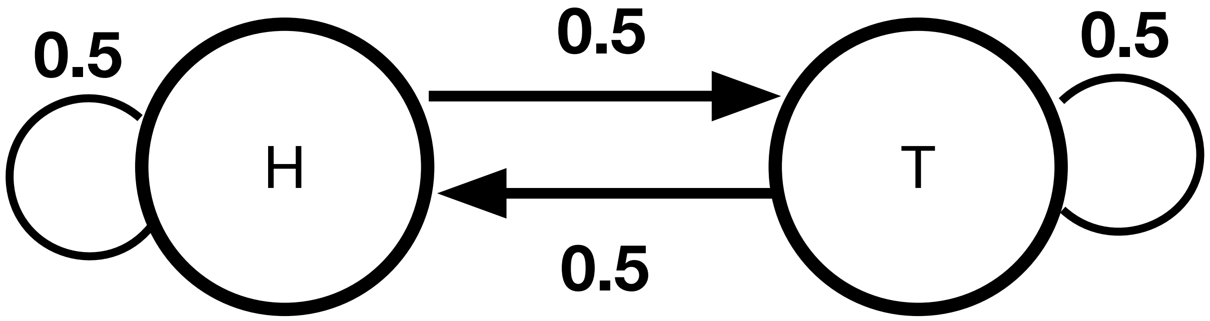
Introduction to Markov State Models

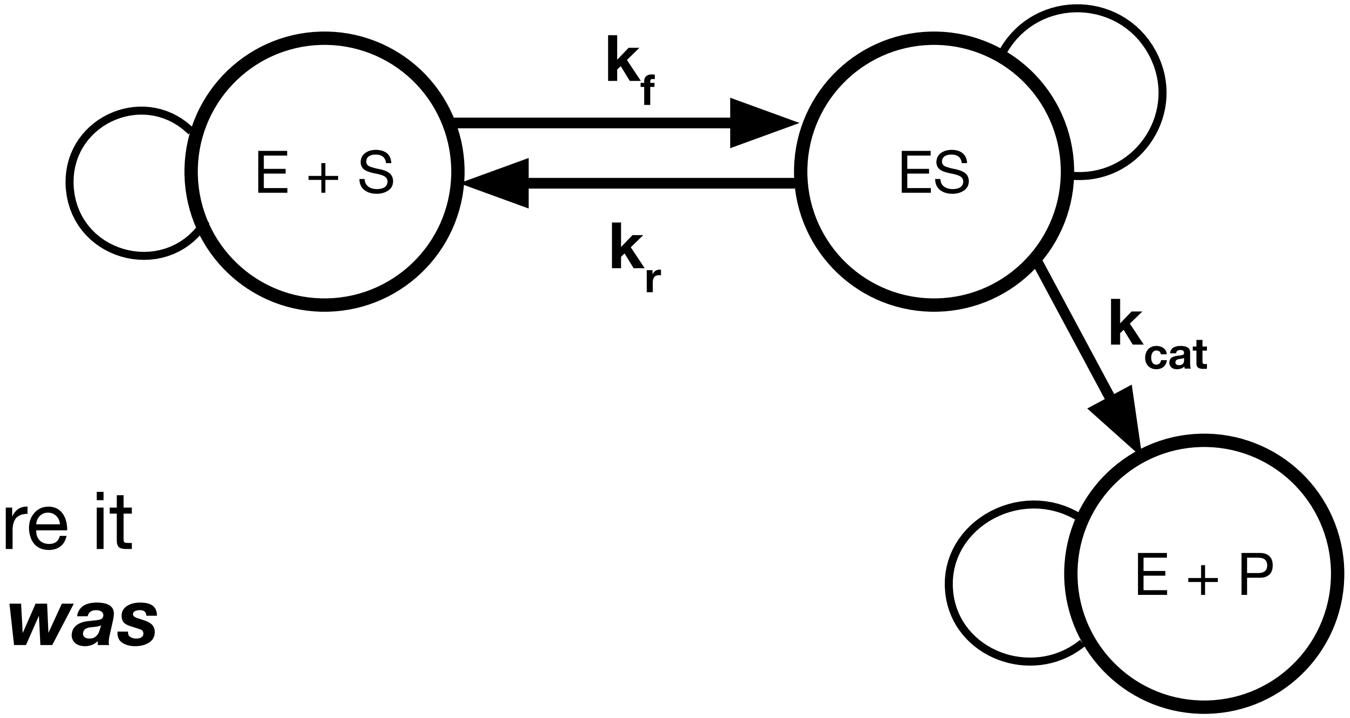
Drug Computing — Mobley Lab — UCI

Markov processes describe the dynamics of systems that have no “long term memory”

Flipping a coin



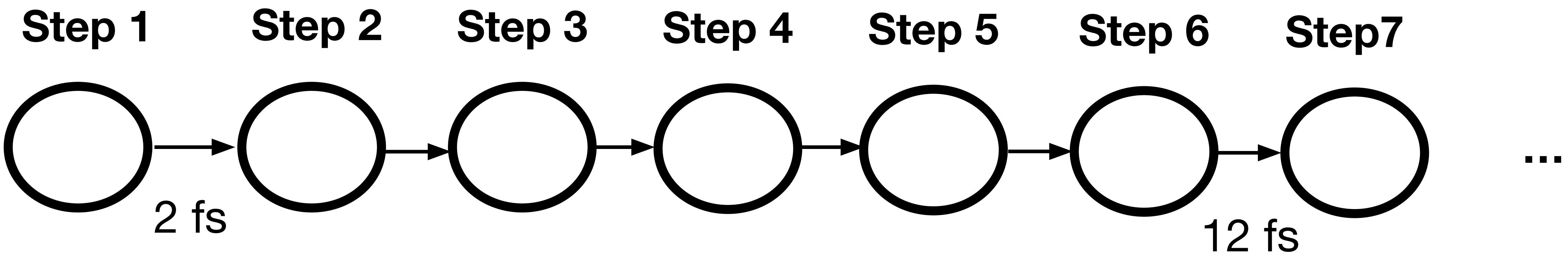
Michaelis-Mentin Kinetics



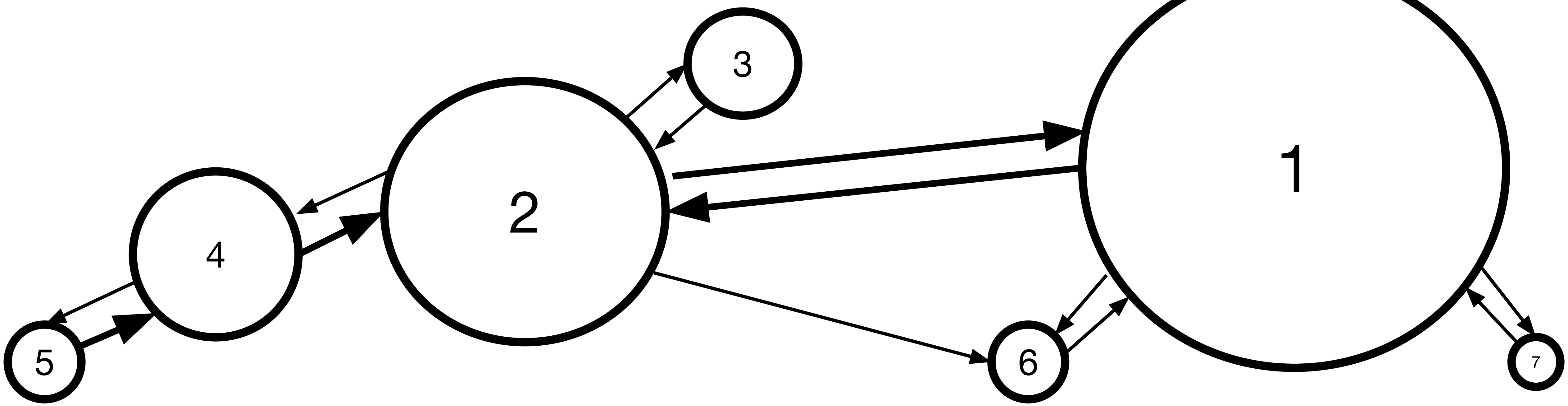
Where the system goes next is completely determined by where it ***currently*** is, and ***not where it was in the past***

