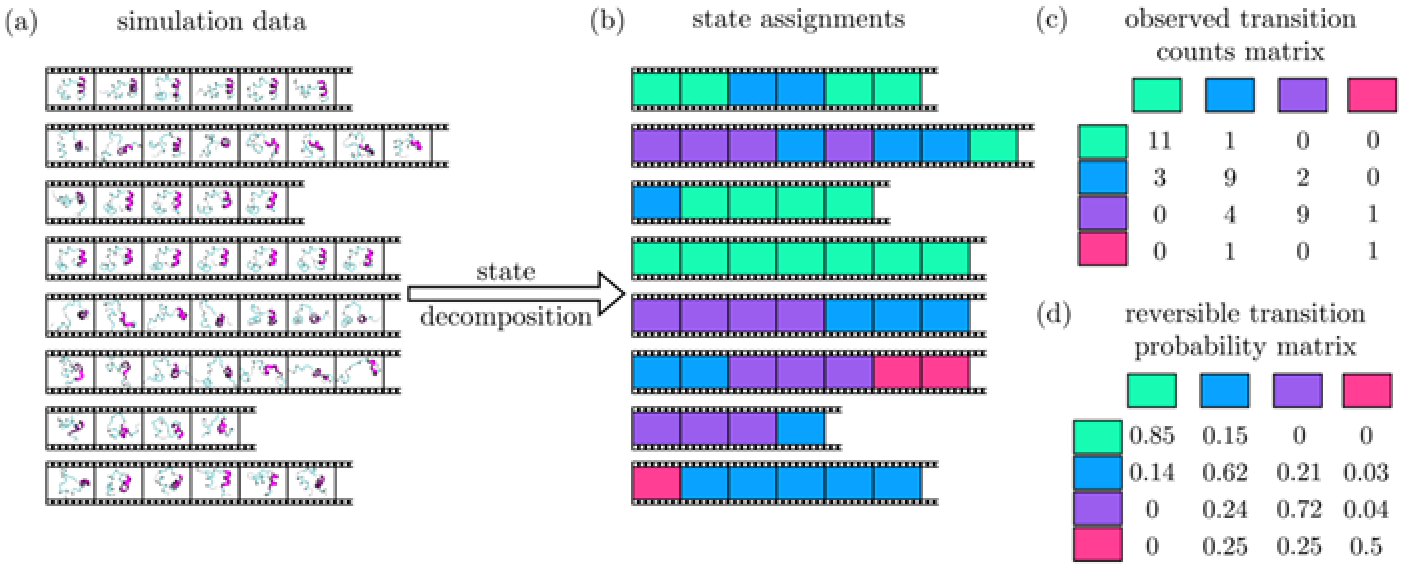
- once you know position and velocity (momentum), every dynamics is determined
- does not depend on previous history
- generalization : Liouville equation ?

Markov State Model ($x(t+\tau) = \mathbb{T}(\tau)\, x(t)$)

- n x n transition probability matrix $\mathbb{T}(\tau)$
- entire configuration space -> divided into n states (n x 1 column vector $x(t)$)
- lag time τ
- Memoryless condition (Markovian)
  : $\mathbb{T}$ does not depend on previous history ($x(t)$)

Markov State Model

J. Am. Chem. Soc. 2018, 140, 2386−2396



A

$\mathbb{T} = normalized\ (A + A^T)/2$

If we can choose

- n states that **capture the dynamics** of the system
- long enough τ to be **Markovian**
- short enough τ to **resolve the system dynamics**

then MSM can predict long-timescale dynamics based on a large ensemble of short MD simulations.

However, MSM has **limitations**

- **dynamic relaxation time < τ**
- **τ < length of individual MD simulations** available to estimate transition probabilities
- **large n → fast relaxation time**, but **hinder** the comprehension of biological mechanisms (Over fitting?)

Solution : **Non-Markovian Dynamics Approaches**?

