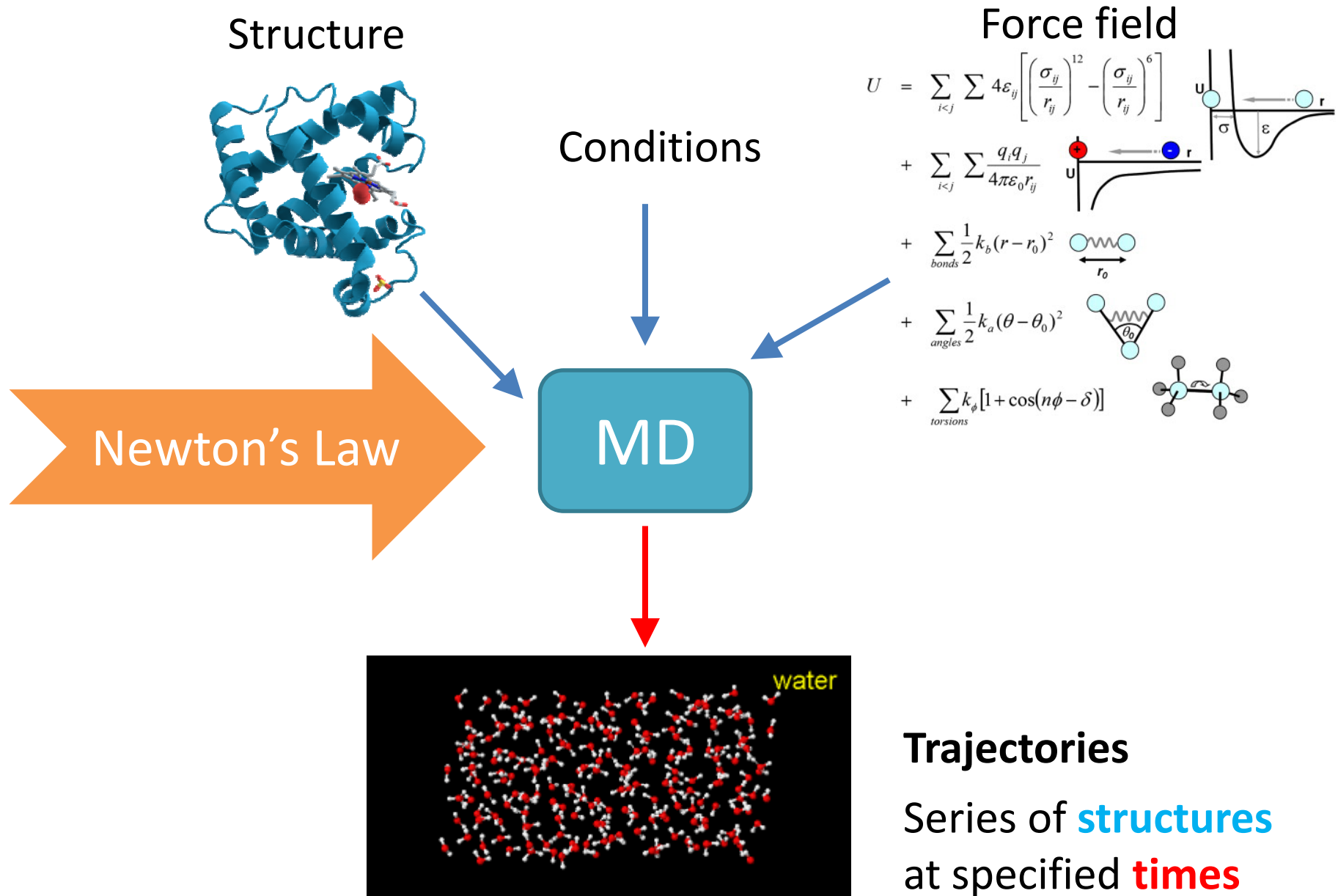
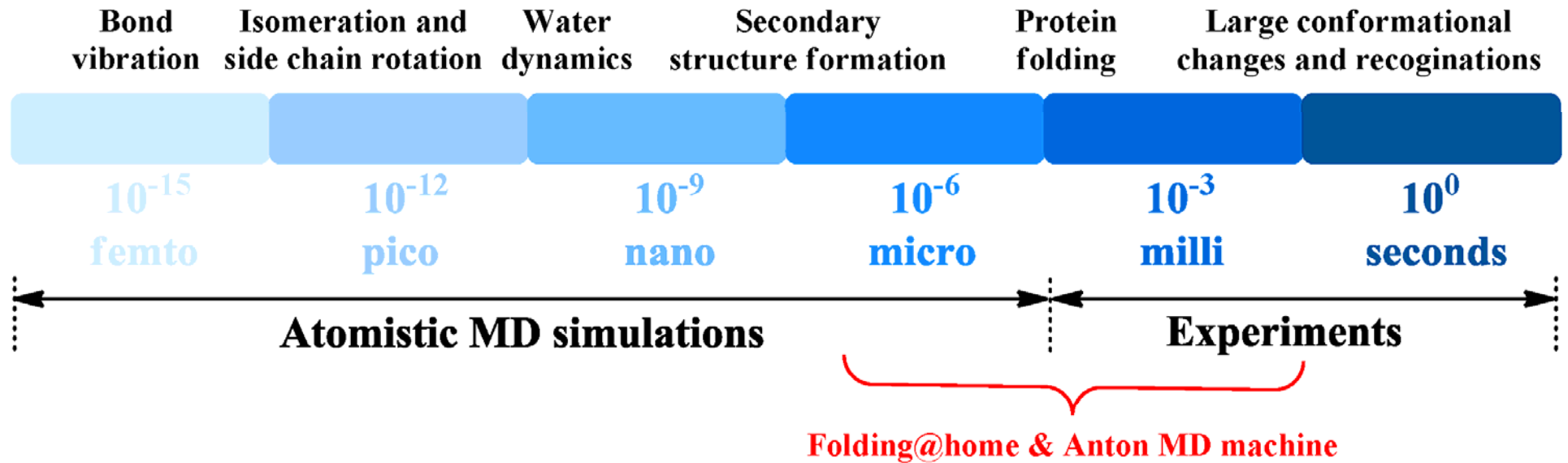


Introduction to Markov State Model (MSM): Part I

Molecular Dynamics Simulation

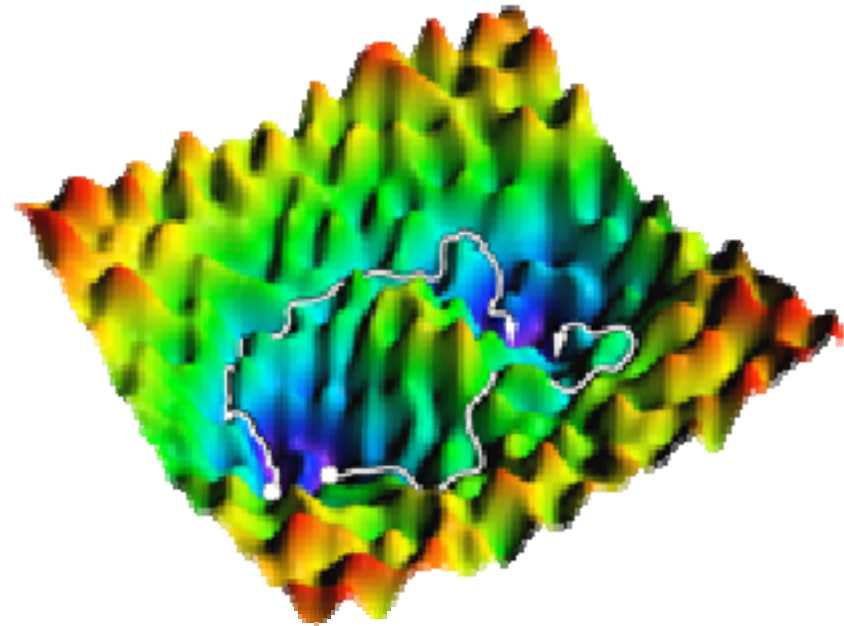
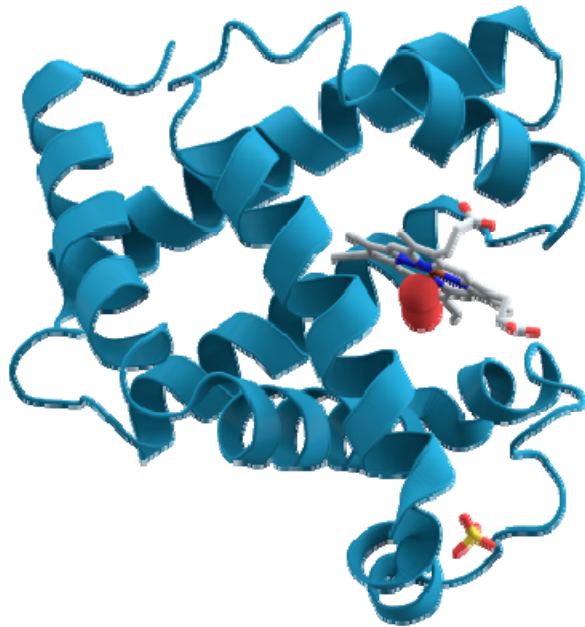


Timescale gap



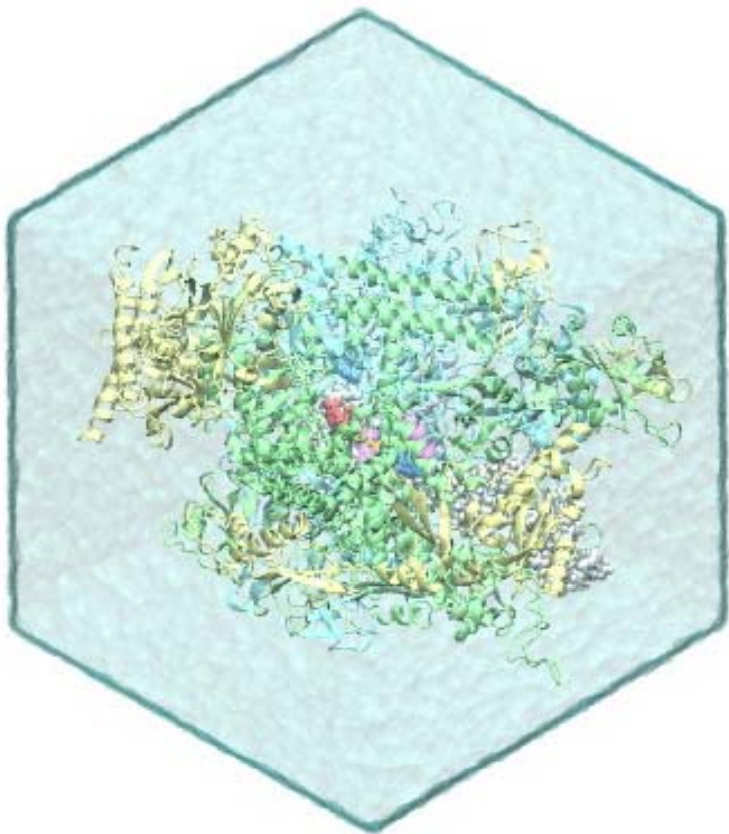
Markov State Model (MSMs): a kinetic network model can enhance sampling and bridge the gap between experiments and simulations.

Energy Landscape of High-dimensional System

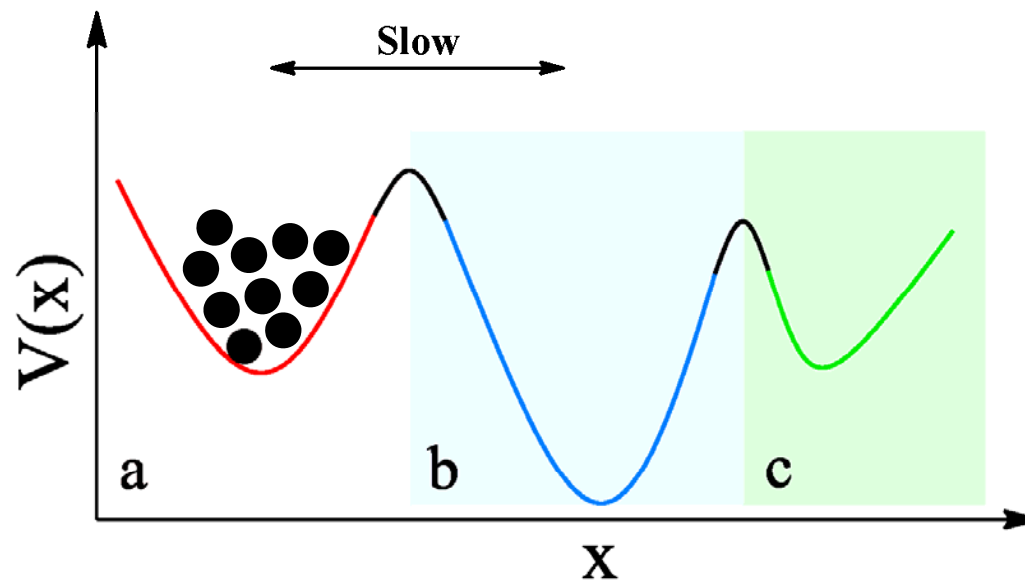


Rugged energy surface in high-dimensional system

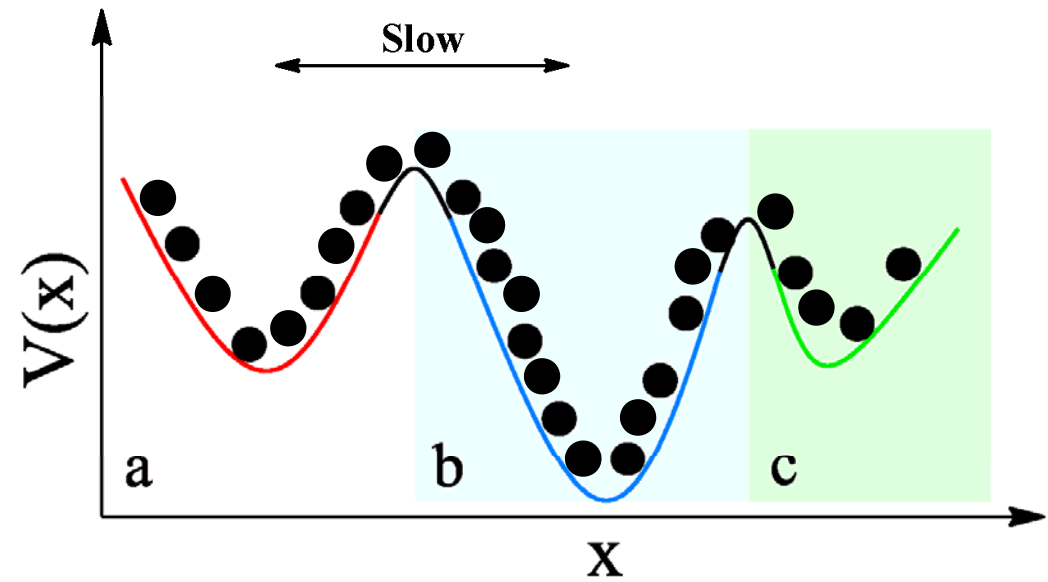
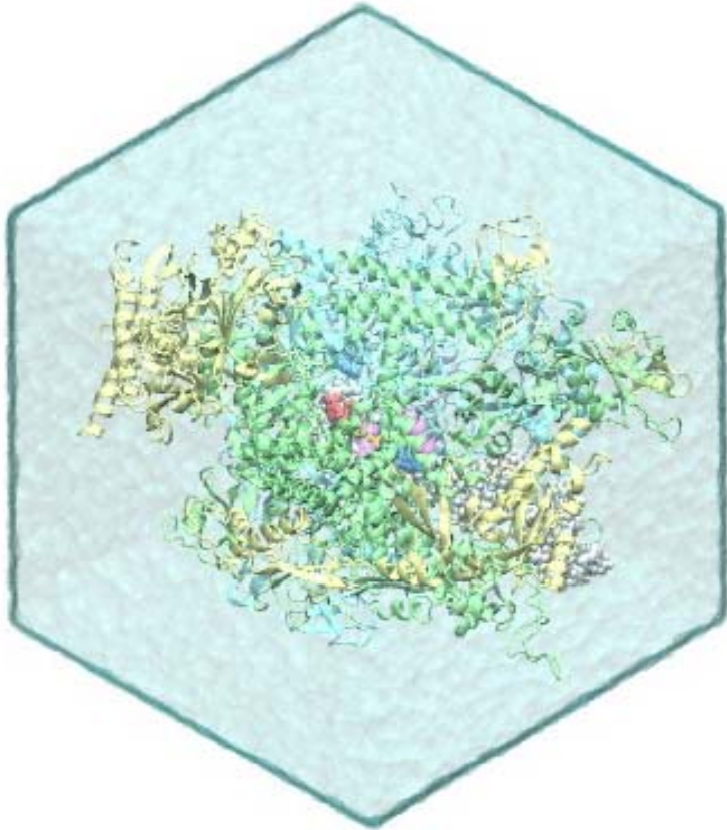
Sampling issue