Markov State Models (MSMs) reveal kinetics of configurational transitions

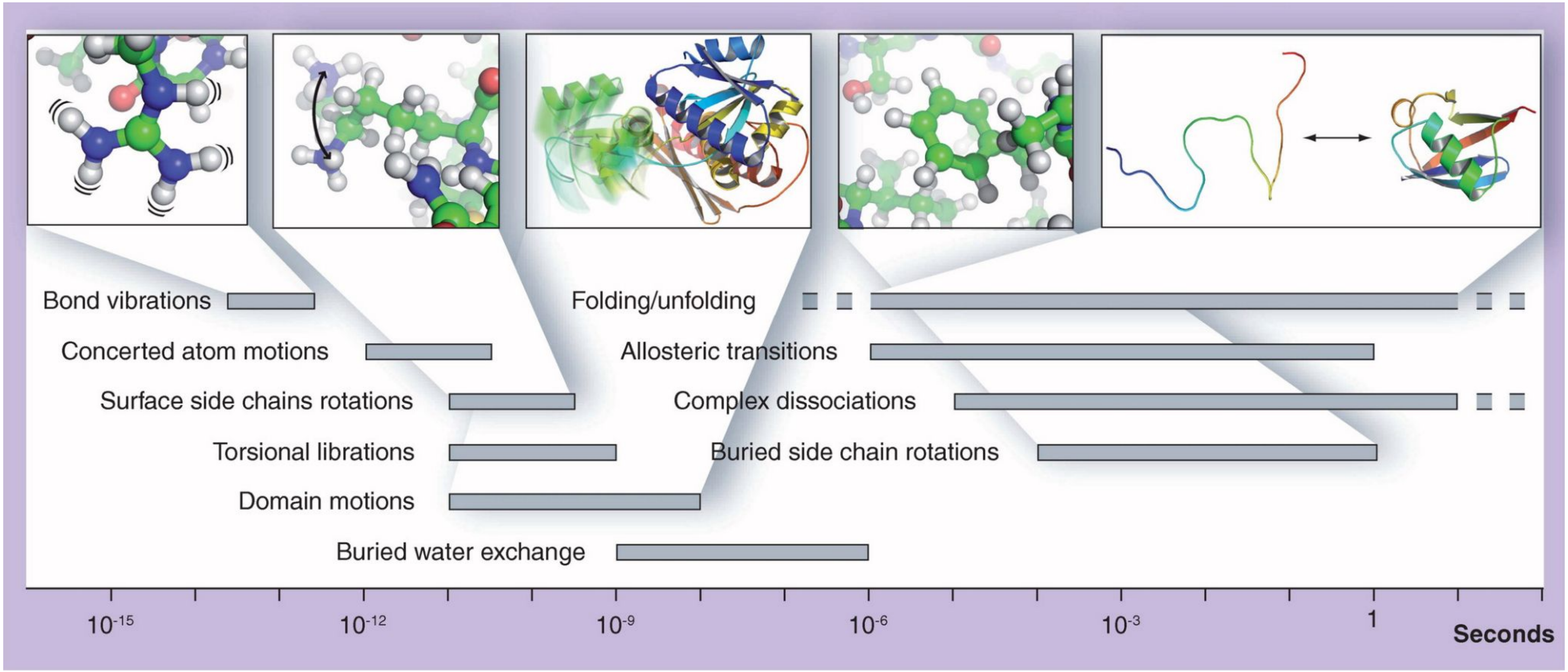
Highly Disordered Trajectory



Markov State Model (MSM)



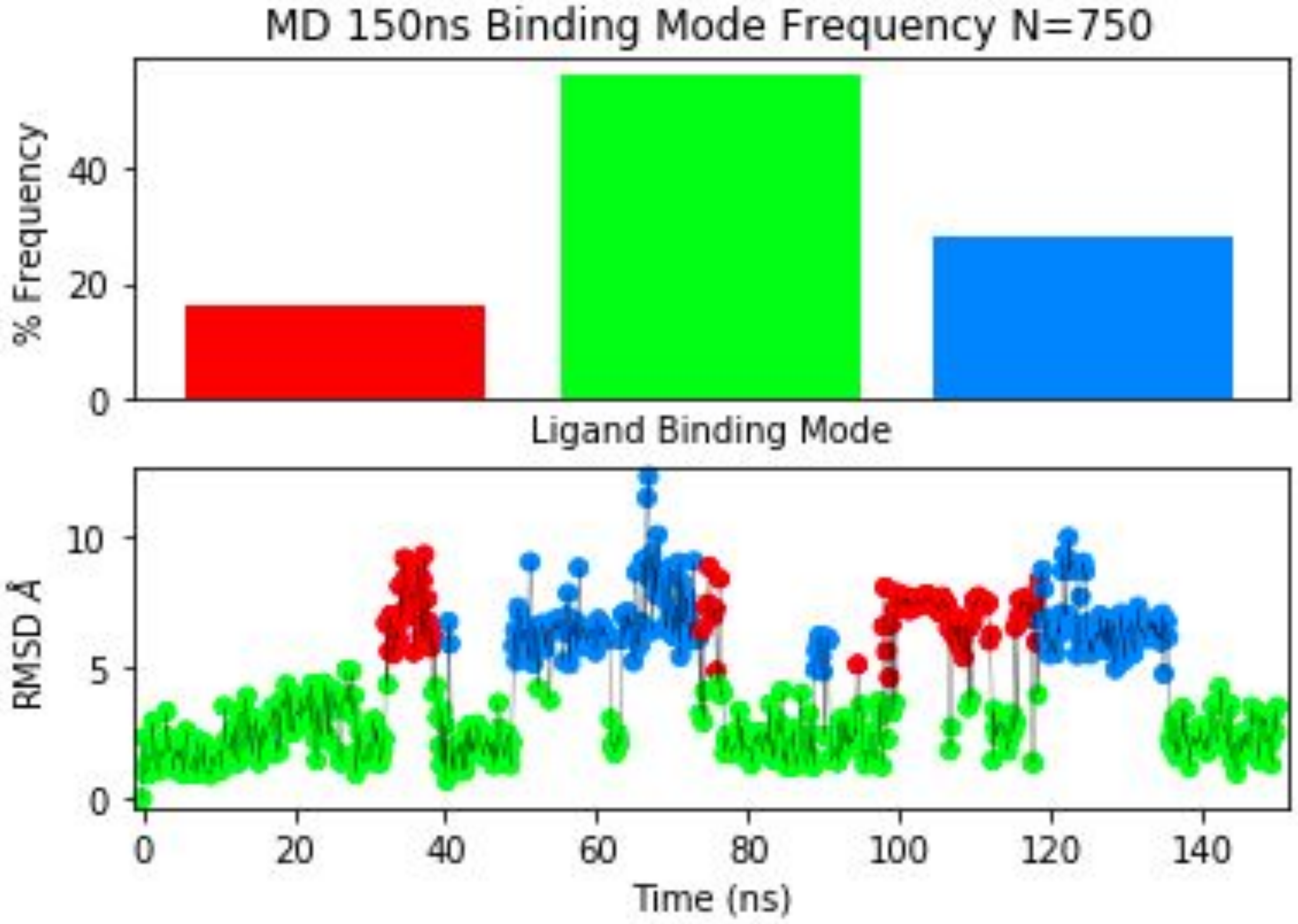
MSMs can be used to capture the macrostates of protein dynamics