Generalized Master Equation (GME) Method

$$\dot{T}(t) = \dot{T}(0)T(t) - \int_0^t d\tau\, \mathcal{K}(\tau)T(t-\tau)$$

$\mathcal{K}(\tau)$: memory kernel
τ_K : memory kernel relaxation time
$\mathcal{K}(t \geq \tau_K) \approx 0$

$$\frac{\dot{T}(t)}{T(t)} = \frac{\dot{T}(0)}{T(0)} - \int_0^t d\tau\, \mathcal{K}(\tau) \frac{T(t-\tau)}{T(t)}, \quad T(0) = \mathbb{I}_{n\times n}$$

- If $\mathcal{K}(\tau) = 0$ (No memory),

$$\frac{\dot{T}(t)}{T(t)} = \frac{\dot{T}(0)}{T(0)} \rightarrow \frac{d\ln T(t)}{dt} = C \rightarrow T(t) = \exp(Ct)$$

- If $\mathcal{K}(\tau) \neq 0$,

$$\ddot{T}(t) = \dot{T}(0)\dot{T}(t) - \mathcal{K}(t)\,T(0) = \dot{T}(0)\dot{T}(t) - \mathcal{K}(t)$$

$$\mathcal{K}(t) = \dot{T}(0)\dot{T}(t) - \ddot{T}(t)$$

Memory kernel $\mathcal{K}(t)$ is something related with the second derivative of function (matrix) $T(t)$?

Generalized Master Equation (GME) Method

$$\dot{\boldsymbol{T}}(t) = \boldsymbol{T}(t)\dot{\boldsymbol{T}}(0) - \int_0^{\boxed{\min[\tau_K, t]}} \boldsymbol{T}(t-s)\boldsymbol{K}(s)ds,$$

Brute-force method utilizing above equation

→ **quasi MSM (qMSM)**

- Theoretically rigorous
- Issue in ensuring the Robustness
- Numerical instability in the computation of $\boldsymbol{K}(\boldsymbol{s})$

Integrate until $K(s)$ is fully decayed.

$$\boldsymbol{M}_n = \int_0^{\tau_K} \boldsymbol{K}(s)s^n ds$$

$$\boldsymbol{T}(t)^{-1}\frac{d}{dt}\boldsymbol{T}(t) = \dot{\boldsymbol{T}}(0) - \boldsymbol{M}_0 - \sum_{n=1}^{\infty}\left[\frac{(-1)^n}{n!}\boldsymbol{T}(t)^{-1}\frac{d^n}{dt^n}\boldsymbol{T}(t)\right]\boldsymbol{M}_n$$

Integration of Memory Kernel $\boldsymbol{K}(\boldsymbol{s})$
(avoid numerical instability)

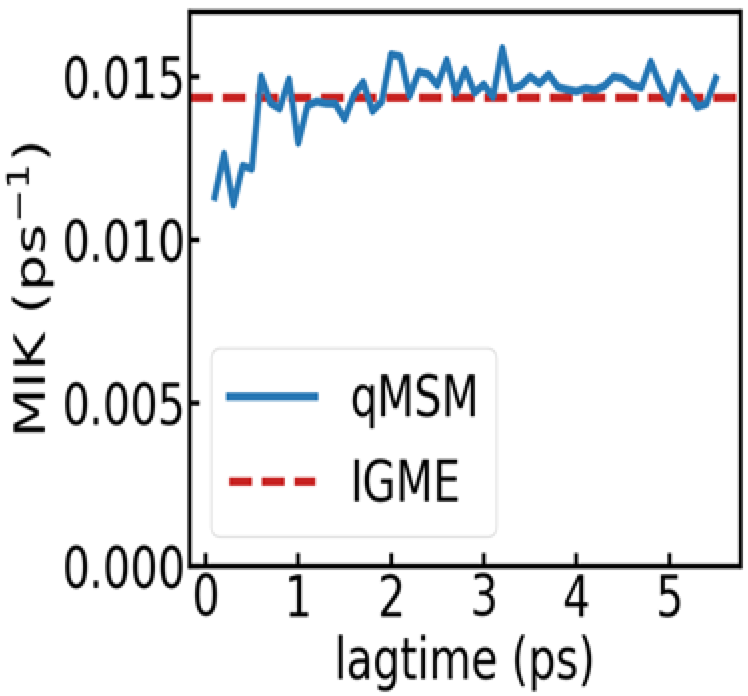
→ **Integrative GME (IGME)**

- Derive Analytic Solution for the GME
- Fitting to MD simulations
- Determine hyperparameters