~370,000 atoms



Sampling issue

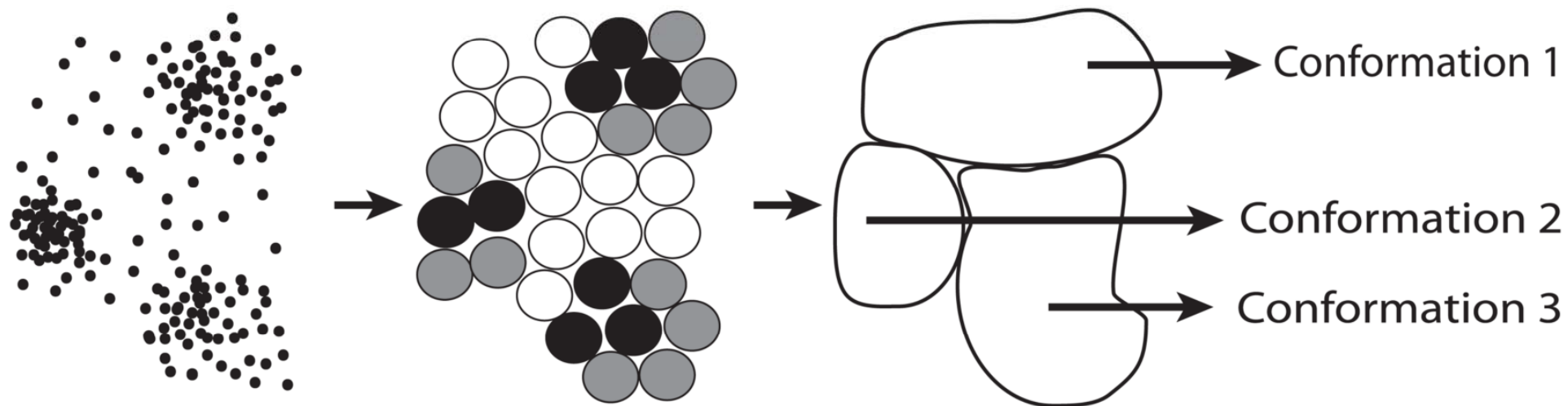


Available Tools

- Enhanced Sampling Techniques
 - Steered MD
 - Targeted MD
 - Accelerated MD
 - Replica Exchange MD (REMD)
 - Metadynamics
- Mathematical Model
 - Markov State Model (MSM)



Building the Markov State Models (MSMs)



Raw data
(conformations)


Microstates
(geometry clustering)


Macrostates
(metastable states)

**Local equilibrium
within discrete state**

Building the Markov State Models (MSMs)

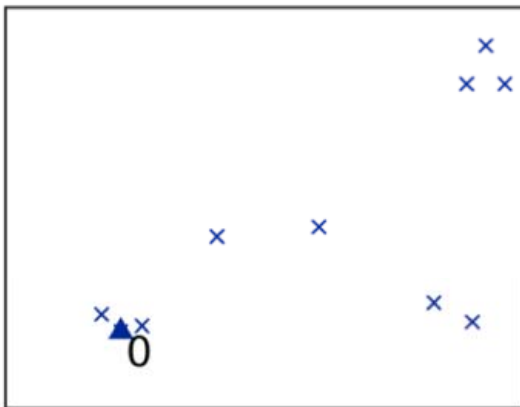
- 
- Geometric clustering for microstates
 - (Assumes similar kinetic behavior within state)

- 
- Drawing the implied timescale plot
 - (Determines the lag time at which the system can be approximated as a Markovian model)
 - (Choose the number of macrostate)

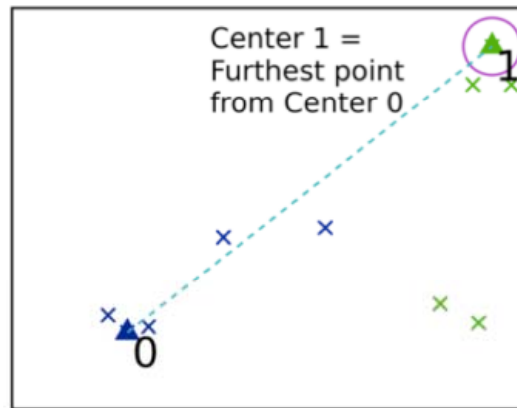
- 
- Lumping the microstates into macrostates
 - (Maintaining similar kinetic behaviour within state)

Geometric clustering for microstates

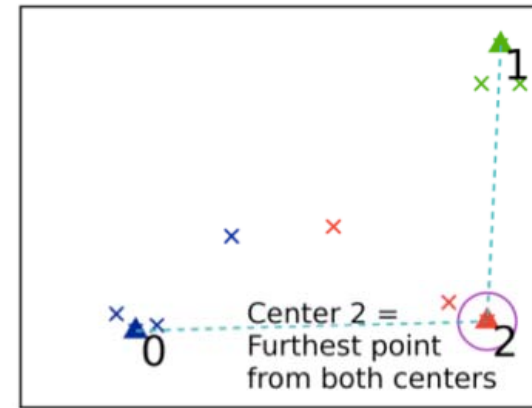
k-centers



(a)

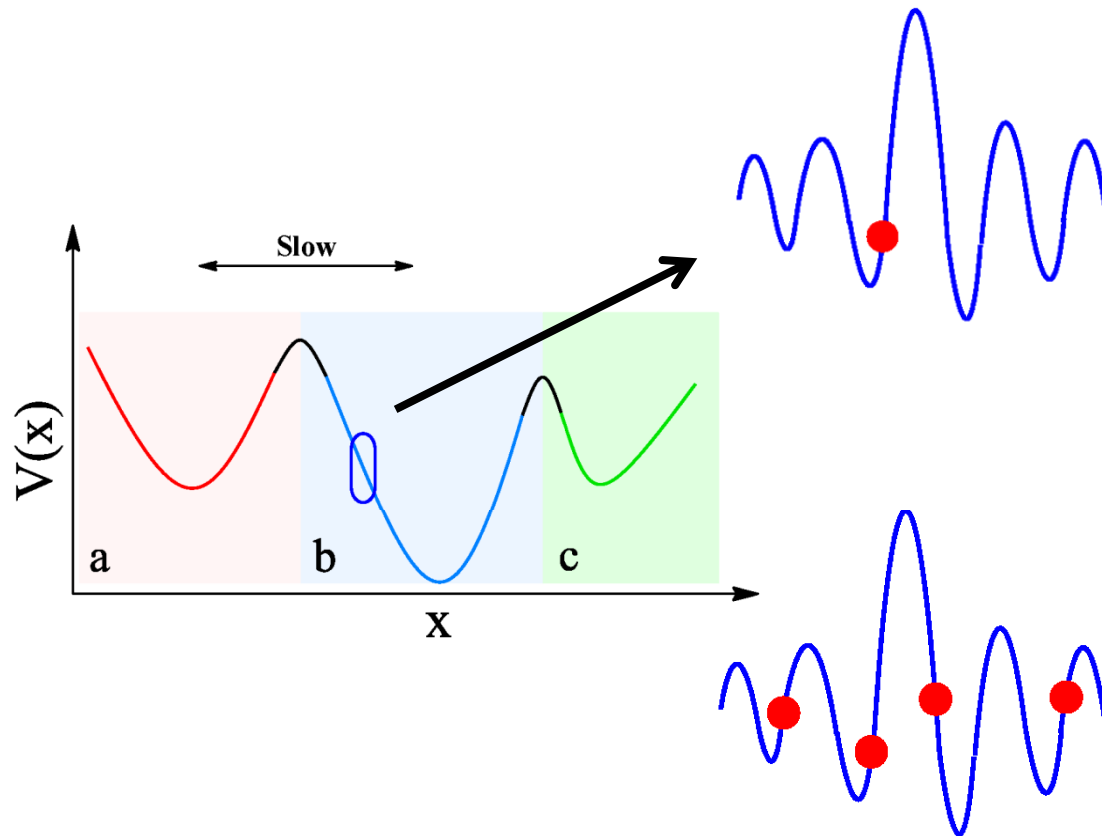


(b)



(c)

MSM Can Describe Conformational dynamics by coarse-grained time and space



Coarse-graining of space into discrete states will introduce memory.

Coarse-graining of time can make the memory appear short if there is separation of timescales.

Transition Probability Matrix

- Central to the MSM
- Constructed from data obtained from MD
- The technique of its construction greatly affects the resulting model

Building Transition Count Matrix

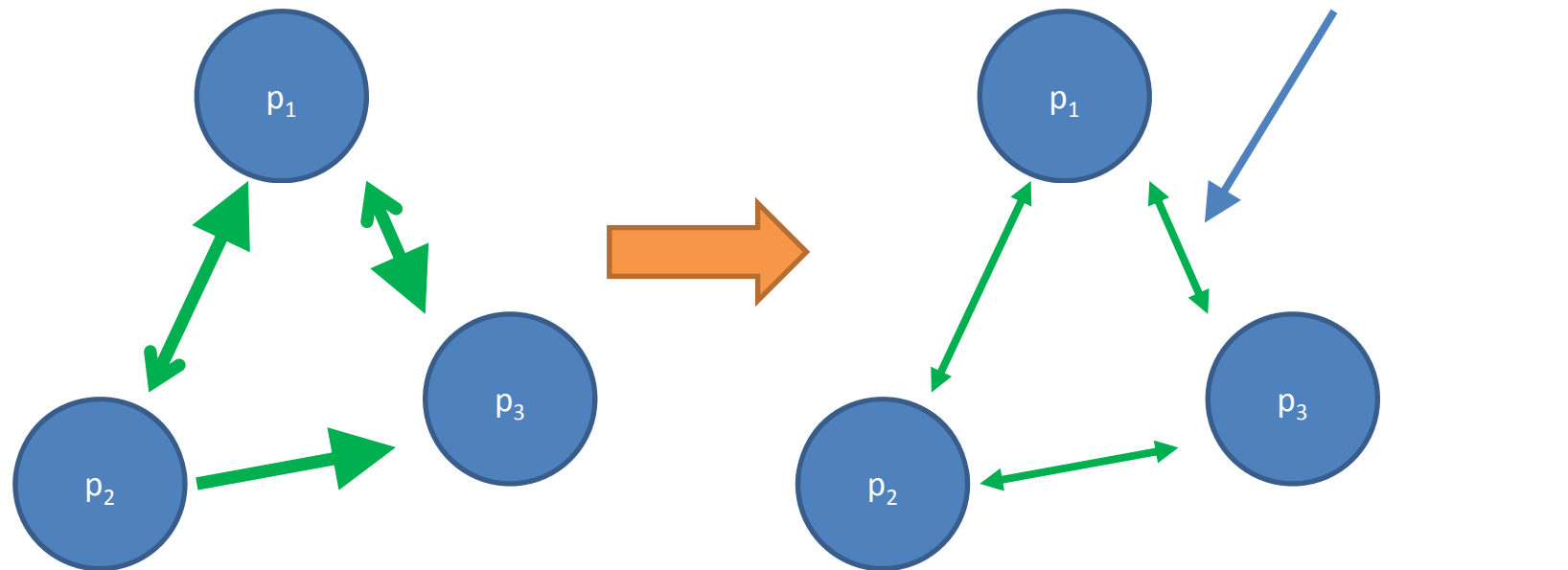
For trajectory: [1,1,2,2,2,1,2,1,2,1,2]
if the lag time is 1:

$$N_{11} = 1, N_{12} = 4, N_{21} = 3, N_{22} = 2$$

From \ To	State 1	State 2
State 1	1	4
State 2	3	2

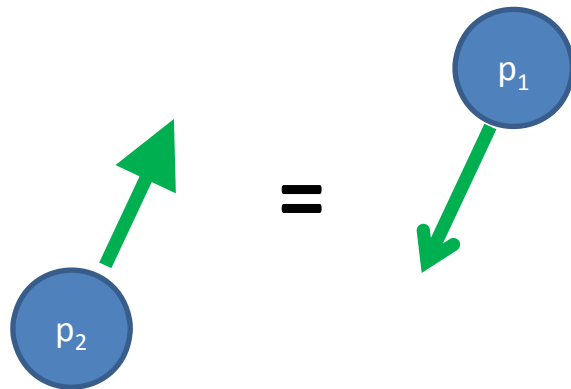
Detailed Balance

- At equilibrium, all the elementary transitions should also be at equilibrium (Detailed Balance)



Detailed Balance

- Detailed balance implies having a transition from state i to j should be equally likely as having a transition from state j to i
- Because the actual number of transition is determined by the product of population and transition probability, we have:



$$\pi_i P_{ij} = \pi_j P_{ji},$$

$$N_{ij} = N_{ji}$$

Fulfilling Detailed Balance

- Symmetrizing TCM by:

$$N^{symm} = \frac{N + N^T}{2}$$

- Generate TPM by:

$$P_{ij} = \frac{N_{ij}^{symm}}{\sum_{ij} (N_{ij}^{symm})}$$

From \ To	State 1	State 2
State 1	1	4
State 2	3	2

From \ To	State 1	State 2
State 1	1	3.5
State 2	3.5	2

From \ To	State 1	State 2
State 1	0.222	0.778
State 2	0.636	0.364

Building the MSM

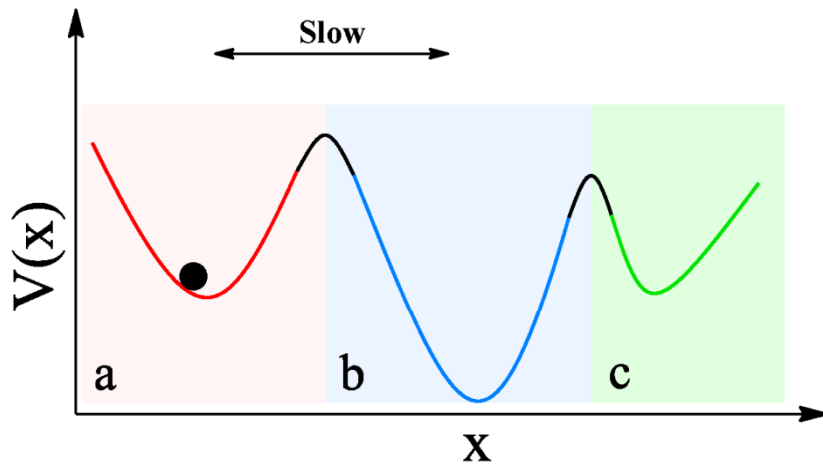
$$T_{ij}(\tau) = \frac{N_{ij}(\tau)}{\sum_j N_{ij}(\tau)} \longrightarrow \mathbf{T}(\tau) = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & & \\ \vdots & & \ddots & \\ P_{n1} & & & P_{nn} \end{bmatrix}$$

Transition Probability Matrix

$$P(n\tau) = [T(\tau)]^n P(0)$$

Where $P(n\tau)$ is a vector of state populations at time $n\tau$.

Simple example



From:

State:	A	B	C	To:
	0.5	0.2	0.3	A
	0.3	0.8	0.3	B
	0.2	0.0	0.4	C

$T(\tau) =$

$$P(0) = \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \end{bmatrix}$$

$$P(n\tau) = [T(\tau)]^n P(0)$$

$$P(1\tau) = T(\tau)P(0) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.8 & 0.3 \\ 0.2 & 0.0 & 0.4 \end{bmatrix} \begin{bmatrix} 1.0 \\ 0.0 \\ 0.0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix}$$

$$P(3\tau) = T(\tau)P(2\tau) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.8 & 0.3 \\ 0.2 & 0.0 & 0.4 \end{bmatrix} \begin{bmatrix} 0.37 \\ 0.45 \\ 0.18 \end{bmatrix} = \begin{bmatrix} 0.329 \\ 0.525 \\ 0.146 \end{bmatrix}$$

$$P(2\tau) = T(\tau)P(1\tau) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.8 & 0.3 \\ 0.2 & 0.0 & 0.4 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.37 \\ 0.45 \\ 0.18 \end{bmatrix}$$