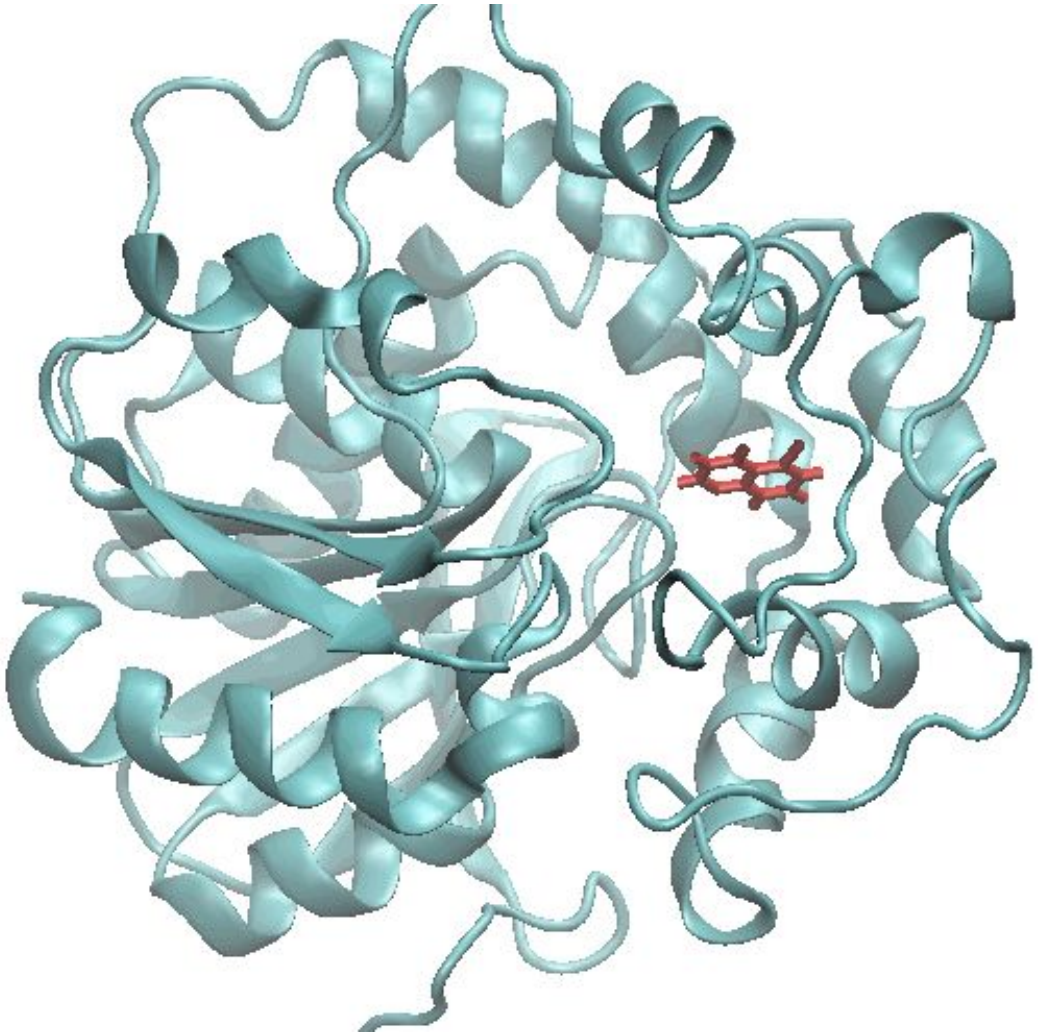


fs ns ms

Future medicinal chemistry 3.16 (2011): 2079-2100.

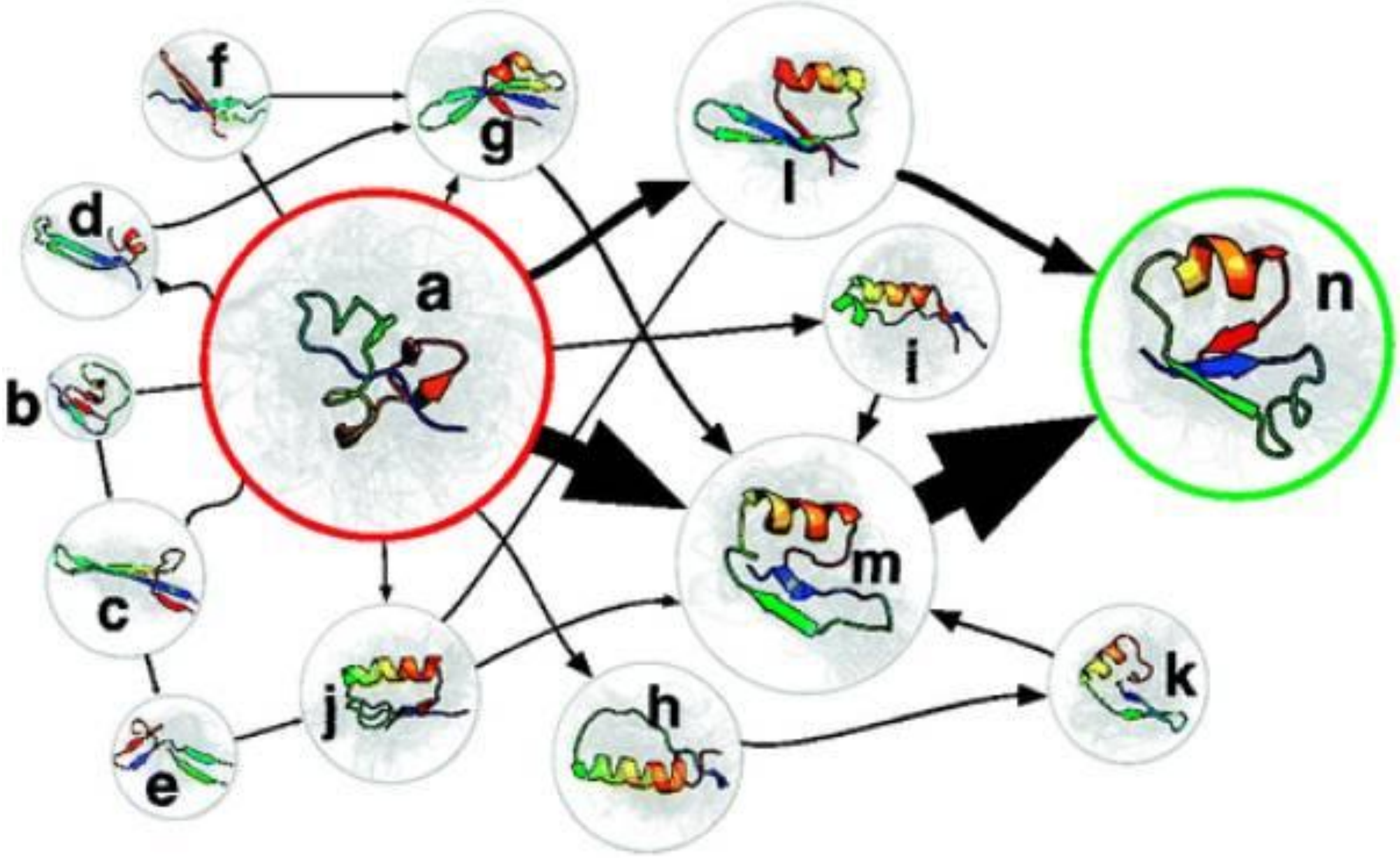
Markov State Models (MSMs) reveal kinetics of configurational transitions





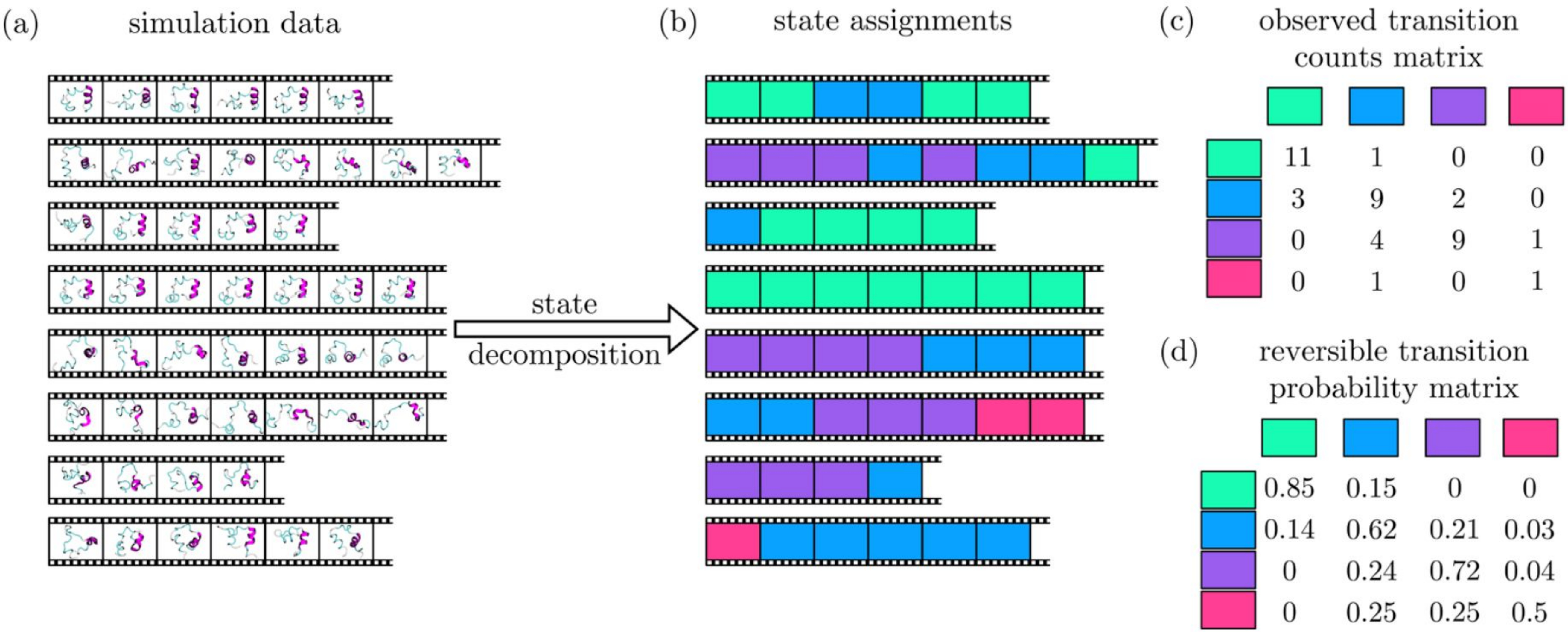
Building a Markov State Model

Useful for combining many different simulations into a single model
Example: building a (long timescale) kinetic protein folding pathway