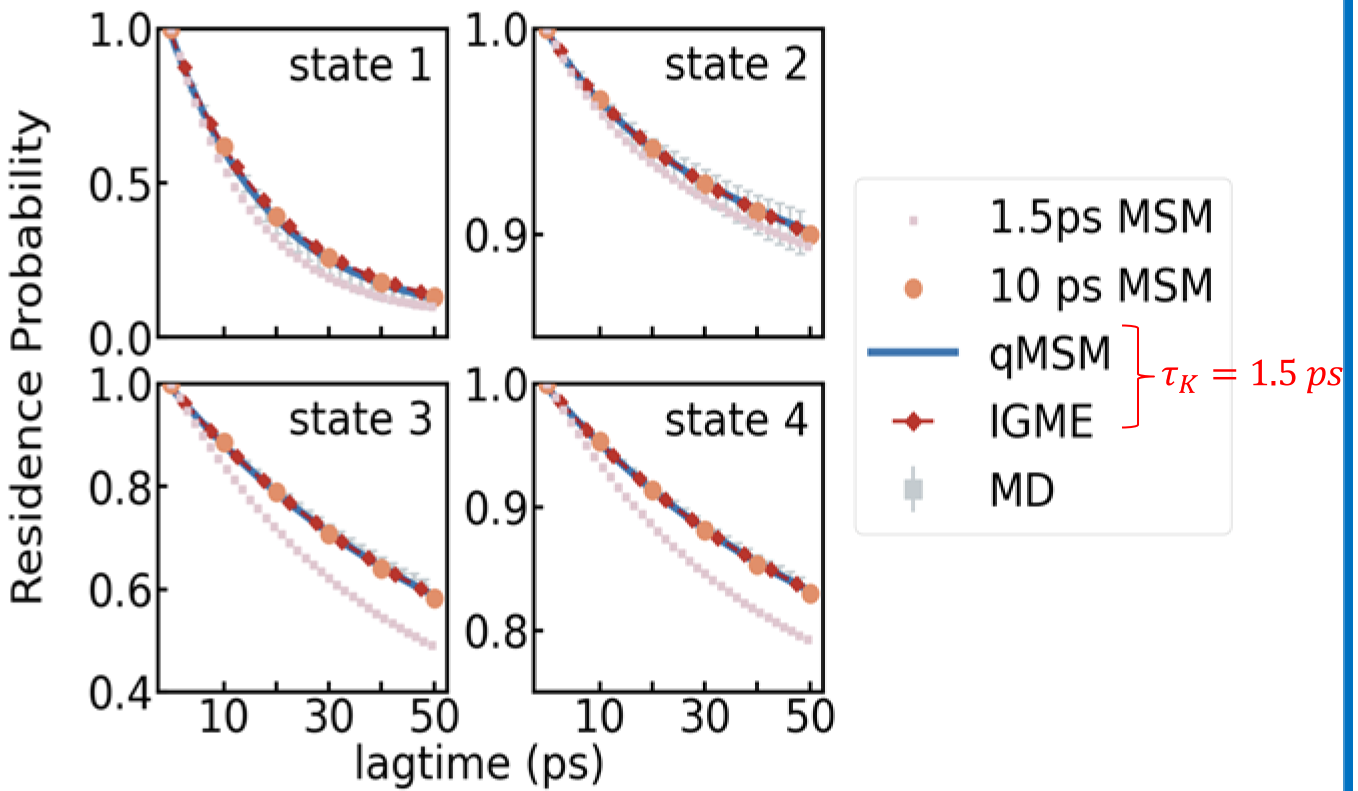
Generalized Master Equation (GME) Method

❖ **Alanine Dipeptide Tutorial**

Mean Integrated K (kernel)



Comparison b/w different approaches



Code Review

- **GME Tutorials - Python 3.11.9**
  - Repository: [GME_tutorials on GitHub]
- **msmbuilder2022**
  - Repository: [msmbuilder2022 on GitHub]
- **Installation Command**:
  pip install git+https://github.com/msmbuilder/msmbuilder2022.git
- **Common Installation Issue**
  - Error: distutils.errors.DistutilsPlatformError: Microsoft Visual C++ 14.0 or greater is required
  - Solution:
    1. Refer to [this guide].
    2. Download [Visual C++ Build Tools].
    3. Select only Desktop development with C++ during installation.
    4. Restart your computer after installation.

- **Issues with Python 3.12**
  - Fastcluster Compatibility
    - On Windows with Python 3.12, msmbuilder2022 installation may trigger a fastcluster error.
  - Numpy Compatibility
    - Python 3.12 includes a numpy version higher than 1.23.5, causing errors in igme.top_outputs in villin_headpiece_tutorial.ipynb.
    - Solution:
      pip install numpy==1.23.5