Simple example

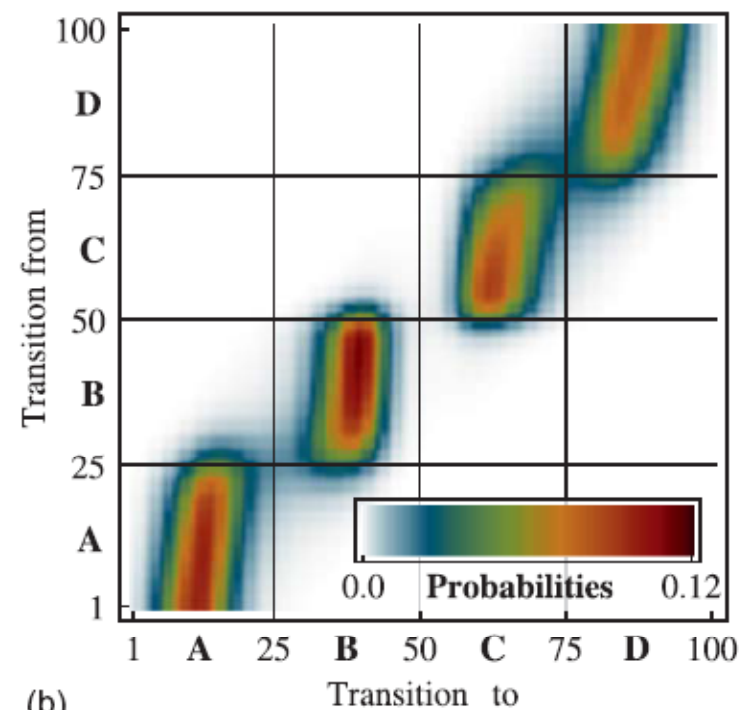
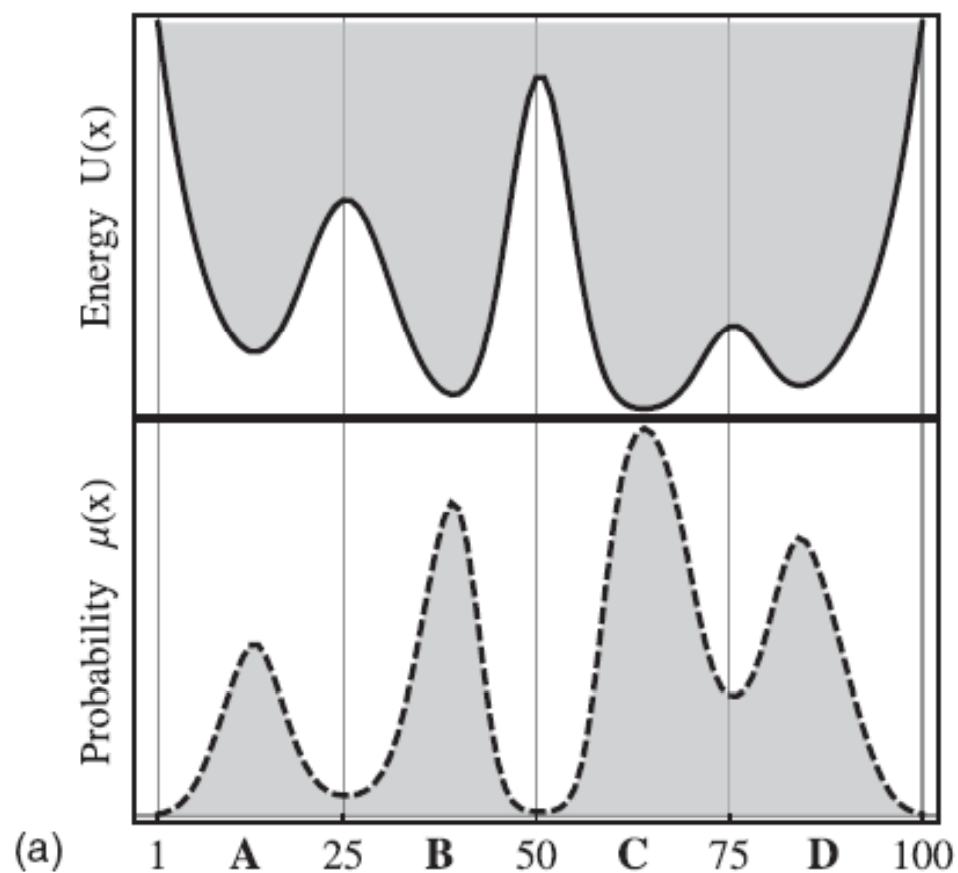
$$P(13\tau) = \begin{bmatrix} 0.30002 \\ 0.59993 \\ 0.10005 \end{bmatrix}$$

$$P(14\tau) = \begin{bmatrix} 0.30001 \\ 0.59996 \\ 0.10002 \end{bmatrix}$$

$$P(15\tau) = \begin{bmatrix} 0.30001 \\ 0.59998 \\ 0.10001 \end{bmatrix}$$

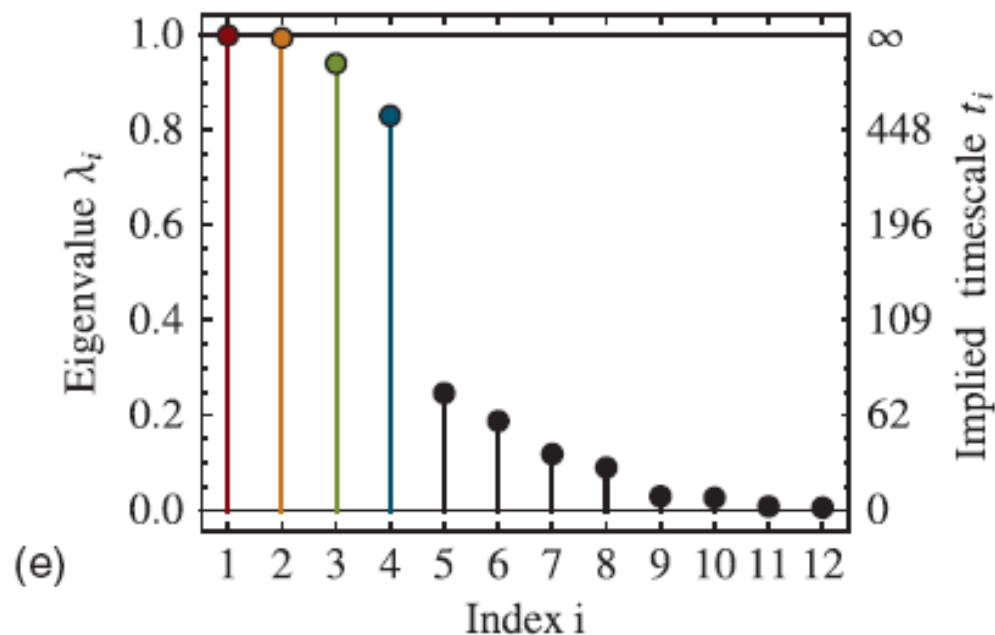
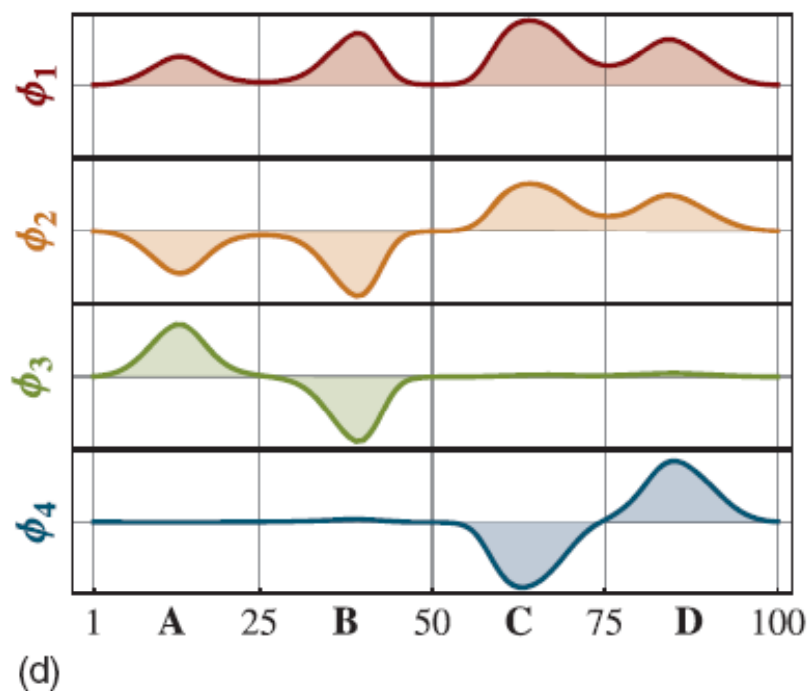
$$P(n\tau) = T(\tau)P((n-1)\tau) = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.8 & 0.3 \\ 0.2 & 0.0 & 0.4 \end{bmatrix} \begin{bmatrix} 0.3 \\ 0.6 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.6 \\ 0.1 \end{bmatrix} = P((n-1)\tau)$$

Building the MSM



$$\mathbf{T}(\tau) = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & & \\ \vdots & & \ddots & \\ P_{nl} & & & P_{nn} \end{bmatrix} \quad n=100$$

Building the MSM



$$T(\tau)\Phi_i = \lambda_i\Phi_i$$

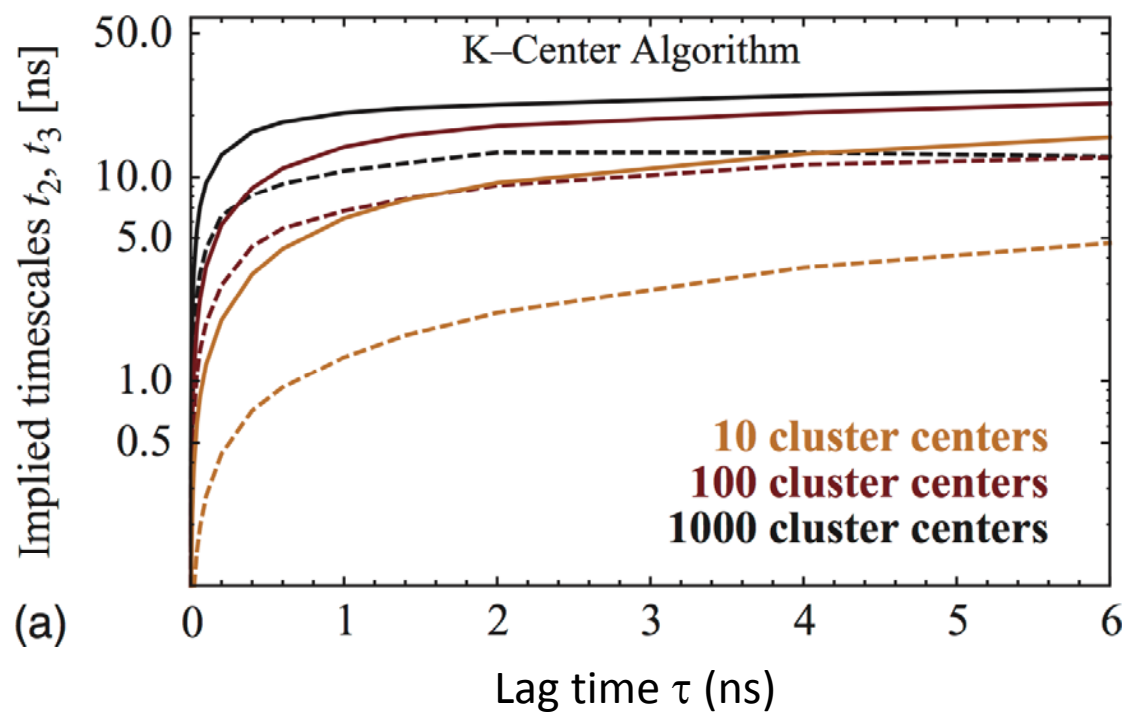
λ_i is an eigenvalue corresponding to eigenvector Φ_i

$$P(n\tau) = \sum_{i=1} c_i \lambda_i^n \Phi_i$$

J. Chem. Phys. **2011**, *134*, 174105

Implied timescales

Validation of MSM: #1



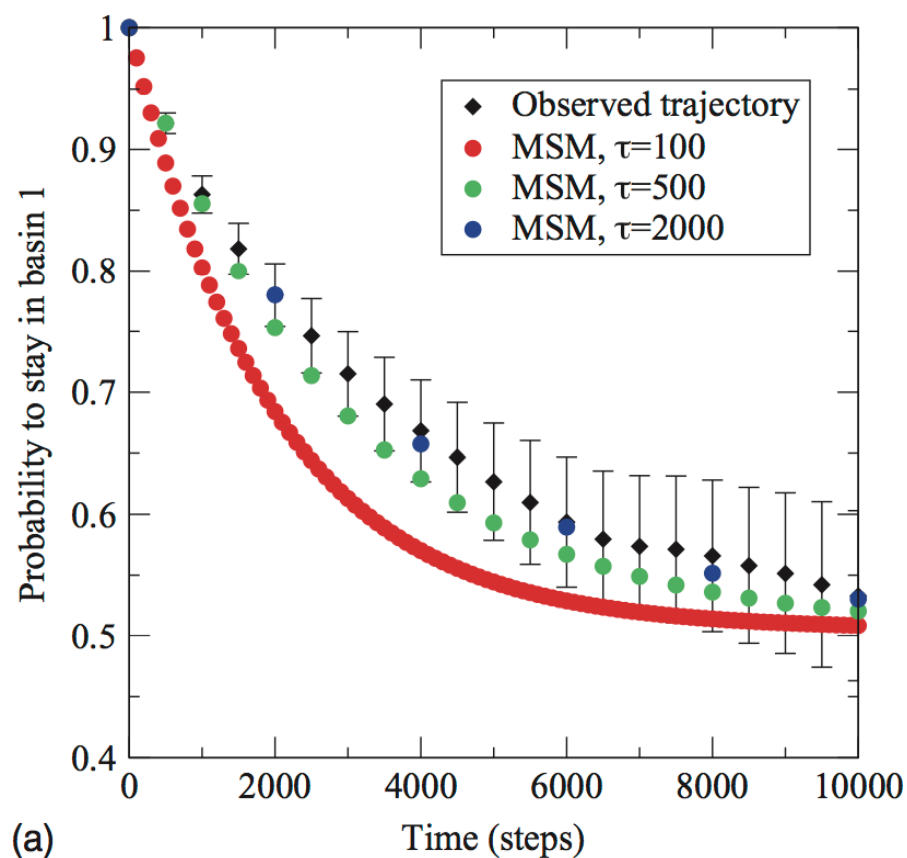
$$\tau_k = -\frac{\tau}{\ln \mu_k(\tau)}$$

Validation of MSM: #2

Chapman-Kolmogorov Test

$$P_{MD}(n\tau) = [P_{MSM}(\tau)]^n$$

To check if A Markovian model can “well-represent” MD.



What can we know from an MSM?

- Thermodynamic information
 - Equilibrium populations

- Kinetic information
 - Mean first passage time
 - Dominant Pathways (pathways with major flux)

Mean First Passage Time (MFPT)

- Considering all pathways that goes from state i to state j , what is the mean time a configuration at state i approaching state j for the first time?

$$F_{if} = \tau + \sum_{j \neq f} P_{ij} F_{jf}$$

MFPT for 3-state Model

$$F_{if} = \tau + \sum_{j \neq f} P_{ij} F_{jf}$$

$$F_{12} = \tau + P_{11}F_{12} + P_{13}F_{32}$$

$$F_{13} = \tau + P_{11}F_{13} + P_{12}F_{23}$$

$$F_{21} = \tau + P_{22}F_{21} + P_{23}F_{31}$$

$$F_{23} = \tau + P_{21}F_{13} + P_{22}F_{23}$$

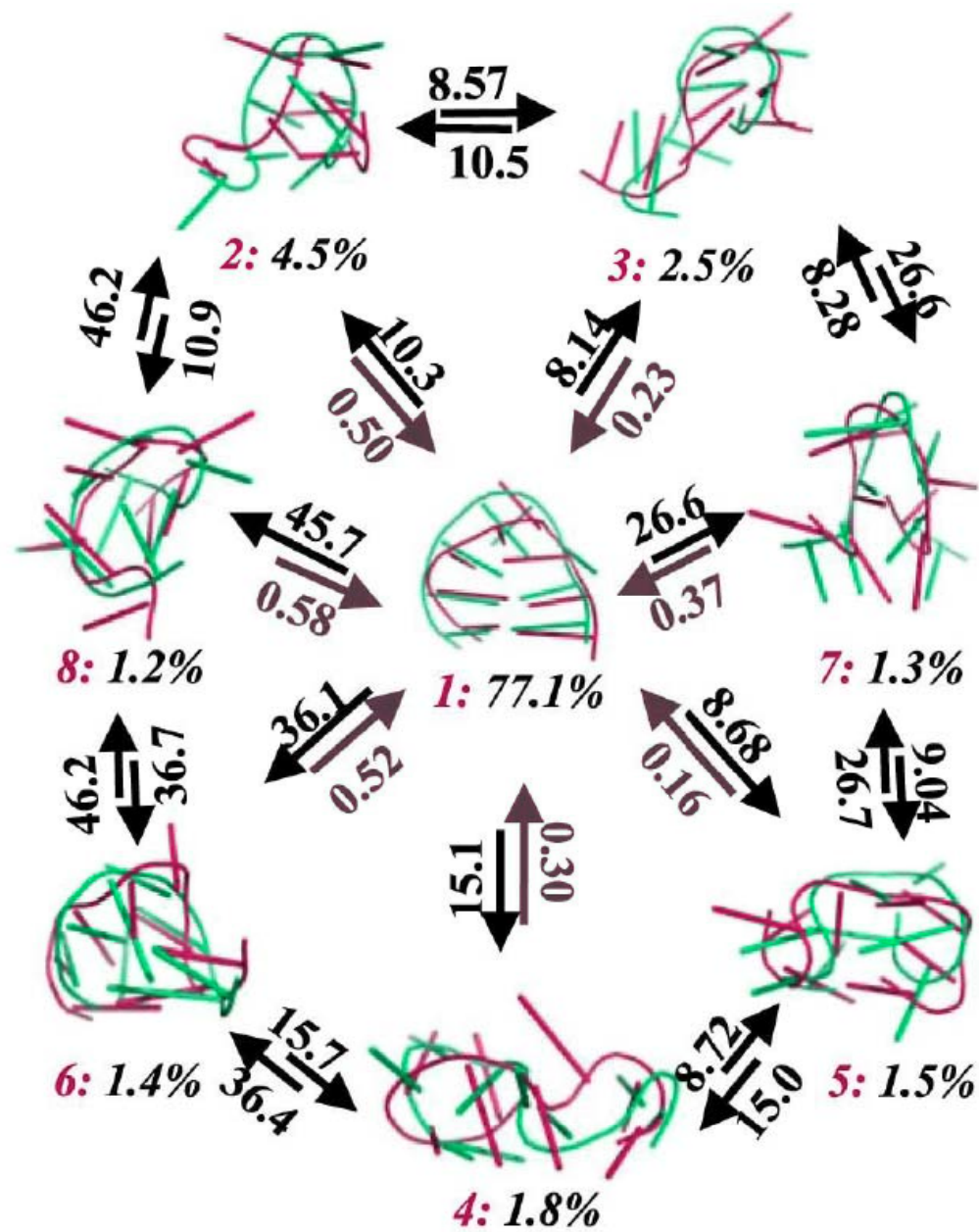
$$F_{31} = \tau + P_{32}F_{21} + P_{33}F_{31}$$