JACS 132.5 (2010): 1526-8

MSMs can be built from many short simulations



(a) simulation data
state
decomposition
(b) state assignments
(c) observed transition counts matrix
(d) reversible transition probability matrix

🟩🟩🟦🟦🟩🟩

🟪🟪🟪🟦🟪🟦🟦🟩

🟦🟩🟩🟩🟩

🟩🟩🟩🟩🟩🟩🟩

🟪🟪🟪🟪🟦🟦🟦

🟦🟦🟪🟪🟪🟥🟥

🟪🟪🟪🟦

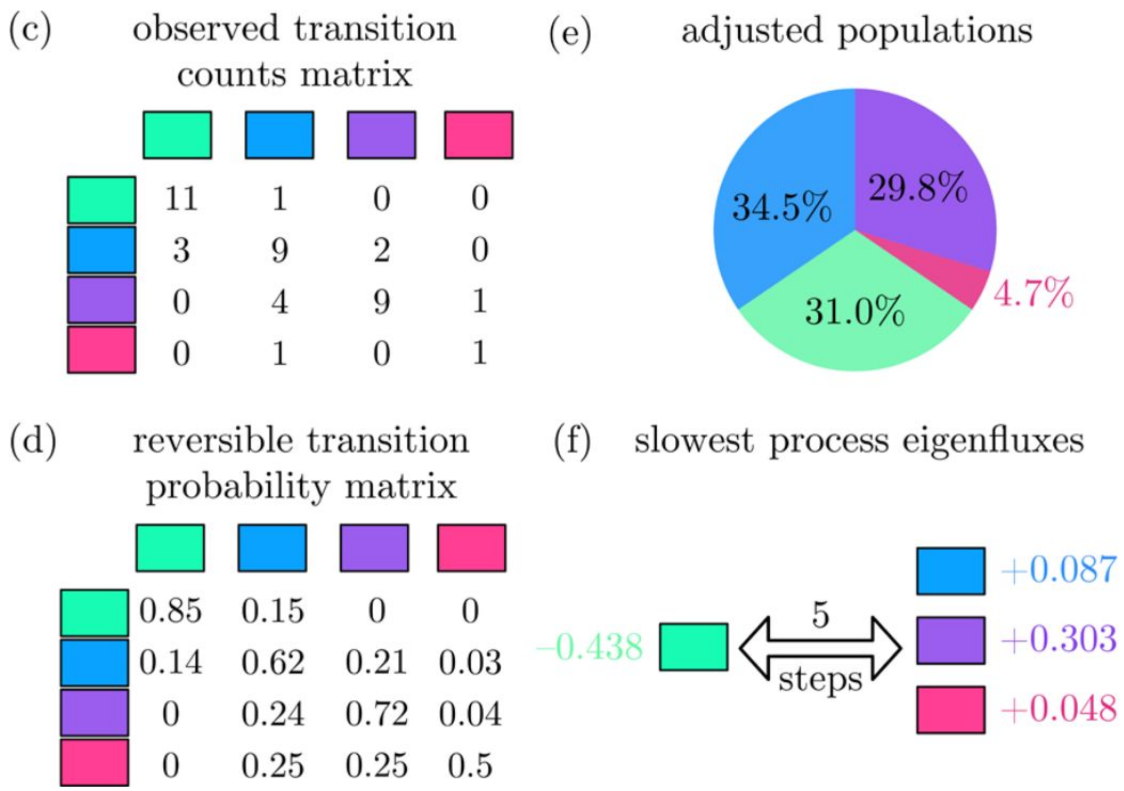
🟥🟦🟦🟦🟦🟦

	🟩	🟦	🟪	🟥
🟩	11	1	0	0
🟦	3	9	2	0
🟪	0	4	9	1
🟥	0	1	0	1

	🟩	🟦	🟪	🟥
🟩	0.85	0.15	0	0
🟦	0.14	0.62	0.21	0.03
🟪	0	0.24	0.72	0.04
🟥	0	0.25	0.25	0.5

JACS 140.7 (2018): 2386-2396.

MSMs can describe the equilibrium process of the model



(c) observed transition counts matrix

	🟩	🟦	🟪	🟥
🟩	11	1	0	0
🟦	3	9	2	0
🟪	0	4	9	1
🟥	0	1	0	1

(e) adjusted populations

(d) reversible transition probability matrix

	🟩	🟦	🟪	🟥
🟩	0.85	0.15	0	0
🟦	0.14	0.62	0.21	0.03
🟪	0	0.24	0.72	0.04
🟥	0	0.25	0.25	0.5

(f) slowest process eigenfluxes

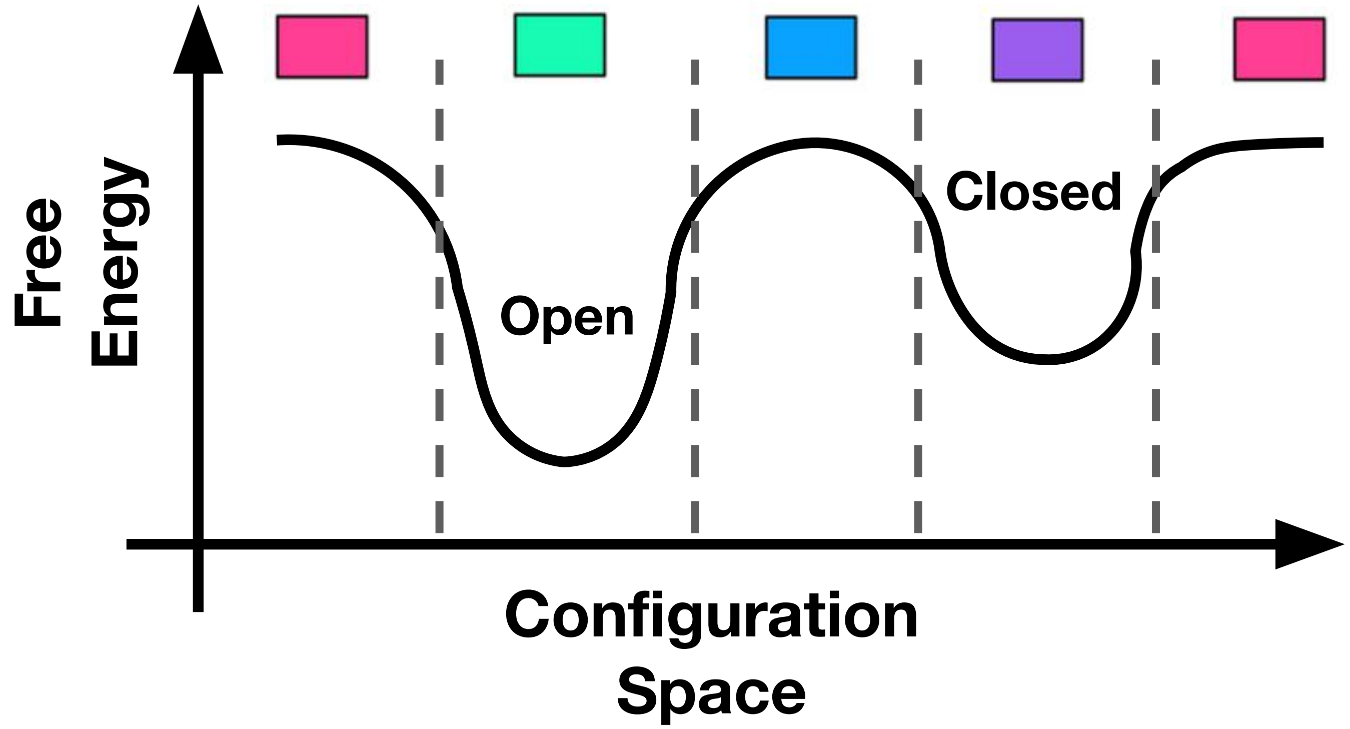
First eigenvector of the probability matrix describes the equilibrium populations