$$F_{32} = \tau + P_{31}F_{12} + P_{33}F_{32}$$

Mean first passage time

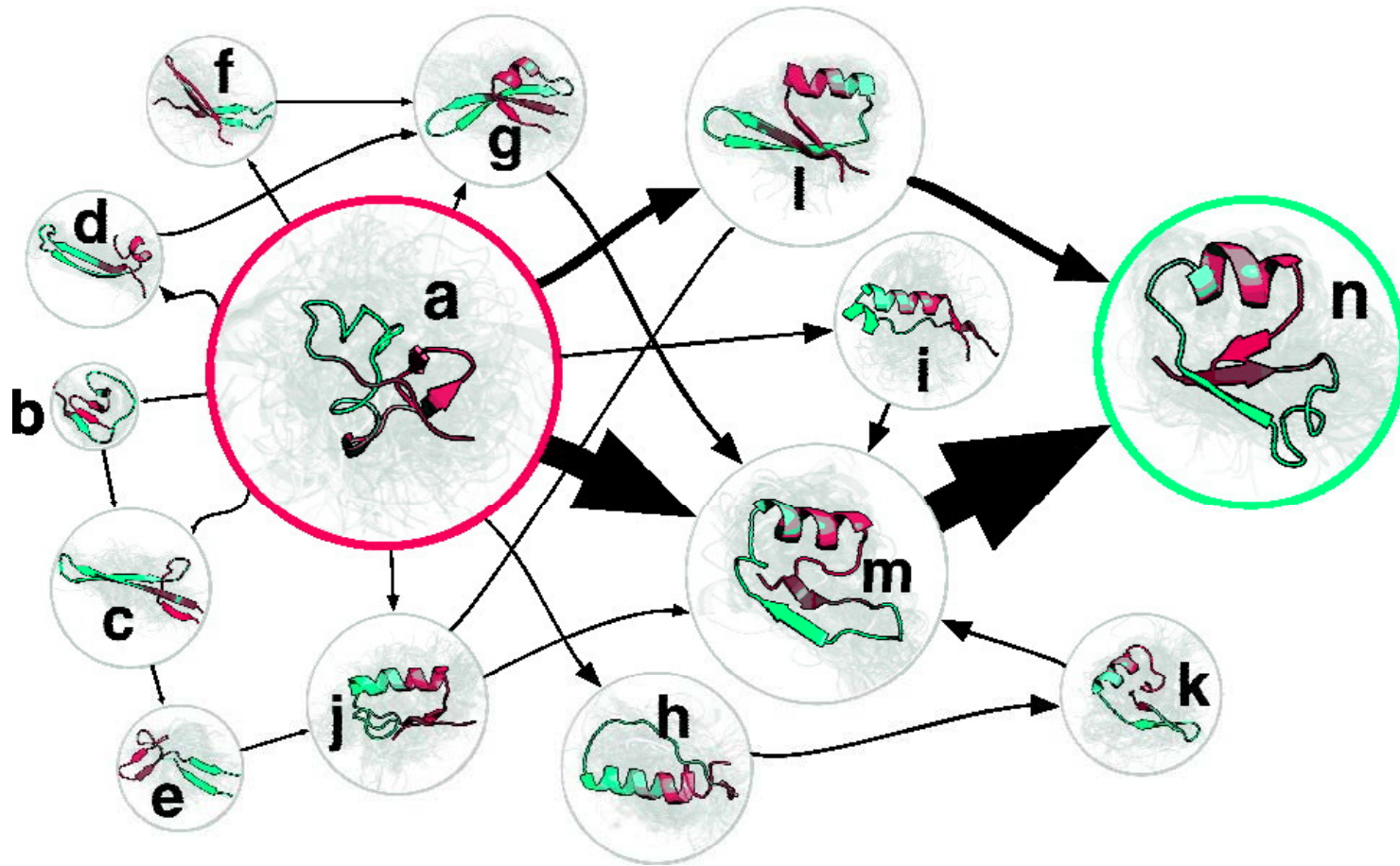
MFPT: the mean time it takes to reach a given metastable state f for the first time when starting from another state i .

$$MFPT_{if} = \sum_j P_{ij} \times (t_{ij} + MFPT_{jf})$$



Unit: μs

Transition path theory

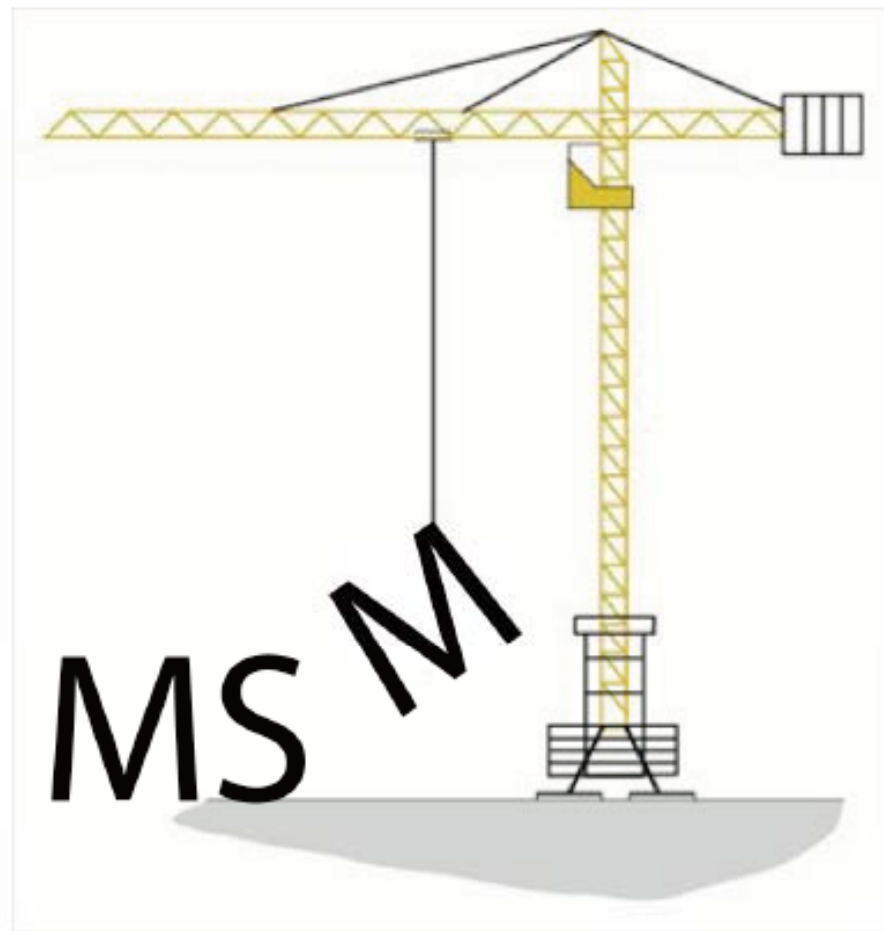


Introduction to MSM: Part II

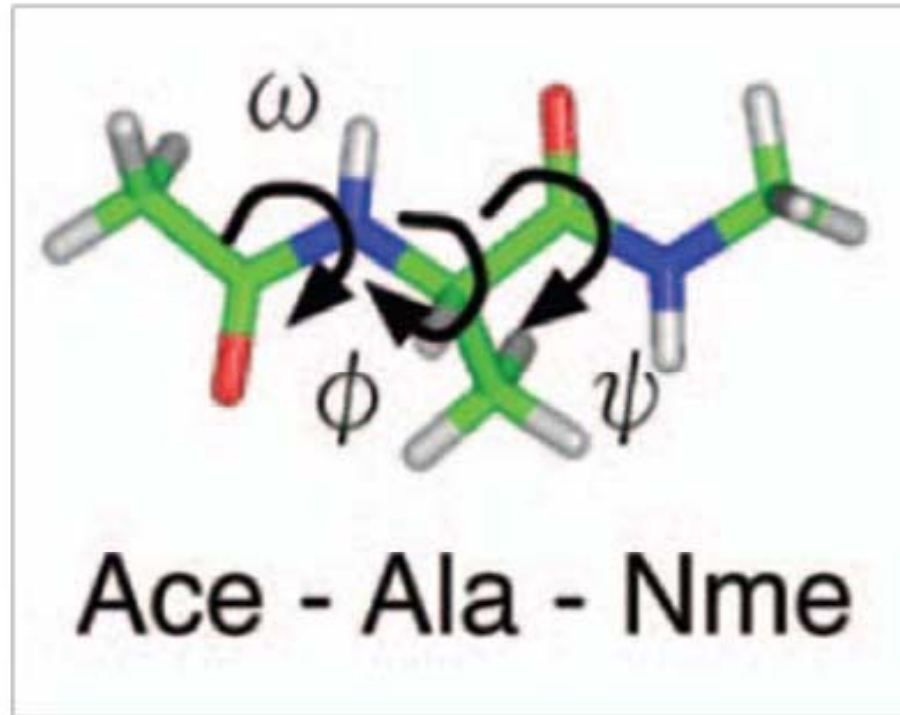
MSMBuilder 1.0

Gregory R. Bowman

Please reference GR Bowman, X Huang, and VS Pande. *Methods* 2009. Using generalized ensemble simulations and Markov state models to identify conformational states



Example system



Constant temperature MD simulations (400K).

Amber96 force field.

100 500-ps MD simulations with conformation stored at 1ps.

Total conformation: 50,000

Fig. 5. Alanine-dipeptide system

```
$ ls  
assignments/ atom_indices generators/ trajectories/  
trajlist
```

```
$ ls trajectories/  
p4432_RUN511_CLONE43/  
p4432_RUN511_CLONE43-trajlist  
p4432_RUN743_CLONE29/  
p4432_RUN743_CLONE29-trajlist
```

```
$ head trajectories/p4432_RUN743_CLONE29-trajlist  
trajectories/p4432_RUN743_CLONE29/4432-743-29-1  
trajectories/p4432_RUN743_CLONE29/4432-743-29-2
```

```
$ head trajlist  
p4432_RUN511_CLONE43-trajlist  
p4432_RUN743_CLONE29-trajlist
```

```
$ head atom_indices  
5 1 3 4 5 9
```

doFastGromacsClustering: Clustering your data

doFastGromacsClustering -a atom_indices -k 120 -t trajlist -w -x 5001

Option	Description
-a	List of atom index files to read in, separated by spaces. At least one is required. There must be one corresponding to each trajectory list file and the ordering of atom index and trajectory list files must be the same.
-c	Specify a cutoff distance. The clusterer will count the number of conformations within each cluster that are within this cutoff distance of the cluster center.
-d	If this flag is specified then don't ignore the last snapshot of each xtc file. By default the last snapshot of each xtc file is dropped to avoid duplication with the first snapshot of the next xtc file. This options turns this behavior off so all snapshots is used.
-j	If this option is specified then just print the expected number of configurations and the size they'll take in memory without actually loading the data into memory or doing any clustering. Requires that the <code>-x</code> option be set.
-k	The number of clusters (or microstates) to generate.
-n	Only use every n'th snapshot if this option is specified.
-o	Name of file to output statistics on each cluster to.
-s	The index of the conformation to use as the first cluster center (ranges from 0 to n-1 where n is the number of conformations to be clustered). This option should generally not be used.
-t	List of trajectory list files to be read in separated by spaces. At least one is required. There must be one corresponding to each atom index file and the ordering of atom index and trajectory list files must be the same.
-w	If this flag is specified then the assignments will be written to the assignments directory after the clustering is finished and statistics on each cluster will be written to the stats.dat file (see the next section for the format of this file).
-x	The number of snapshots in each xtc file.

```
$ ls
assignments/ atom_indices generators/ trajectories/
trajlist
```

```
$ head assignments/ p4432_RUN743_CLONE29-trajlist
0 0
0 0
1 5
...
```

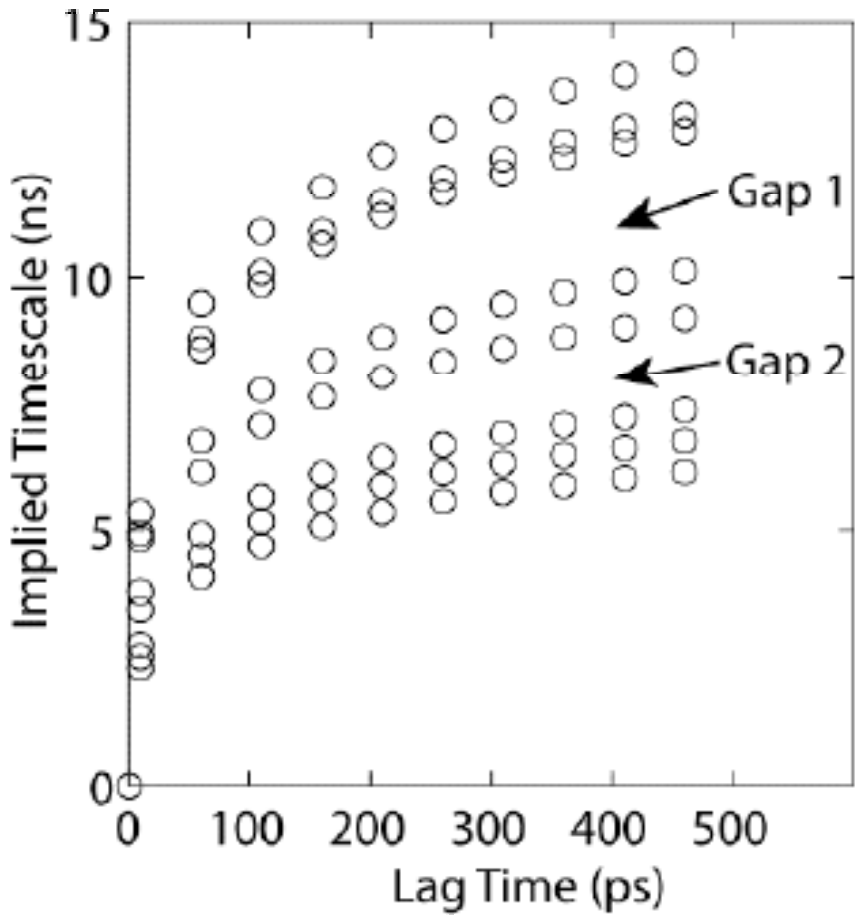
Check your ***stats.dat*** file

BuildMSMsAsVaryLagTime.py: validating a model

BuildMSMsAsVaryLagTime.py -d ../ -f ../trajlist -i 200 -m 5000 -t 2 -n 20 -s 120

Option	Description
-b	Number of iterations of bootstrapping to perform. [default: 0]
-d	Head directory containing the assignments and trajectories directories. [default: .]
-f	Filename of trajectory list. [default: trajlist]
-i	(required) Interval between lag times to build models for. Builds models for lag times of 1, interval, 2*interval... maxLagTime.
-m	(required) Maximum lag time to go up to.
-n	number of eigenvalues (or implied timescales) to consider in analysis [default: 100]
-p	File listing mapping of microstates to macrostates.
-s	(required) Number of microstates.
-t	(required) Time in ps between entries in the assignment files.
-u	Number of macrostates to build. If unspecified then just get MicroMSMs as vary lag time, otherwise build MacroMSMs.

Lumping Microstates into Macrostates



BuildMacroMSM.py: Building the Macrostate MSM

BuildMacroMSM.py -d ../ -f ../trajlist -l 1 -m 3 -s 300 -t 1 -n 0 -x

Option	Description
-d	Head directory containing the assignments and trajectories directories. [default: .]
-f	Filename of trajectory list. [default: trajlist]
-i	Number of steps to do for each SA run. [default: 20,000]
-l	(required) Lag time in number of entries in the assignment files to use for the model.
-m	(required) Number of macrostates to build.
-n	Number of simulated annealing (SA) runs to do. [default: 20]
-o	Save the micro and macro msm transition matrices to the specified directory.
-p	Use this flag to print the 50 largest eigenvalues without making an MSM. Useful for determining the number of Perron clusters (number of macrostates).
-r	Read the micro MSM transition matrices from the specified directory.
-s	(required) Number of microstates.
-t	(required) Time in ps between entries in the assignment files.
-x	Use the simplex version of PCCA (called PCCA+). When using this flag, we recommend also using “-n 0” to turn off simulated annealing because PCCA+ includes its own optimization stage.
-z	Disable the optimization stage in PCCA+.

WriteMacroAssignments.py -d ../ -f ../trajlist -m mapMicroToMacro.dat

GetMicroCentersByMacroState.py: Getting All Microstate Centers

GetMicroCenters.py -d ./ -f trajlist -t ../file00001.pdb -x ./index.ndx

Option	Description
-d	Head directory containing the assignments and trajectories directories. [default: .]
-e	Full path to trjconv executable. [default: \$MSMBUILDERHOME/Extras/trjconv/trjconv]
-f	Filename of trajectory list. [default: trajlist]
-m	(required) File listing mapping of microstates to macro states.
-o	Directory name to put output into. [default: StateCenters]
-p	Type of file to output (gro, pdb, or other file types allowed by trjconv). [default: gro]
-t	(required) Path to tpr file.
-x	(required) Path to index (.ndx) file.

GetMacroMSMPopStats.py: Bootstrapping

GetMacroMSMPopStats.py -s 300 -m 3 -d ../ -f ../trajlist -t 1 -l 3500 -n 95 -b 5 -y Yes

Option	Description
-b	Number of iterations of bootstrapping to do. [default: 10]
-d	Head directory containing the assignments and trajectories directories. Should only be specified if you want to override the value in the existing MSM read in with the -r option.
-f	Filename of trajectory list. Should only be specified if you want to override the value in the existing MSM read in with the -r option.
-l	Lag time in number of entries in the assignment files to use for the model. If this option is not specified then the value used to build the MSM will be used.
-m	Number of macrostates.
-n	(required) Number of trajectories to include in each iteration of bootstrapping. The number of trajectories listed in the trajlist file is generally a good choice.
-s	Number of microstates.
-t	Time in ps between entries in the assignments files.

Mean first passage time (MFPT): the mean time τ_i it takes to reach a given metastable state m for the first time when starting from another state i .

$$\begin{aligned}
 f_{1m} &= \tau + T_{11}f_{1m} + T_{12}f_{2m} + \cdots + T_{1m}f_{mm} & -\tau &= -f_{1m} + T_{11}f_{1m} + T_{12}f_{2m} + \cdots + T_{1m}f_{mm} \\
 f_{2m} &= \tau + T_{21}f_{1m} + T_{22}f_{2m} + \cdots + T_{2m}f_{mm} & -\tau &= -f_{2m} + T_{21}f_{1m} + T_{22}f_{2m} + \cdots + T_{2m}f_{mm}
 \end{aligned}$$

$$\begin{bmatrix}
 T_{11}^{-1} & & \cdots & & T_{1m} \\
 \vdots & T_{22}^{-1} & & & T_{2m} \\
 & & \ddots & & \\
 T_{m1} & \cdots & & T_{m-1,m-1}^{-1} & T_{m-1,m} \\
 0 & 0 & \cdots & 0 & 1
 \end{bmatrix} \times \begin{bmatrix}
 f_{1m} \\
 f_{2m} \\
 \vdots \\
 f_{m-1} \\
 f_{mm}
 \end{bmatrix} = \begin{bmatrix}
 -\tau \\
 -\tau \\
 \vdots \\
 -\tau \\
 -\tau
 \end{bmatrix}$$

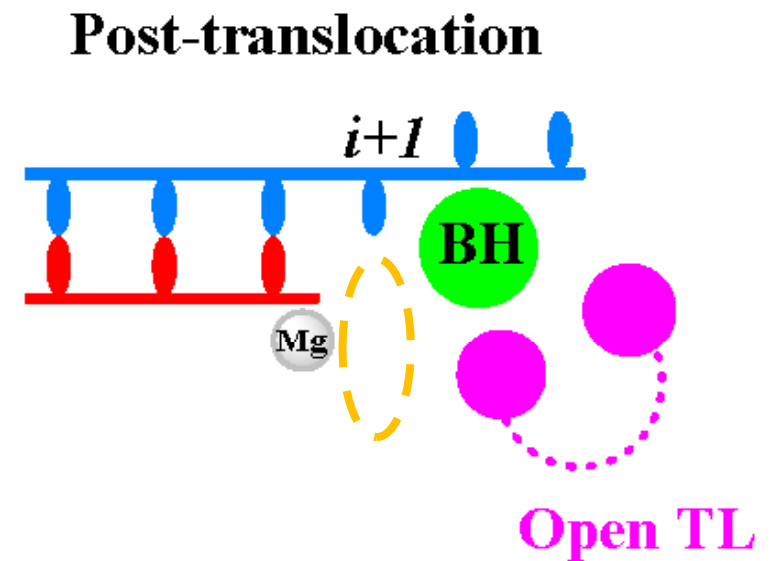
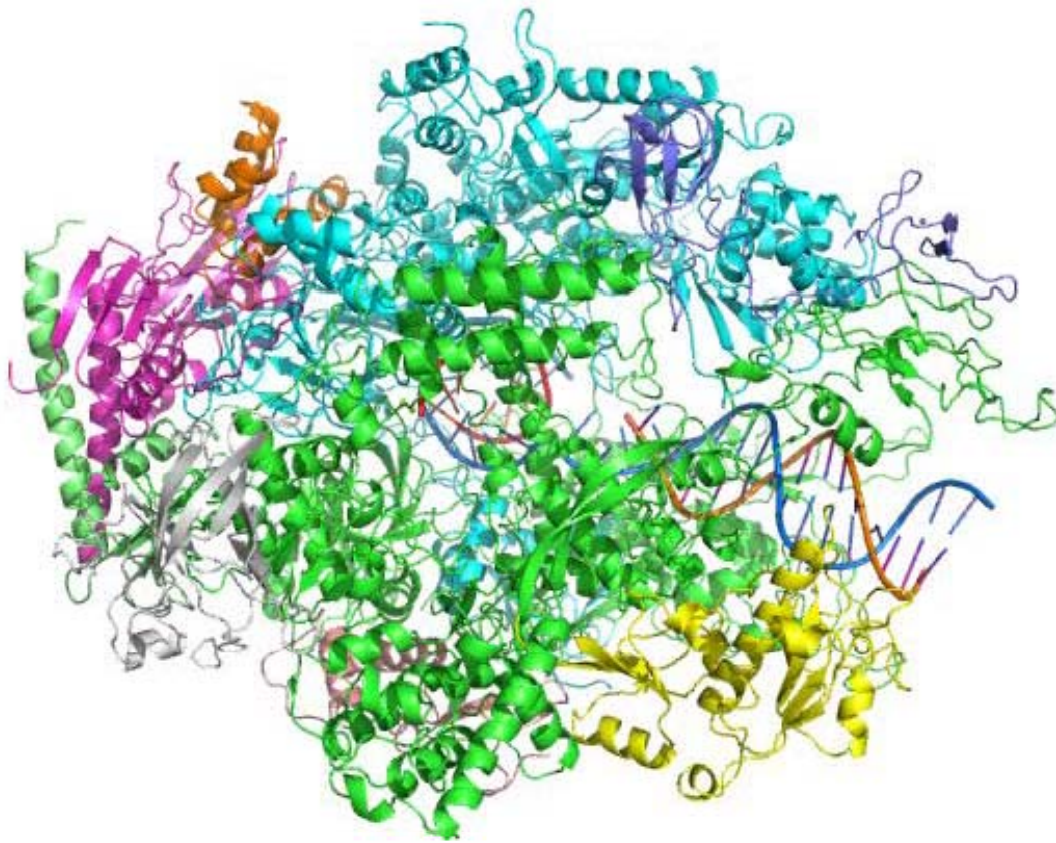
MFPT.py -f ../trajlist -s 300 -t 1 -l 3500 -u 3 -d ../

Options:

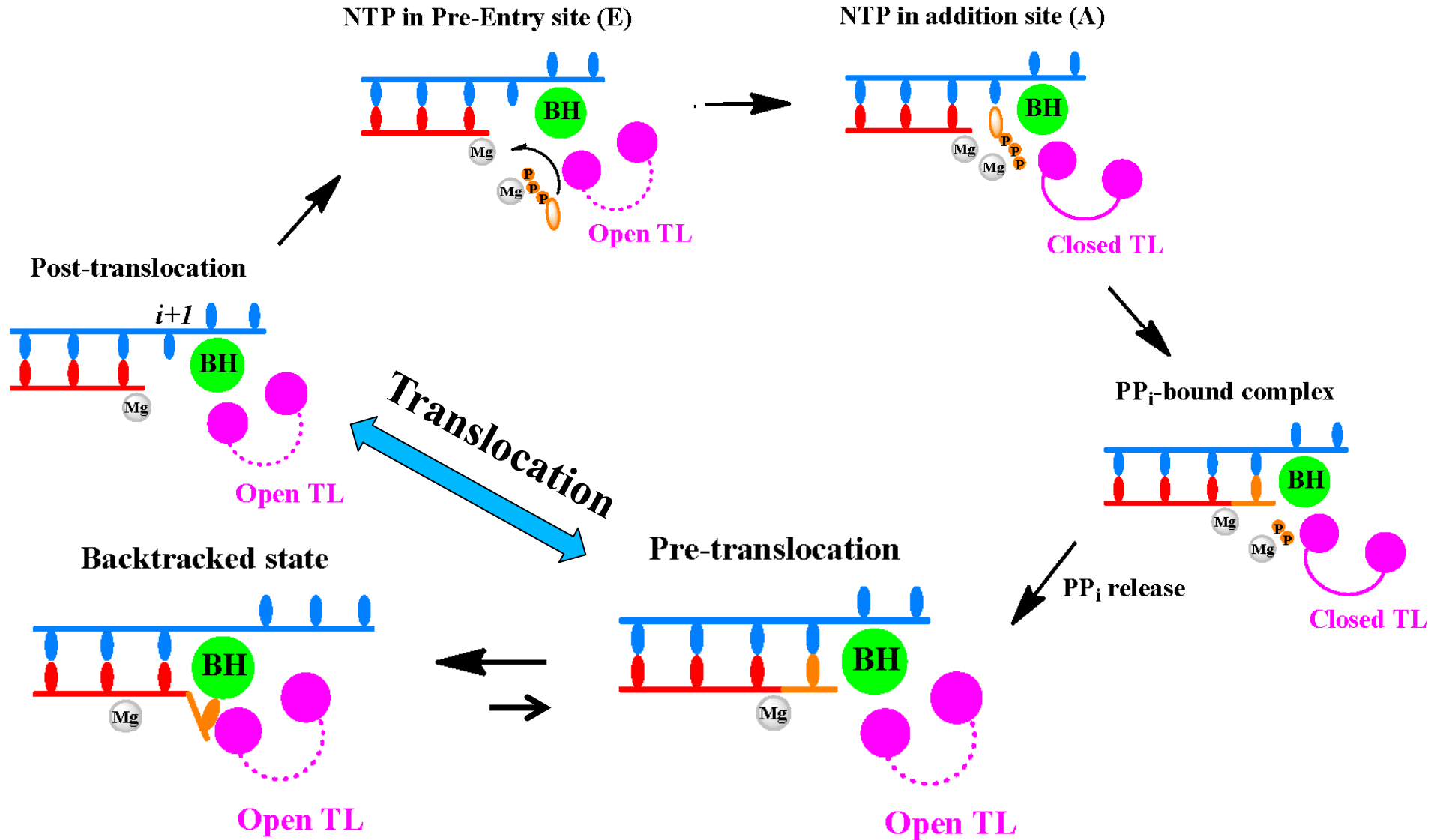
- h, --help show this help message and exit
- d HEADDIR, --head_dir=HEADDIR
Head directory containing the assignments and trajectories directories. [default: .]
- f TRAJLISTFN, --traj_list=TRAJLISTFN
Filename of trajectory list. [default: trajlist]
- p MAPMICROTOMACROFN, --mapMicroToMacroFn=MAPMICROTOMACROFN
File listing mapping of micro states to macro states.
- s NMICROSTATES, --numMicroStates=NMICROSTATES
(required) Number of microstates.
- t DT, --time_step=DT
(required) Time in ps between entries in the assignment files.
- k SKIPMACROZERO, --Skip_Macro_Zero=SKIPMACROZERO
Are we skipping Macrostate 0 when computing Macrostate implied timescale plots? Only useful when interfacing with Mapper, default should be 0
- l LAGT, --Lag_time=LAGT
(required) Lag time in step number.
- u NUMMACRO, --num_macro=NUMMACRO
Number of macrostates in the system.
- a COMPUTEALL, --Compute_ALL=COMPUTEALL
compute all the MFPT or only a subset of the states
- y GETRIDOF, --yes=GETRIDOF
if you want to get rid of the recrossing

Introduction to MSM: Part III

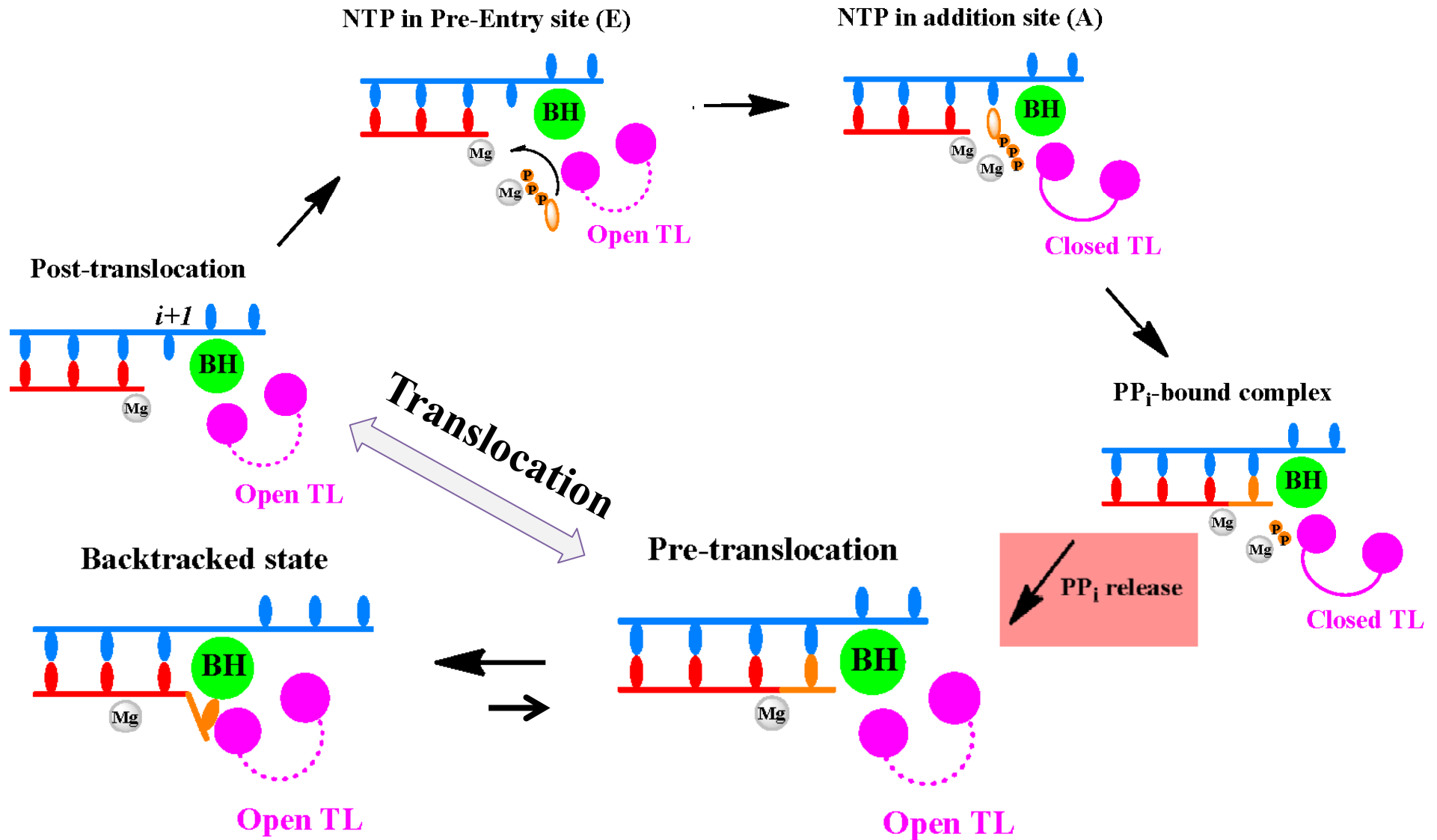
Structure of RNA polymerase II



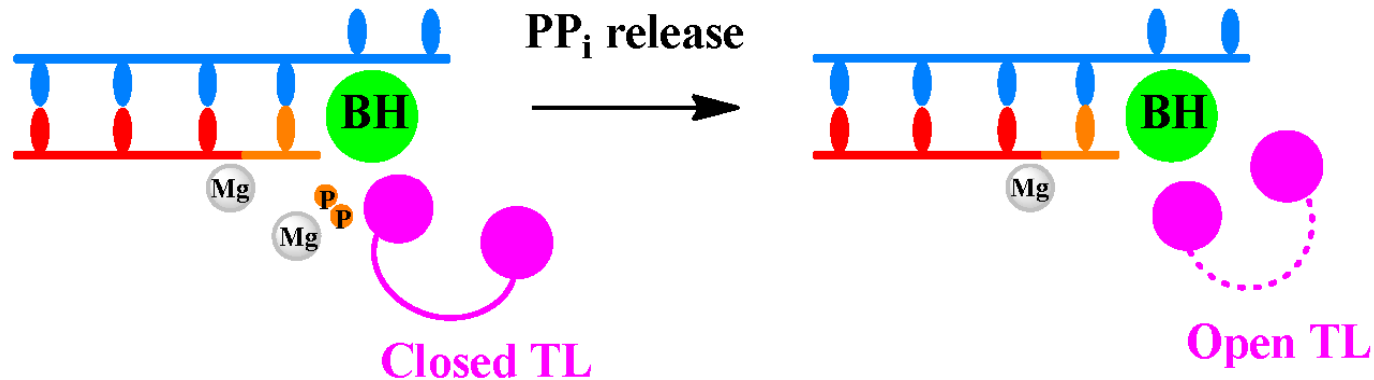
Nucleic Addition Cycle (NAC)