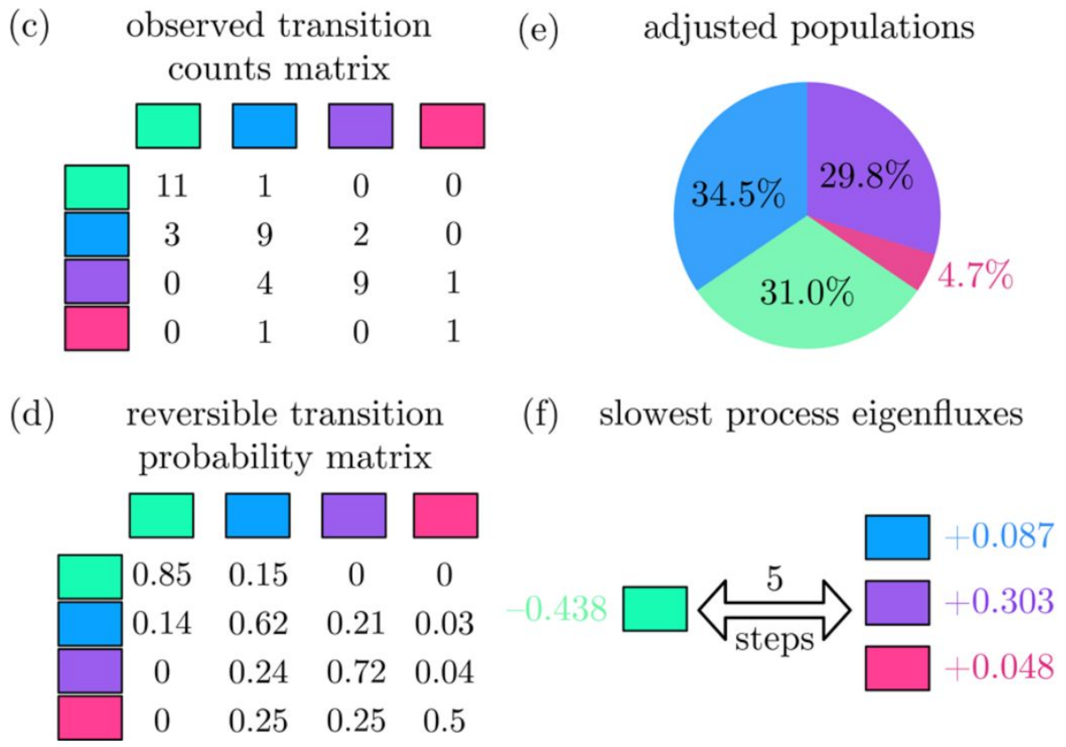
$$\pi^T = \pi^T \mathbf{T}(\tau)$$

Other eigenvectors (processes) show fluxes between states

JACS 140.7 (2018): 2386-2396.

Building a Markov State Model




(c) observed transition counts matrix

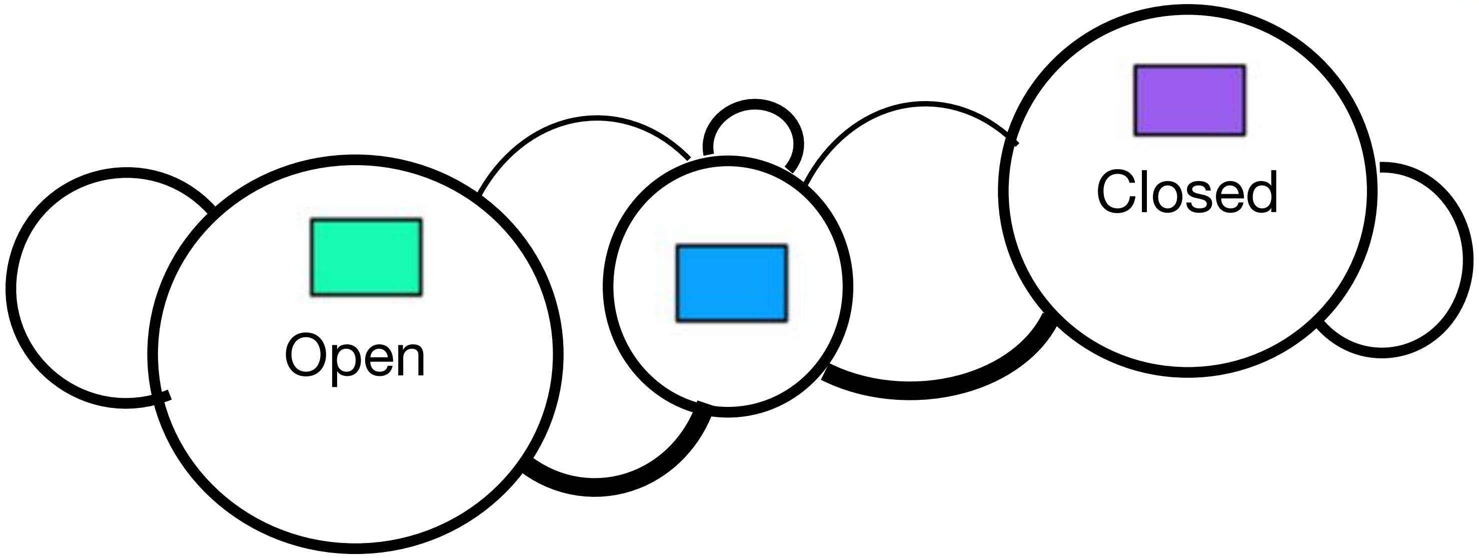
	🟩	🟦	🟪	🟥
🟩	11	1	0	0
🟦	3	9	2	0
🟪	0	4	9	1
🟥	0	1	0	1

(d) reversible transition probability matrix

	🟩	🟦	🟪	🟥
🟩	0.85	0.15	0	0
🟦	0.14	0.62	0.21	0.03
🟪	0	0.24	0.72	0.04
🟥	0	0.25	0.25	0.5