PP_i release mechanism

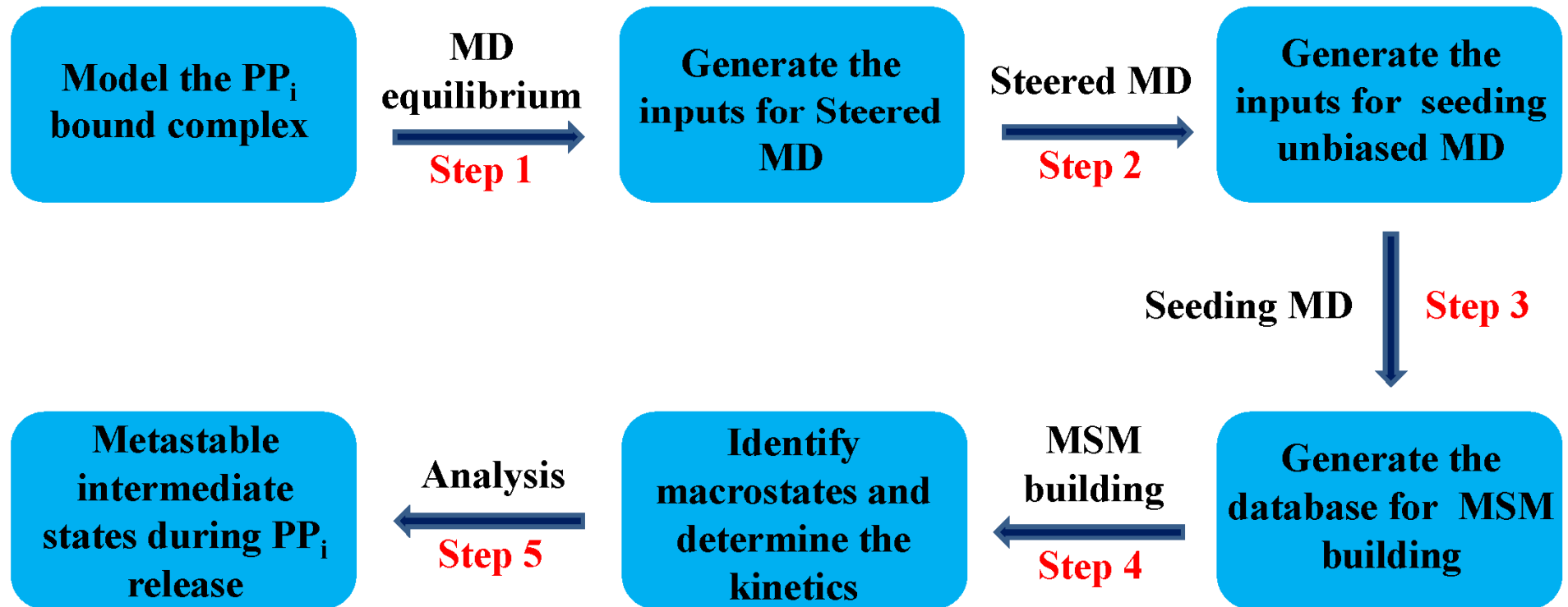


The problems we are interested

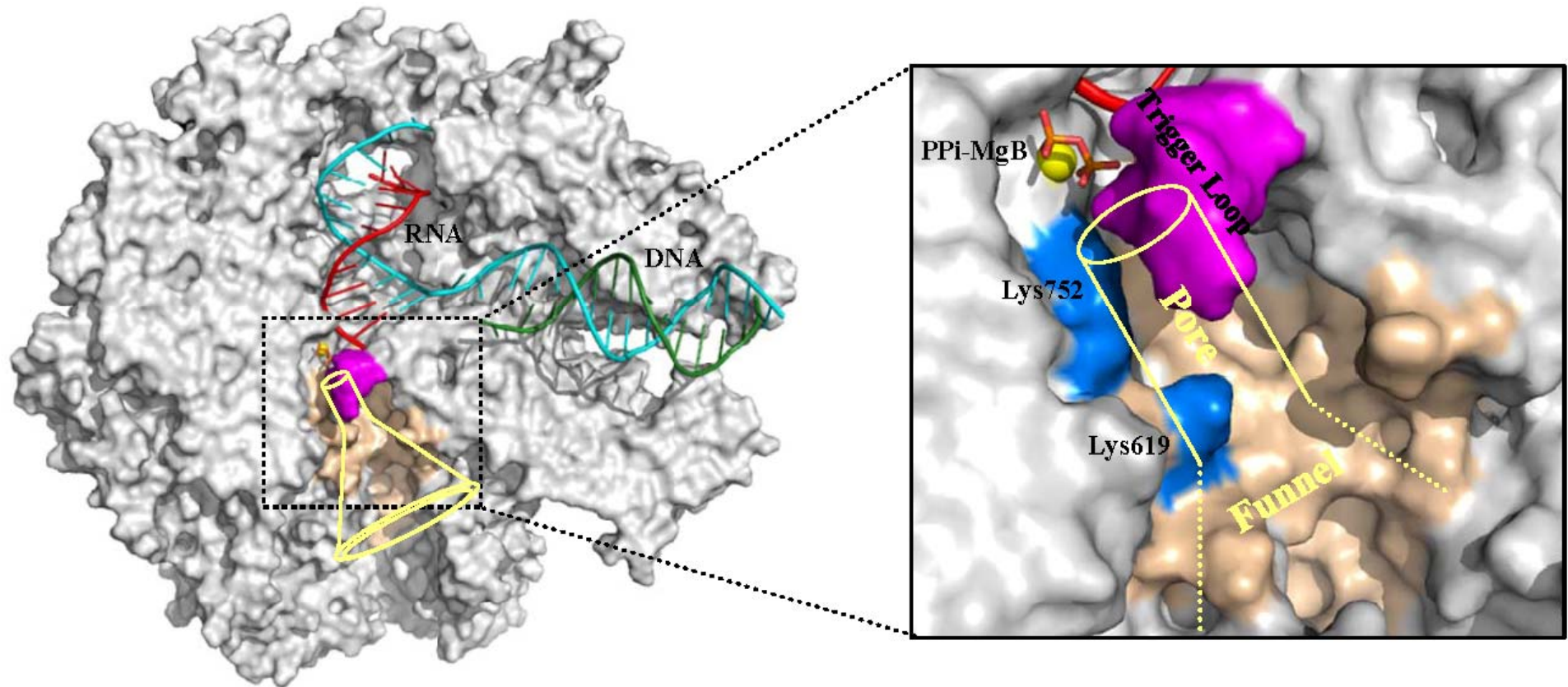


1. *How does the PP_i ion release along the secondary channel of pol II ?*
2. *What is the correlations between the PP_i release and the conformational changes of trigger loop? And translocation?*
3. *What is the difference of the PP_i release between the pol II and bacterial RNA polymerase?*

Simulation methodology



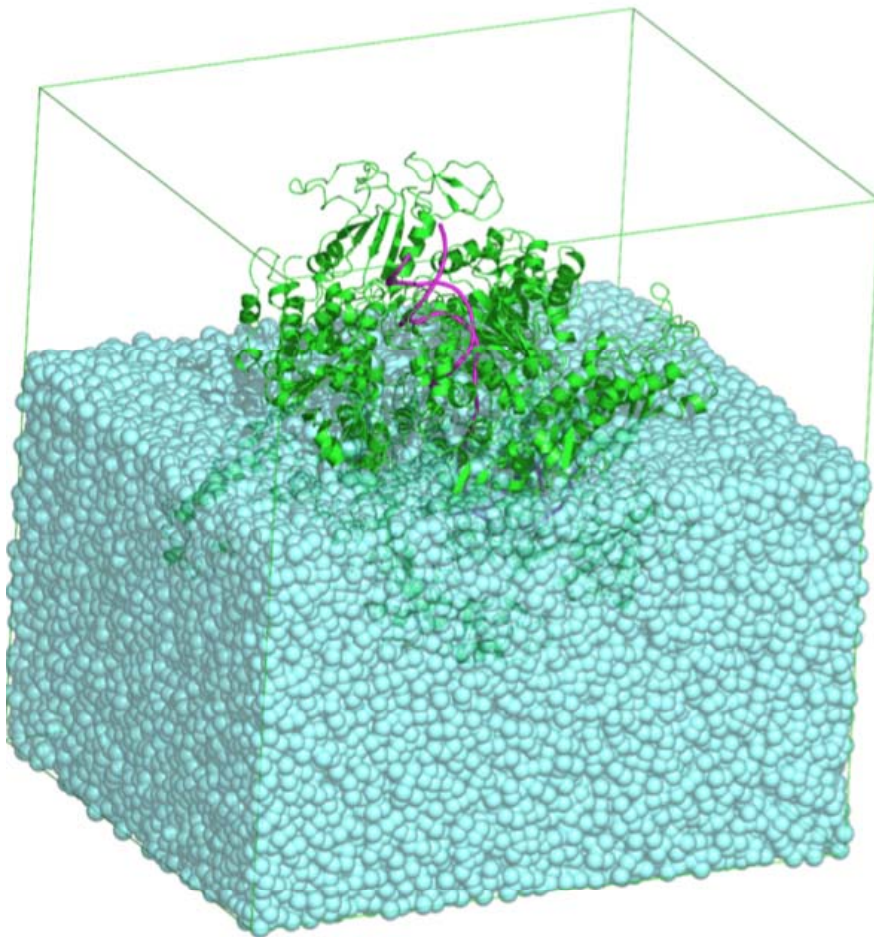
PPi-bound pol II complex



Based on GTP-bound pol II complex (2E2H)

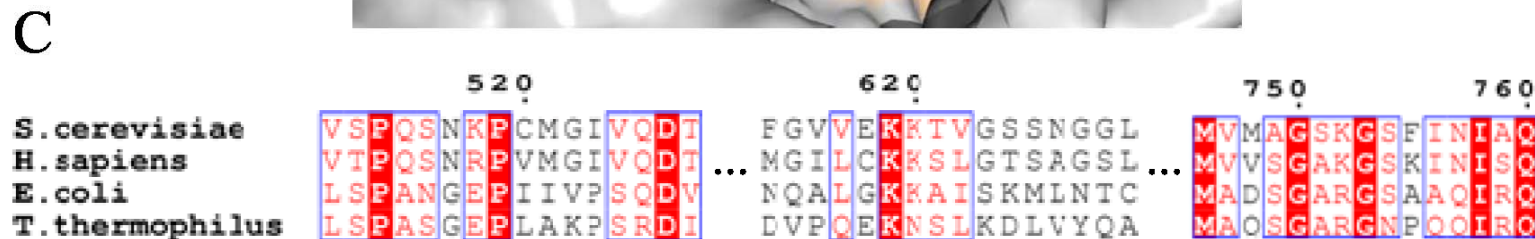
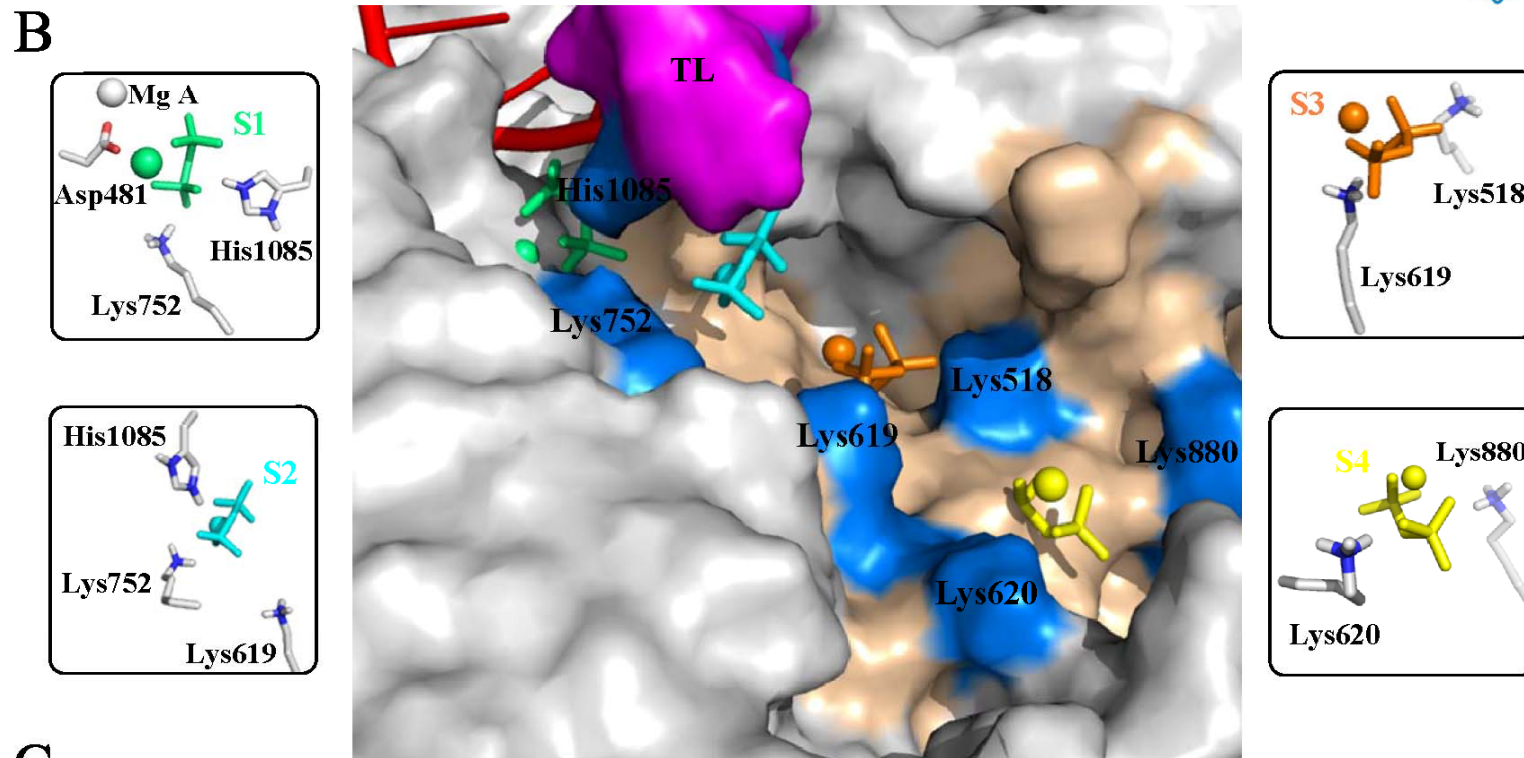
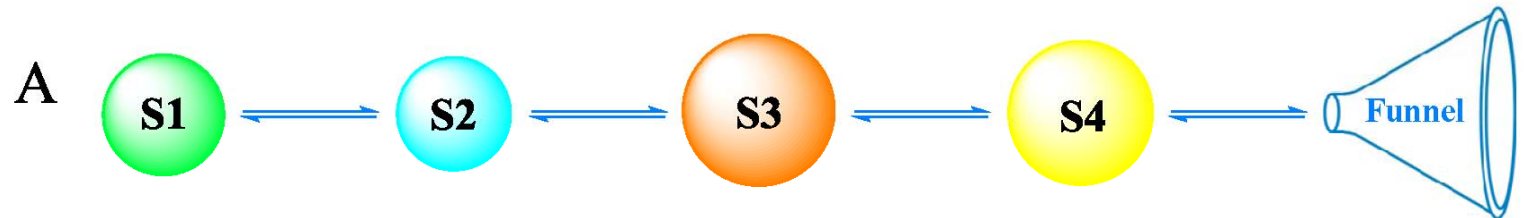
Da, LT et al, *J. Am. Chem. Soc.*, 134, 2399, (2012)

System setup

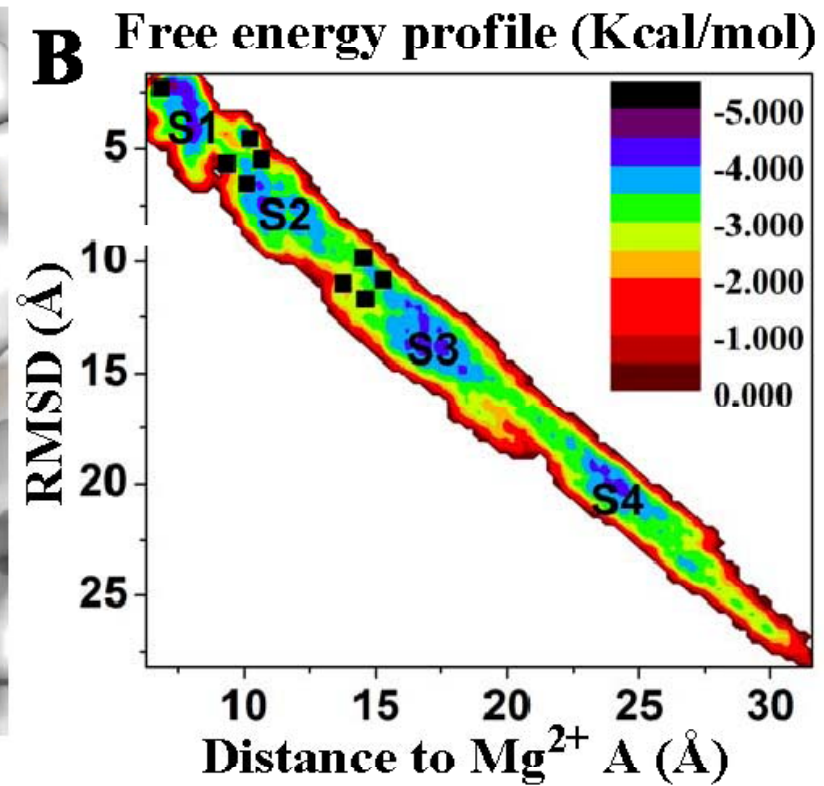
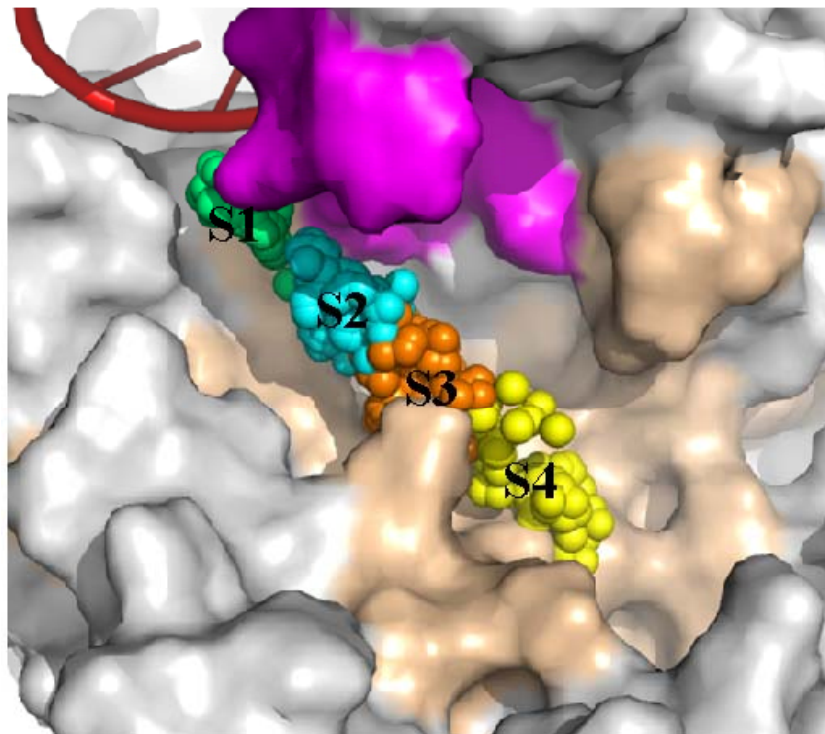


1. System size $\sim 370,000$ atoms. Protein (> 3000 residues), water, DNA, RNA, ions, etc.
2. Built the MSM based on 122 trajectories (6ns each).
3. Traced the PPI release dynamics around $1.5\mu\text{s}$

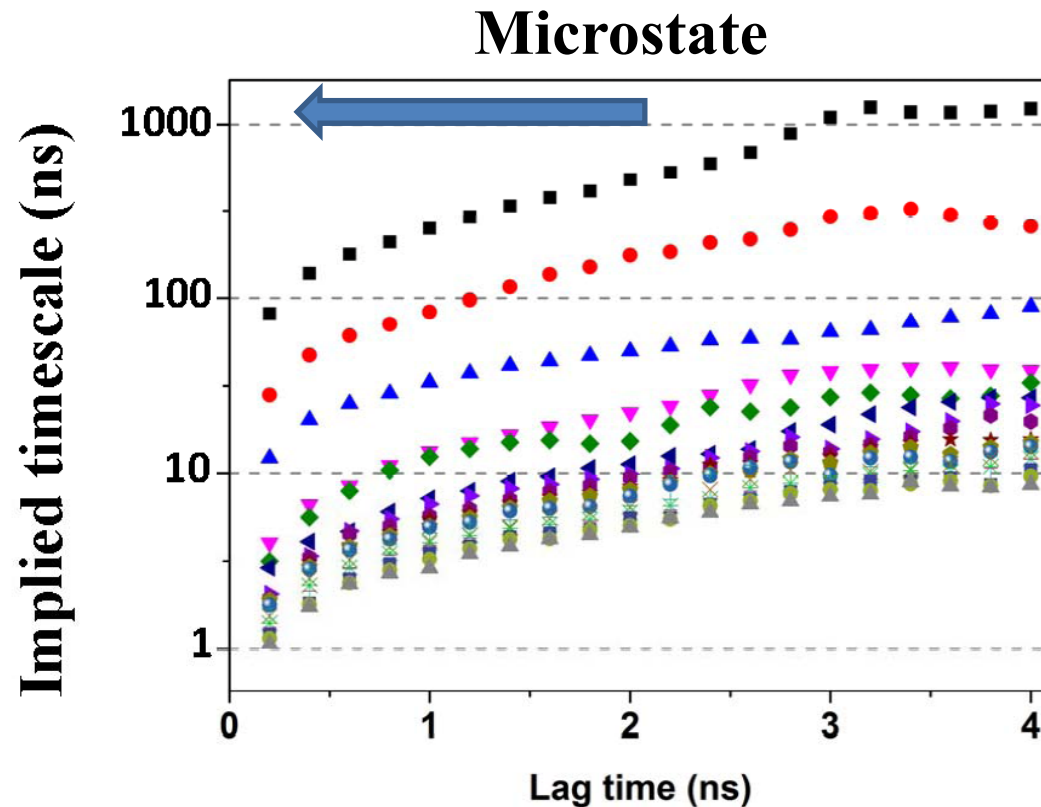
PPi release adopts a hopping mode



Projection of the MD conformations onto two coordinations



Implied timescale for the PPI release

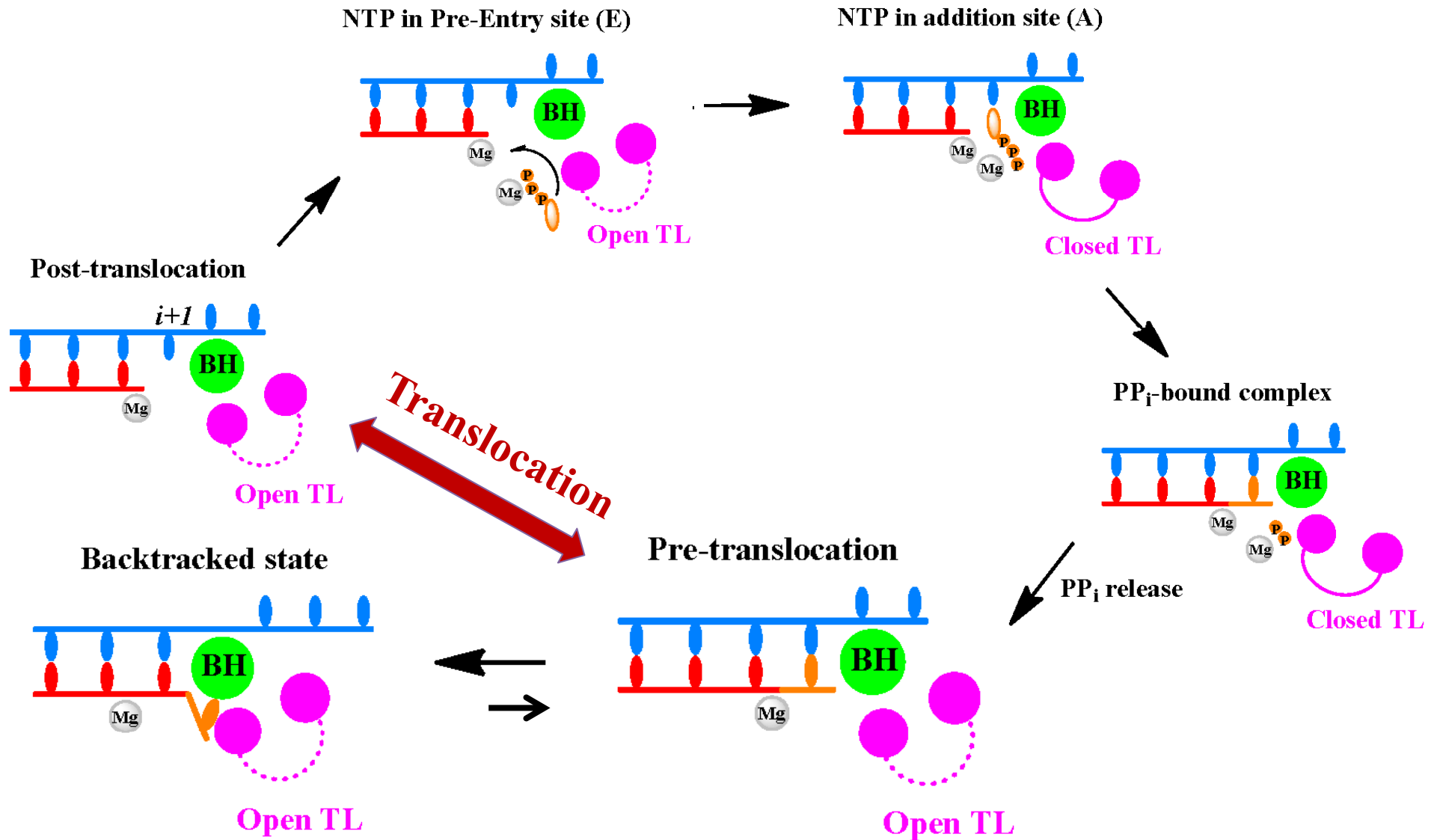


MFPT: the mean time it takes to reach a given metastable state f for the first time when starting from another state i .

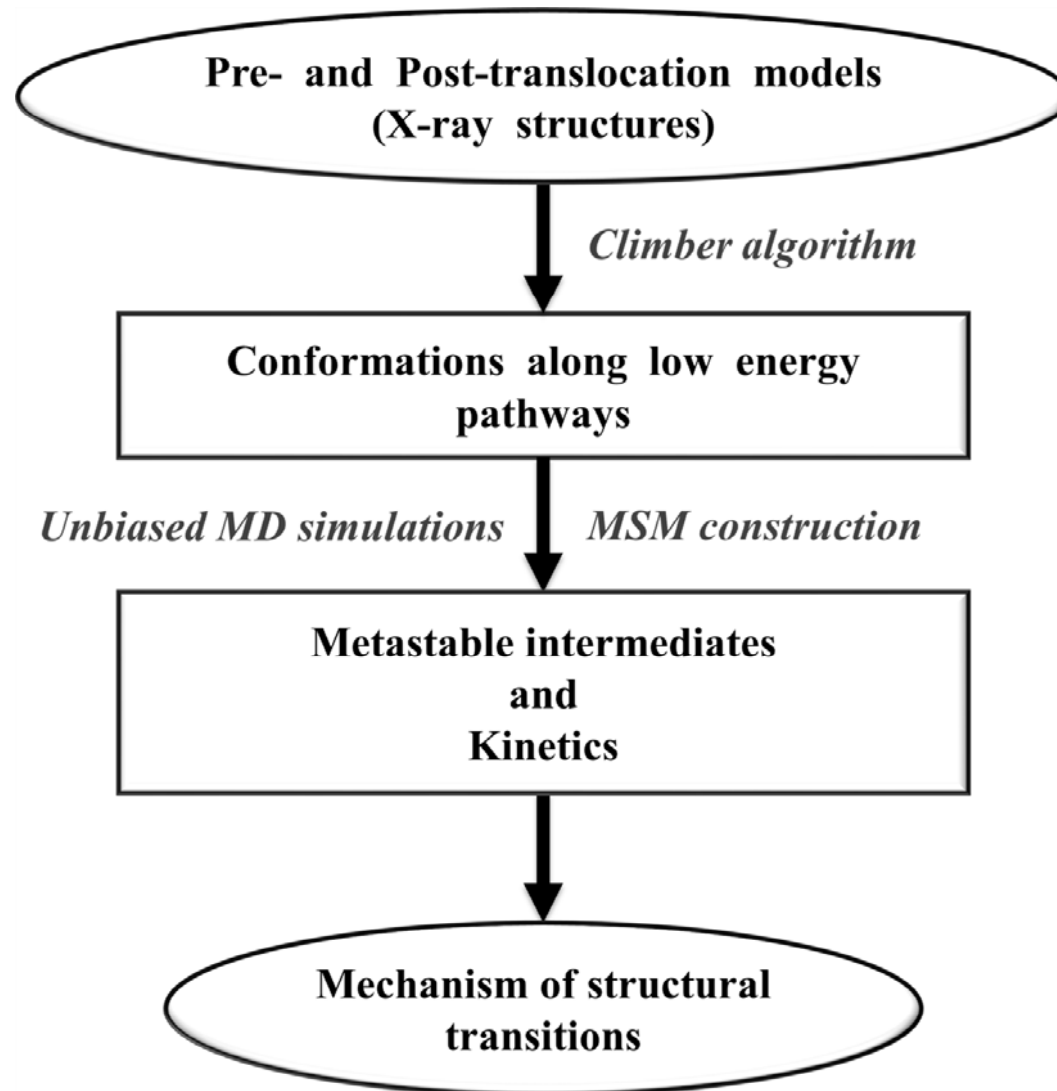
$$MFPT_{if} = \sum_j P_{ij} \times (t_{ij} + MFPT_{jf})$$

The timescale of the PPI release along the pore region is around $1.5\mu\text{s}$