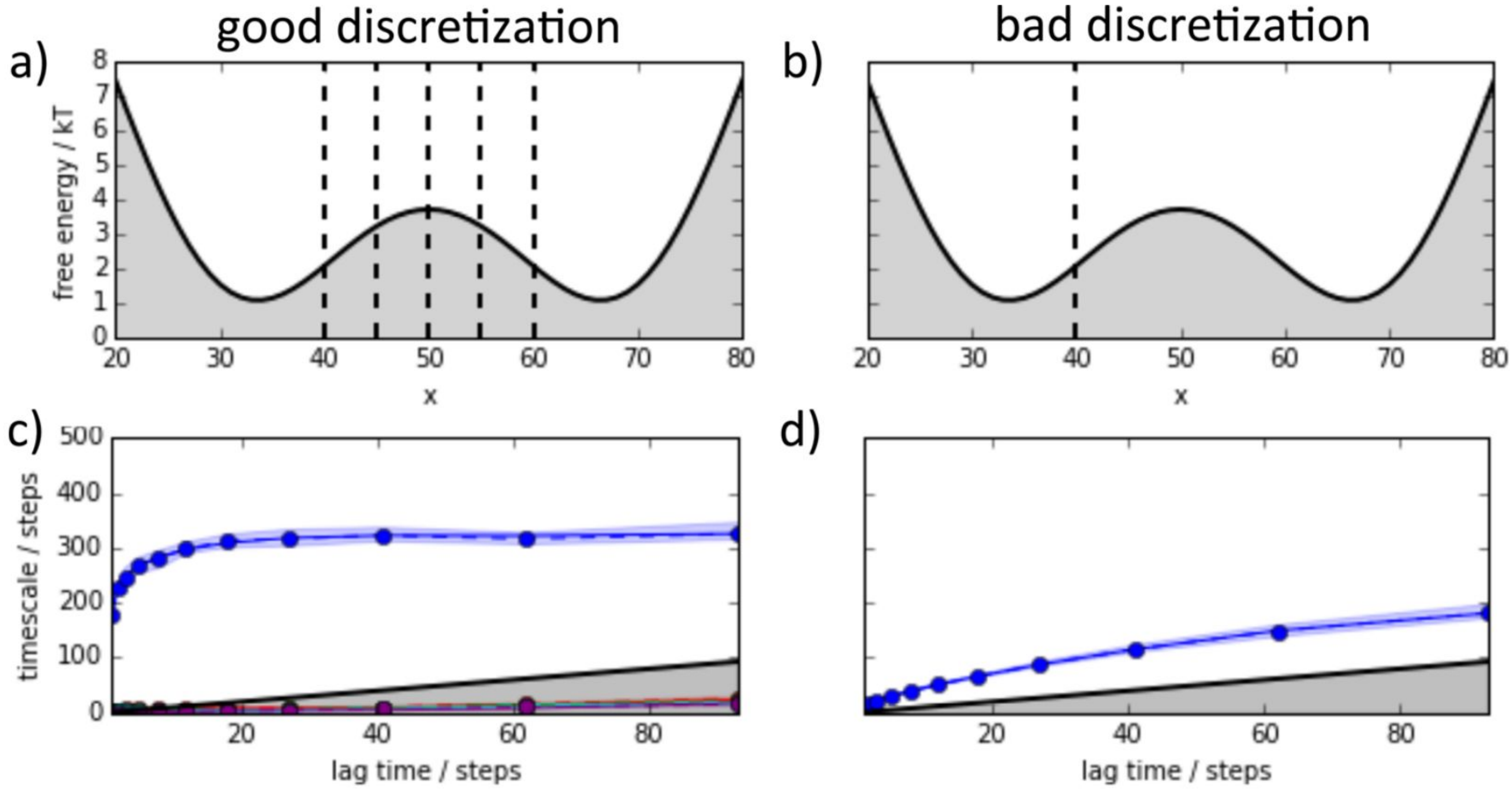
(e) adjusted populations

(f) slowest process eigenfluxes



$$F_i = -k_B T \ln \frac{\pi_i}{\max_j \pi_j}$$

Good clustering is necessary for maintaining Markovian properties



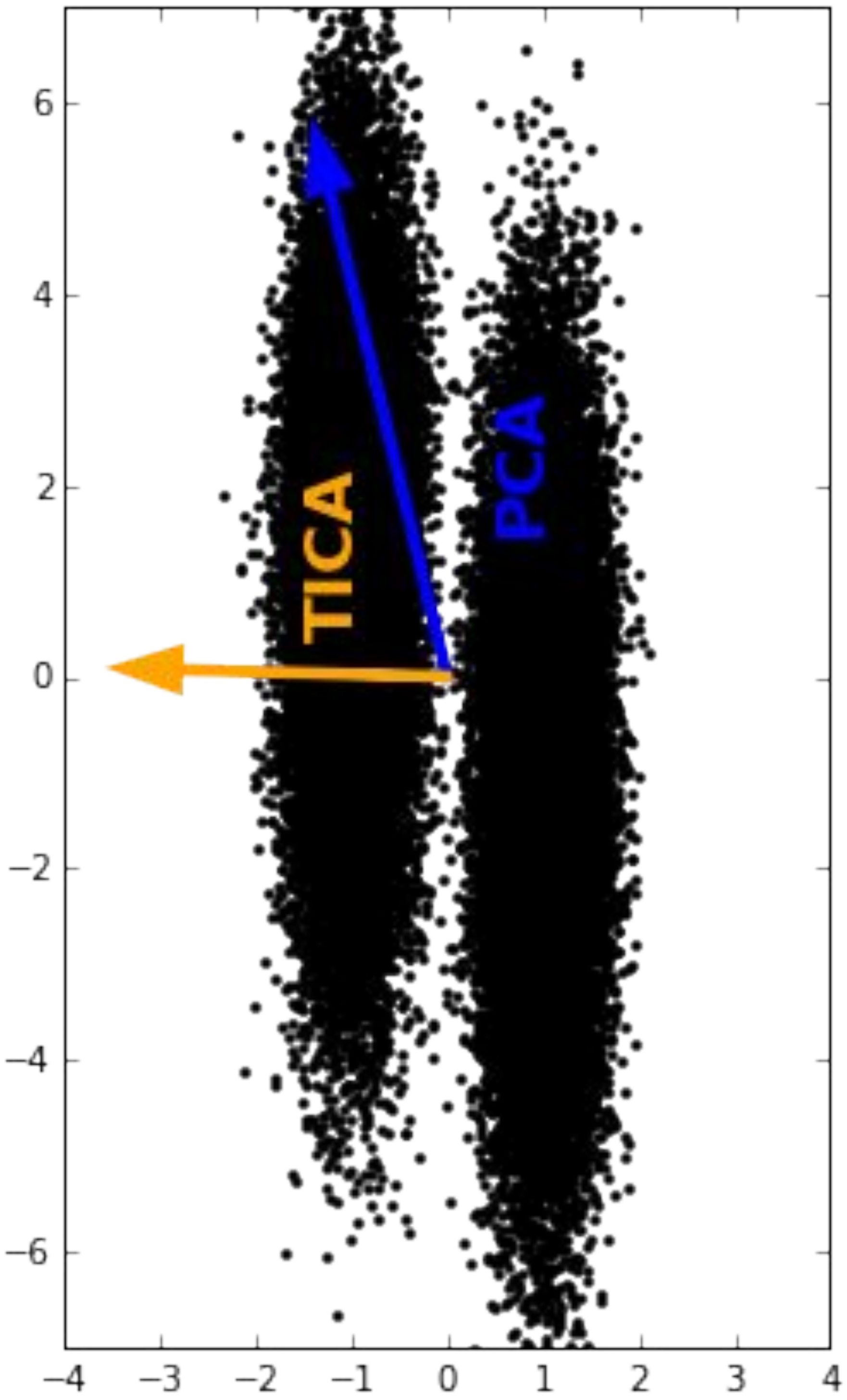
τ

JCTC 11.11 (2015): 5525-5542.

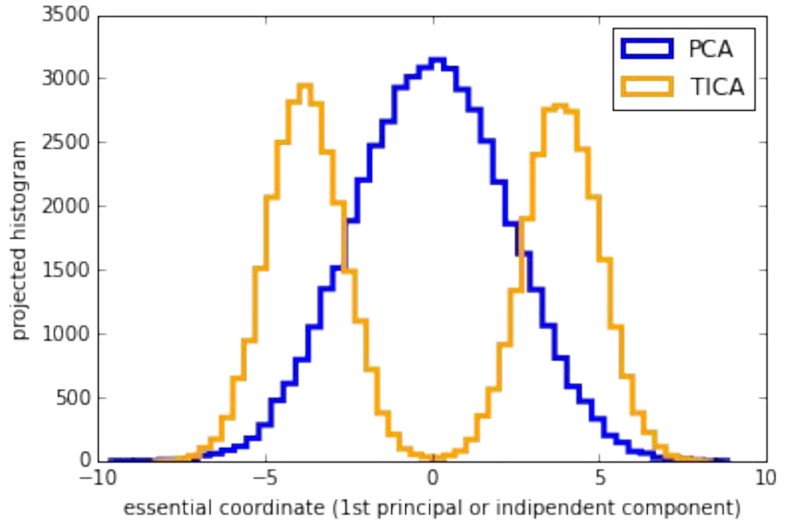
Dimensionality Reduction: Principle Component Analysis (PCA) and Time-independent Component Analysis (TICA)

PCA finds the direction of maximum variance in the data

TICA identifies the slowest reaction coordinates (maximum autocorrelation)

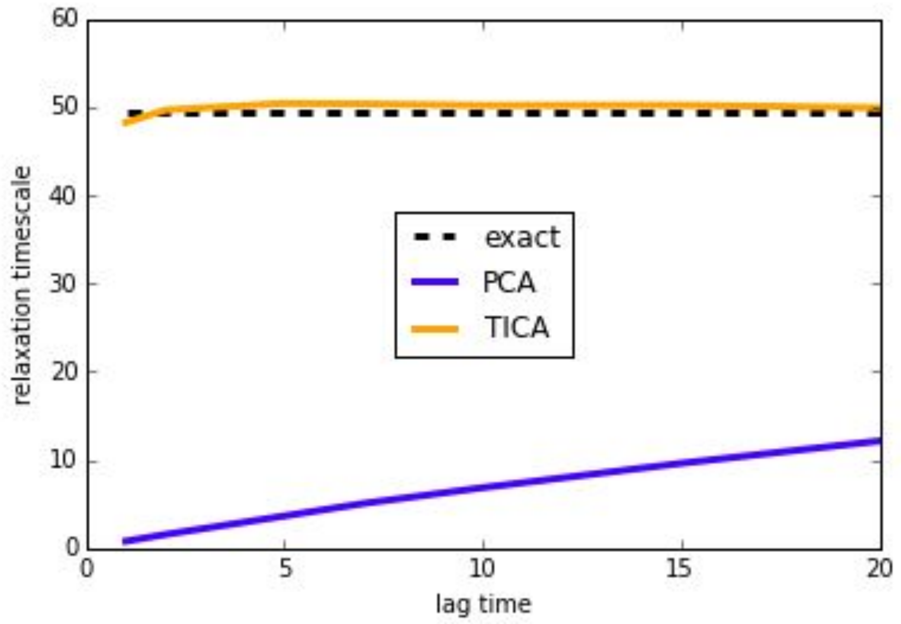


Projection:



http://docs.markovmodel.org/lecture_tica.html

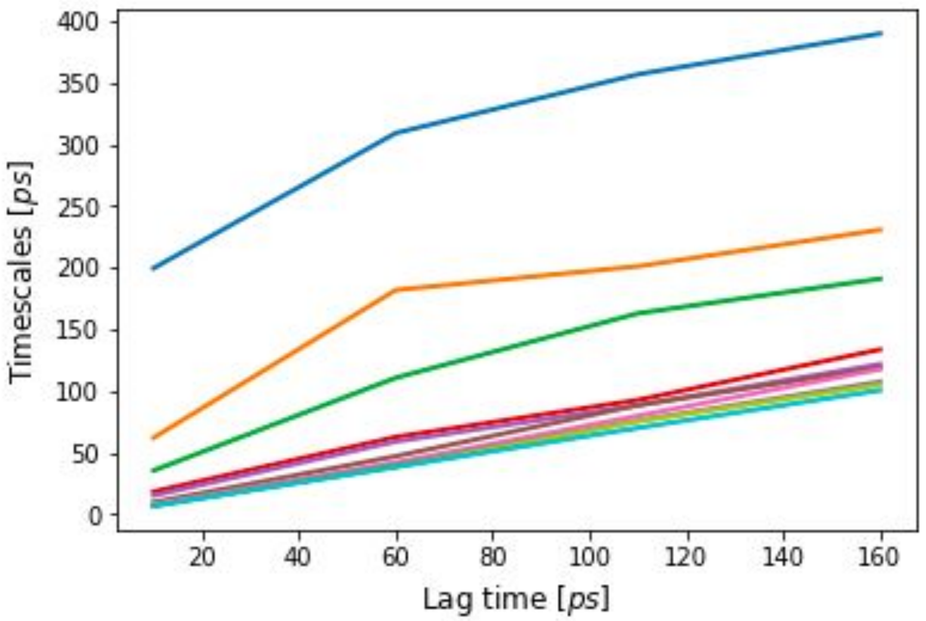
Testing the dimensionality reduction: Implied Timescales



Perfectly Markovian Process

$$t_i(k\tau_0) = t_i = -\frac{\tau_0}{\ln \lambda_i(\tau_0)}$$

http://docs.markovmodel.org/lecture_implied_timescales.html



Imperfect Markov Model

$$t_i(k\tau_0) = -\frac{k\tau_0}{\ln \lambda_i(k\tau_0)}$$

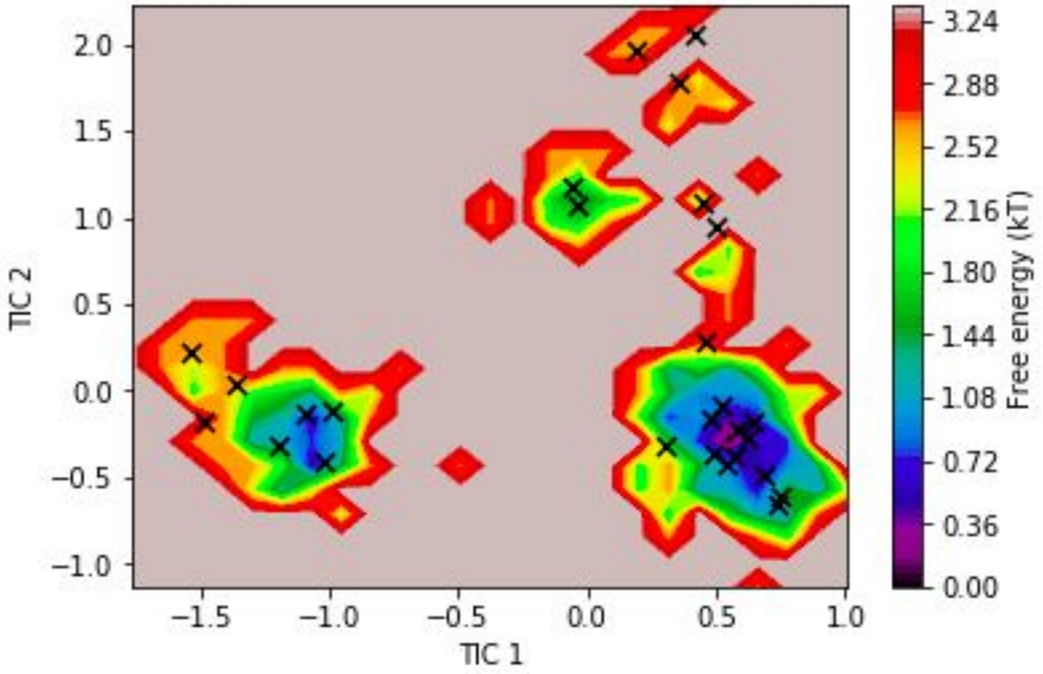
Choose a time scale at which the model appears to start behaving Markovian

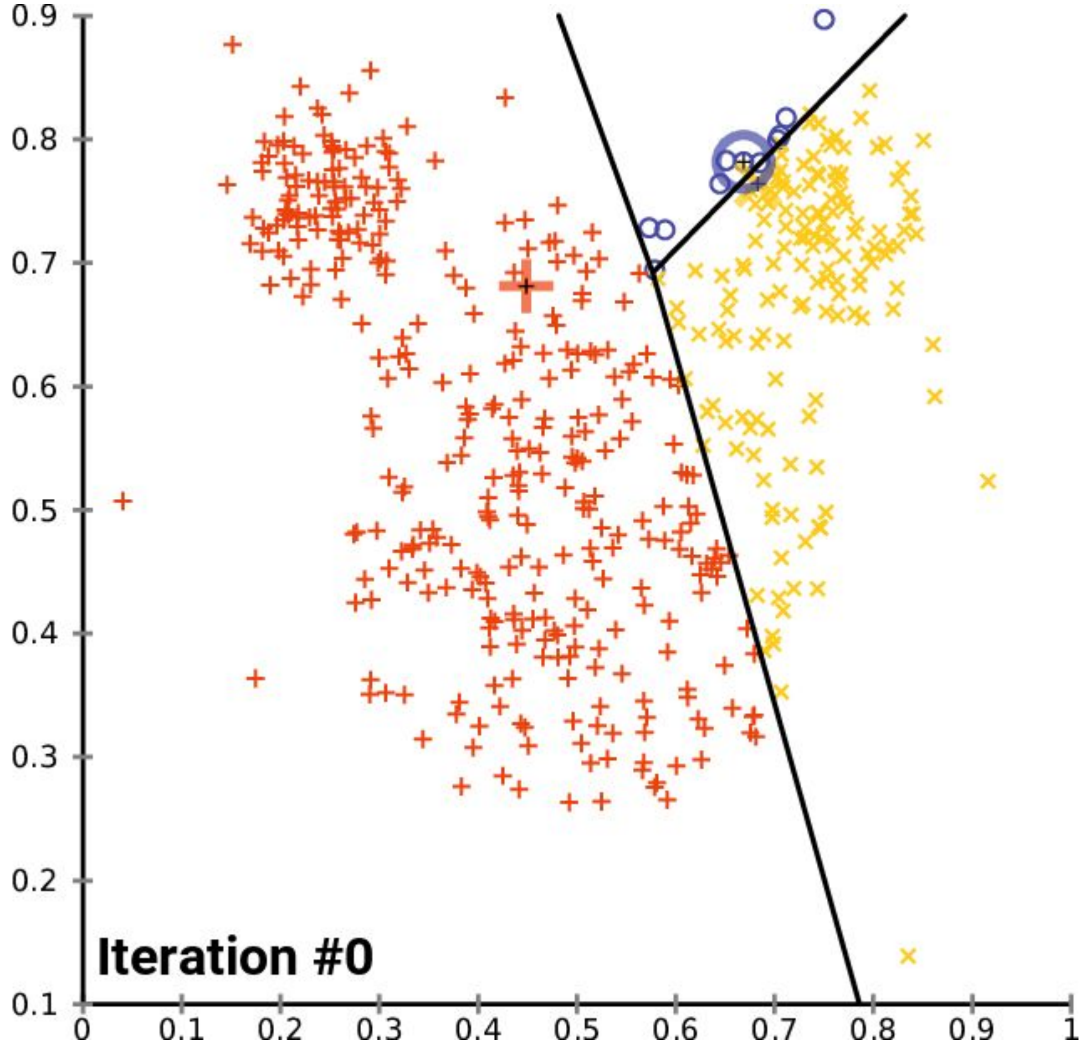
Using the dimension reduction to identify macrostates of the system

PCCA and K-means Clustering

K-means Clustering:

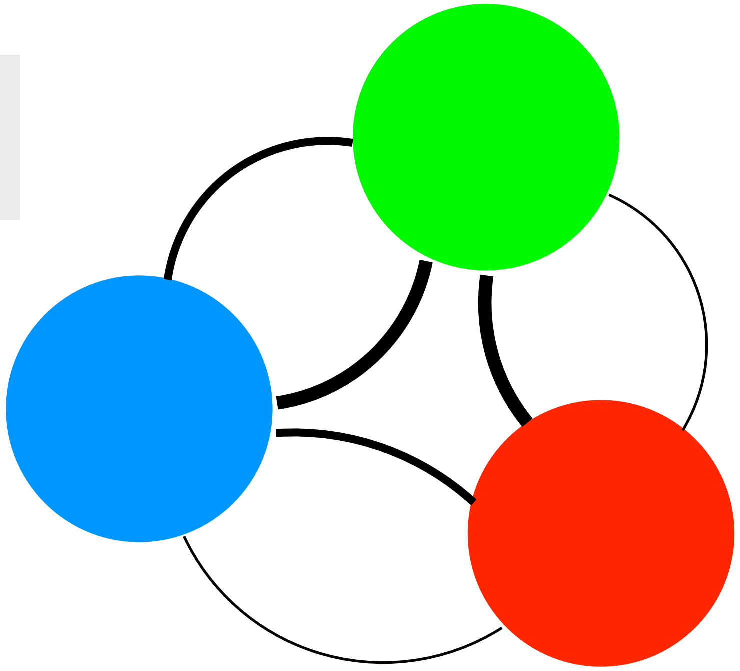
- Identifies clusters based on distance in TICA space





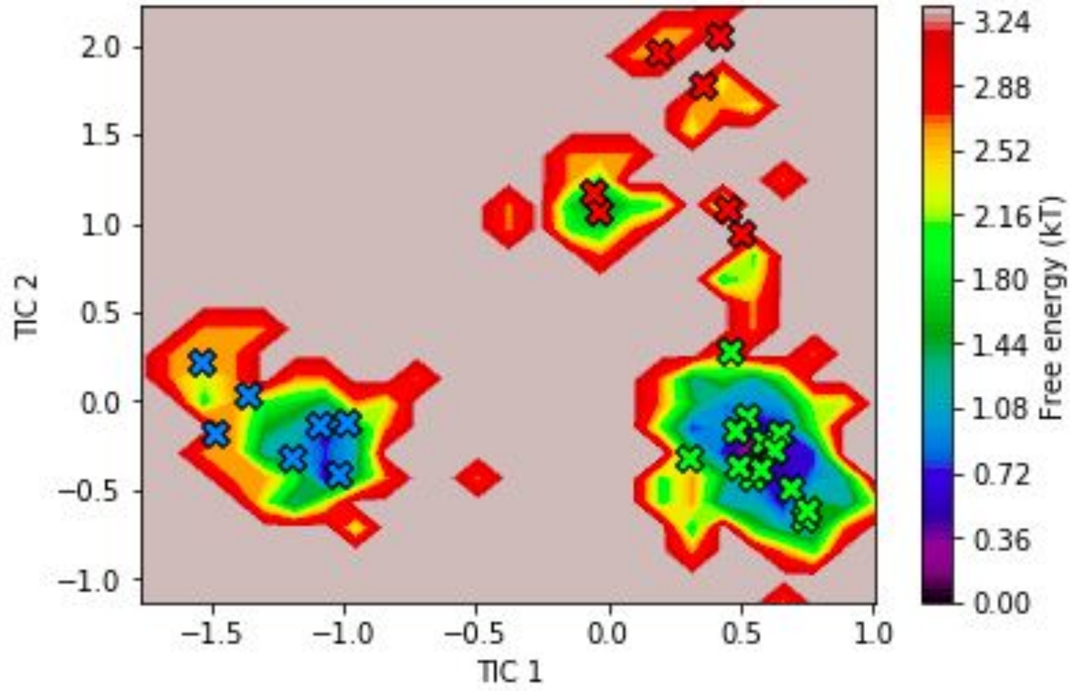
Using the dimension reduction to identify macrostates of the system

PCCA and K-means Clustering



PCCA: Perron-Cluster Cluster Analysis

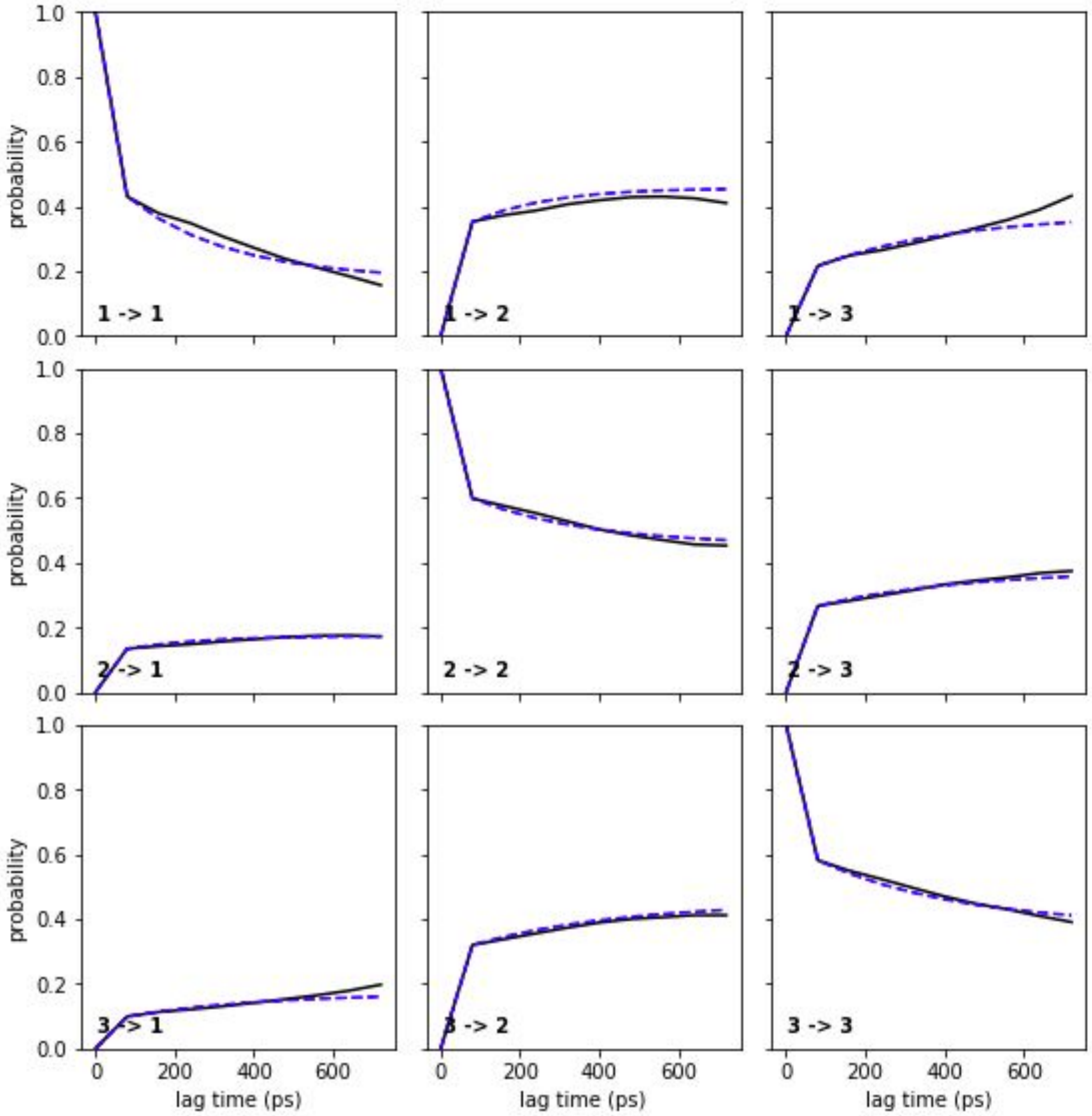
- Uses the eigenvector-eigenvalue decomposition to identify meta-stable states
- Splits the available conformational state space in order of longest relaxation time



Testing the Markov Model: Chapman-Kolmogorov test

How well do the interconversions between the coarse-grained macro-states of the model compare to the real trajectory?

$$[\hat{\mathbf{T}}(\tau)]^k \approx \hat{\mathbf{T}}(k\tau)$$



Core concepts and procedure to use Markov models

(1) Identify Configuration Microstates

- Dimensionality Reduction
    - ○ TICA, PCA
- Clustering
    - ○ K-means

(2) Construct Transition Matrix Between States

(3) Test for Self-Consistency

(4) Calculate Kinetics + Free Energy Between State