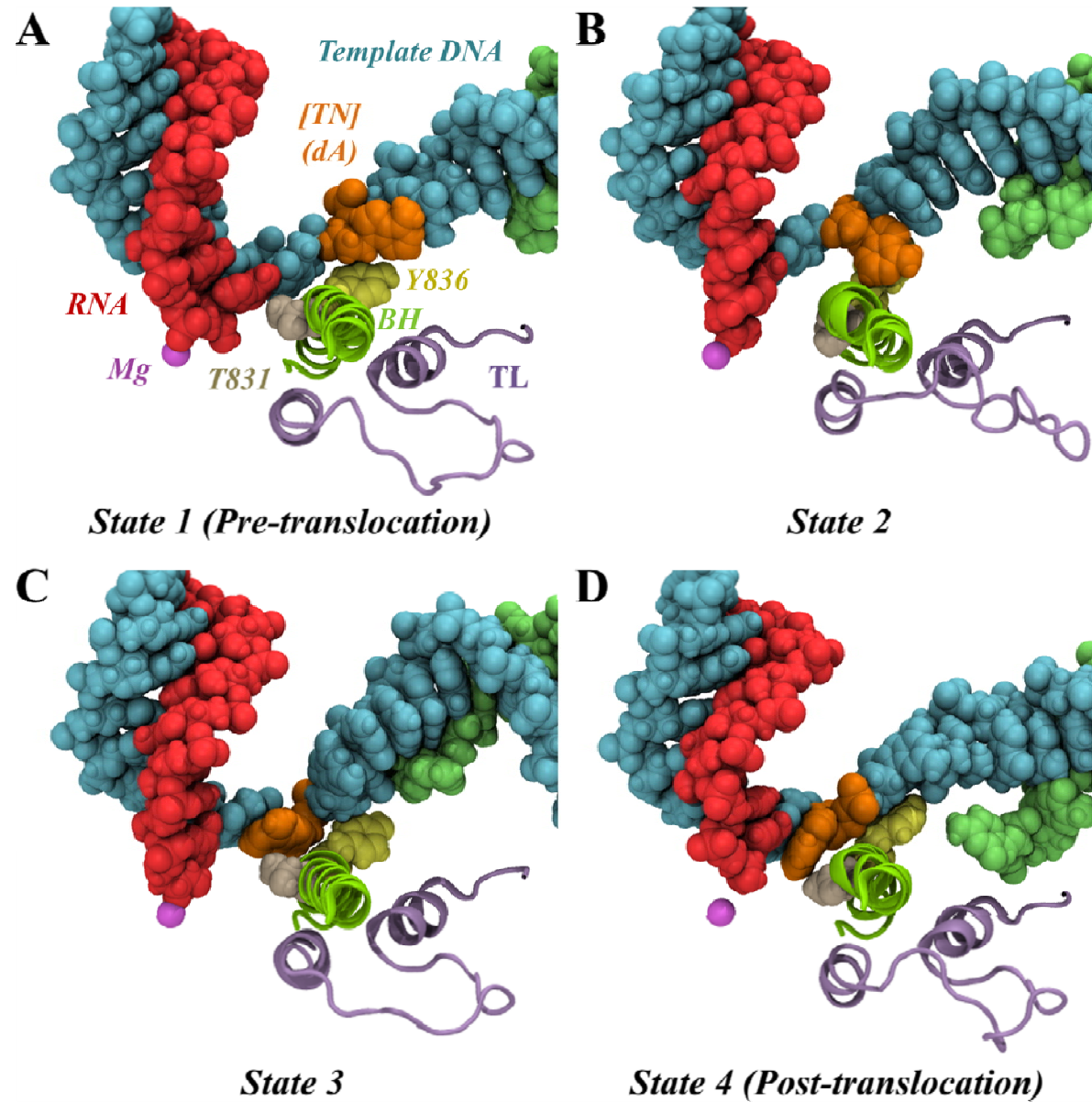
Translocation mechanism



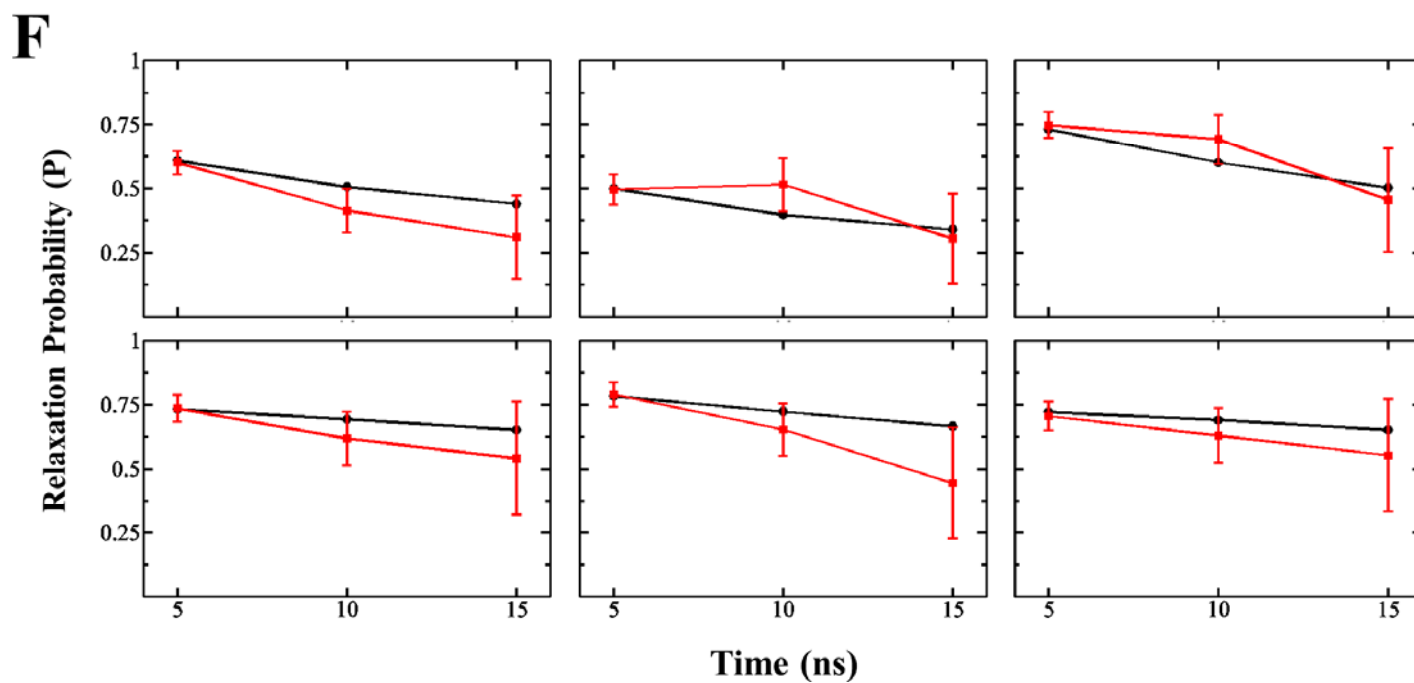
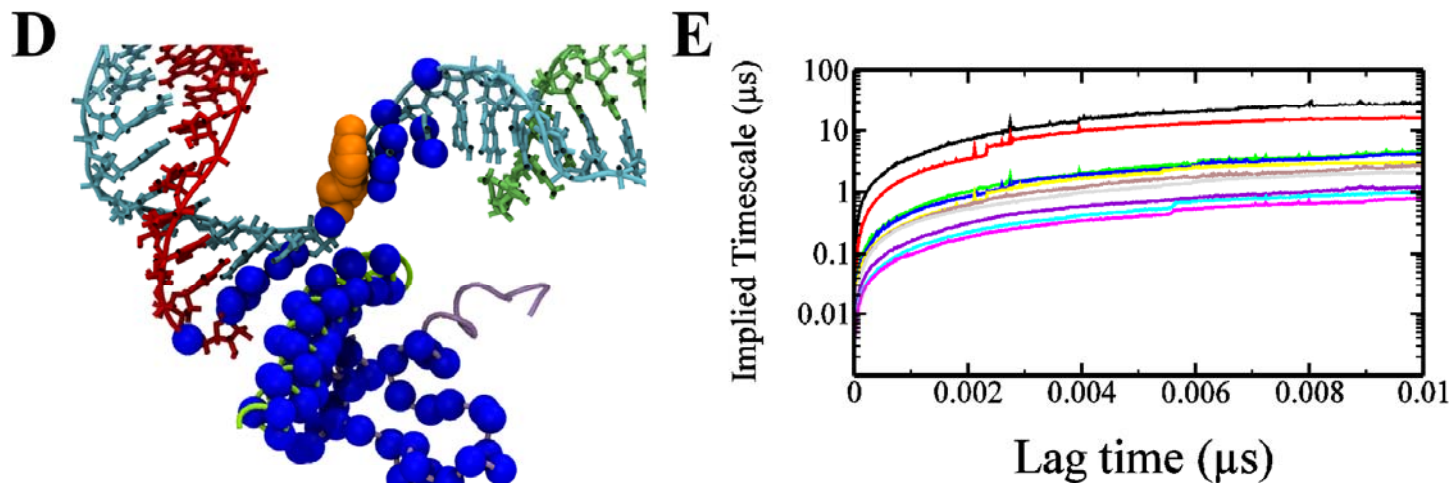
Simulation methodology



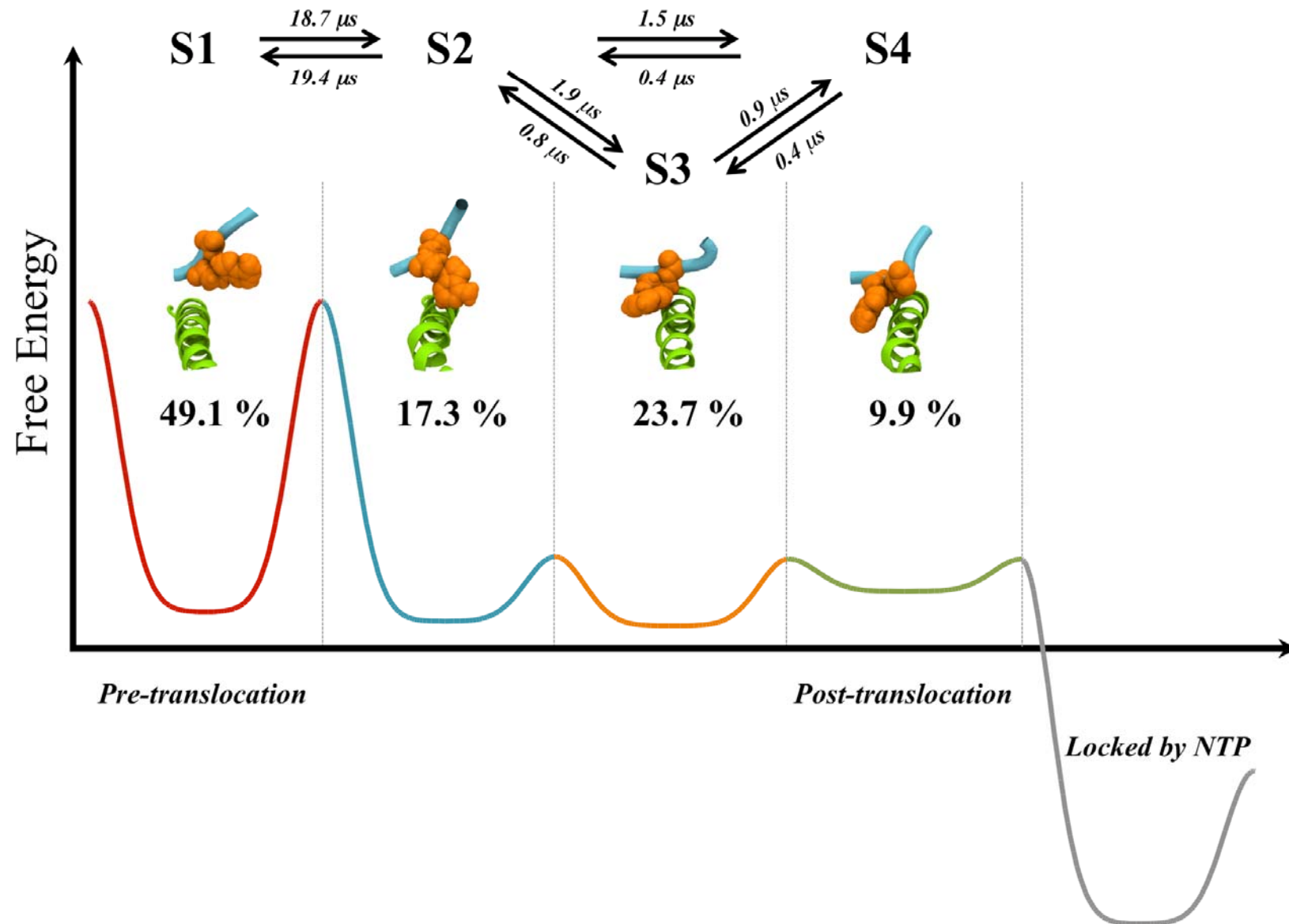
Four metastable states



Validation of the MSM



Thermodynamic and kinetic property



谢谢!

Two proposed backtracking models

Pre-, frayed and backtracked models

Climber algorithm

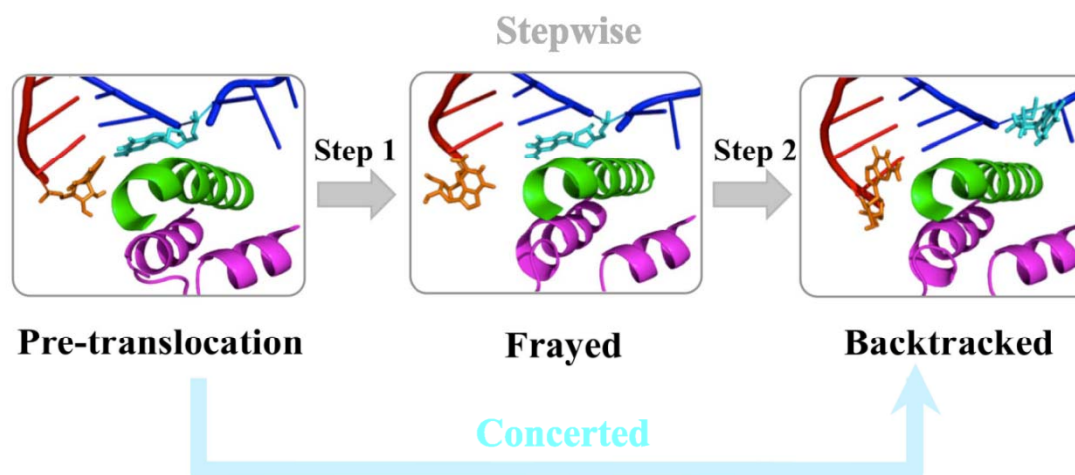
Conformations along low energy pathways

Unbiased MD simulations

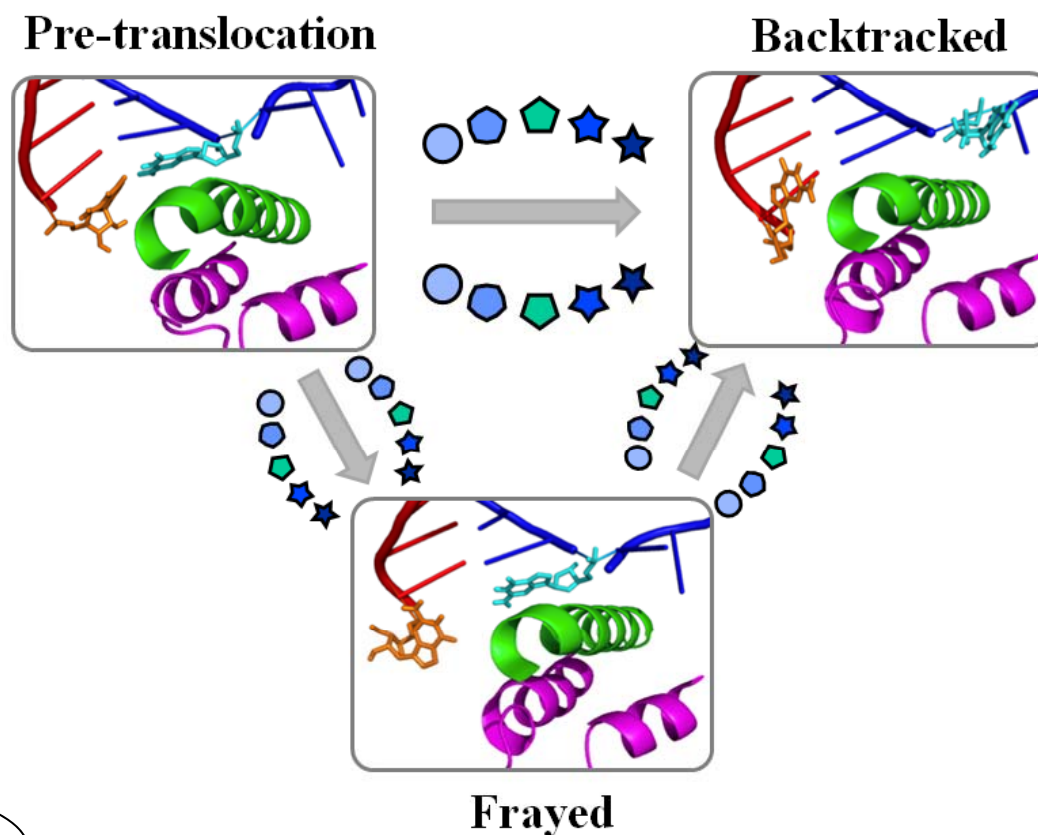
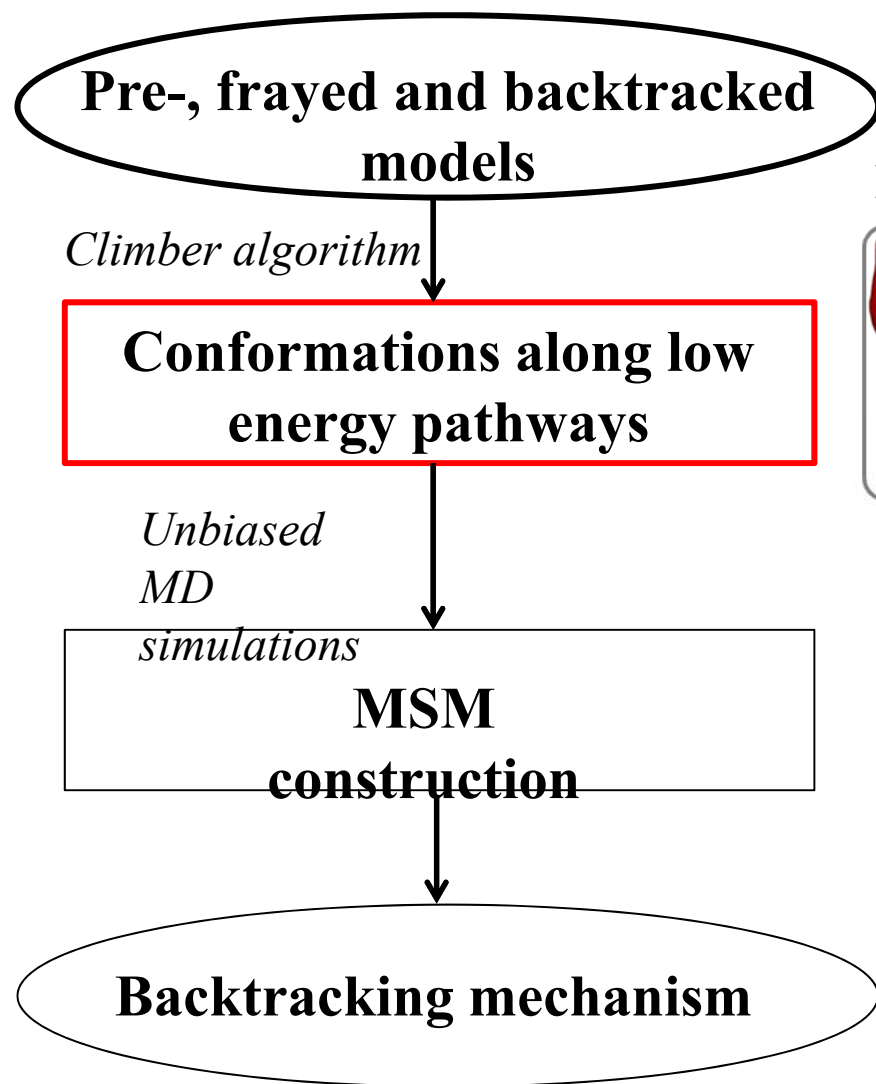
simulations

MSM construction

Backtracking mechanism

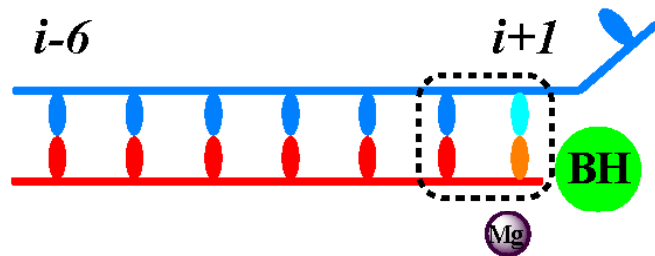


Initial low-energy backtracking pathways

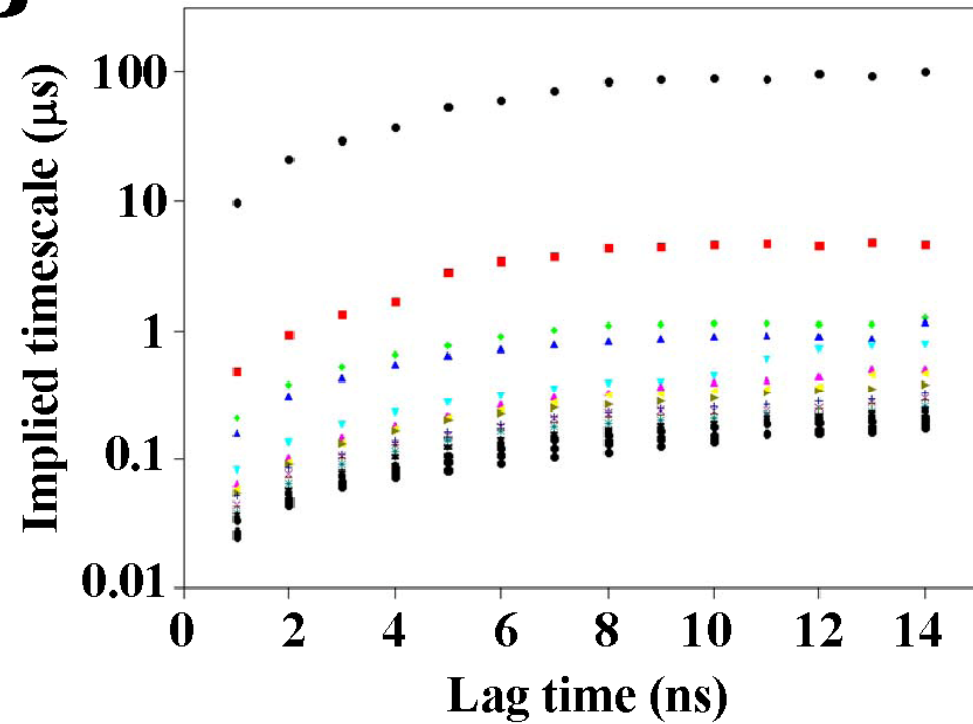


Validating the MSM

A



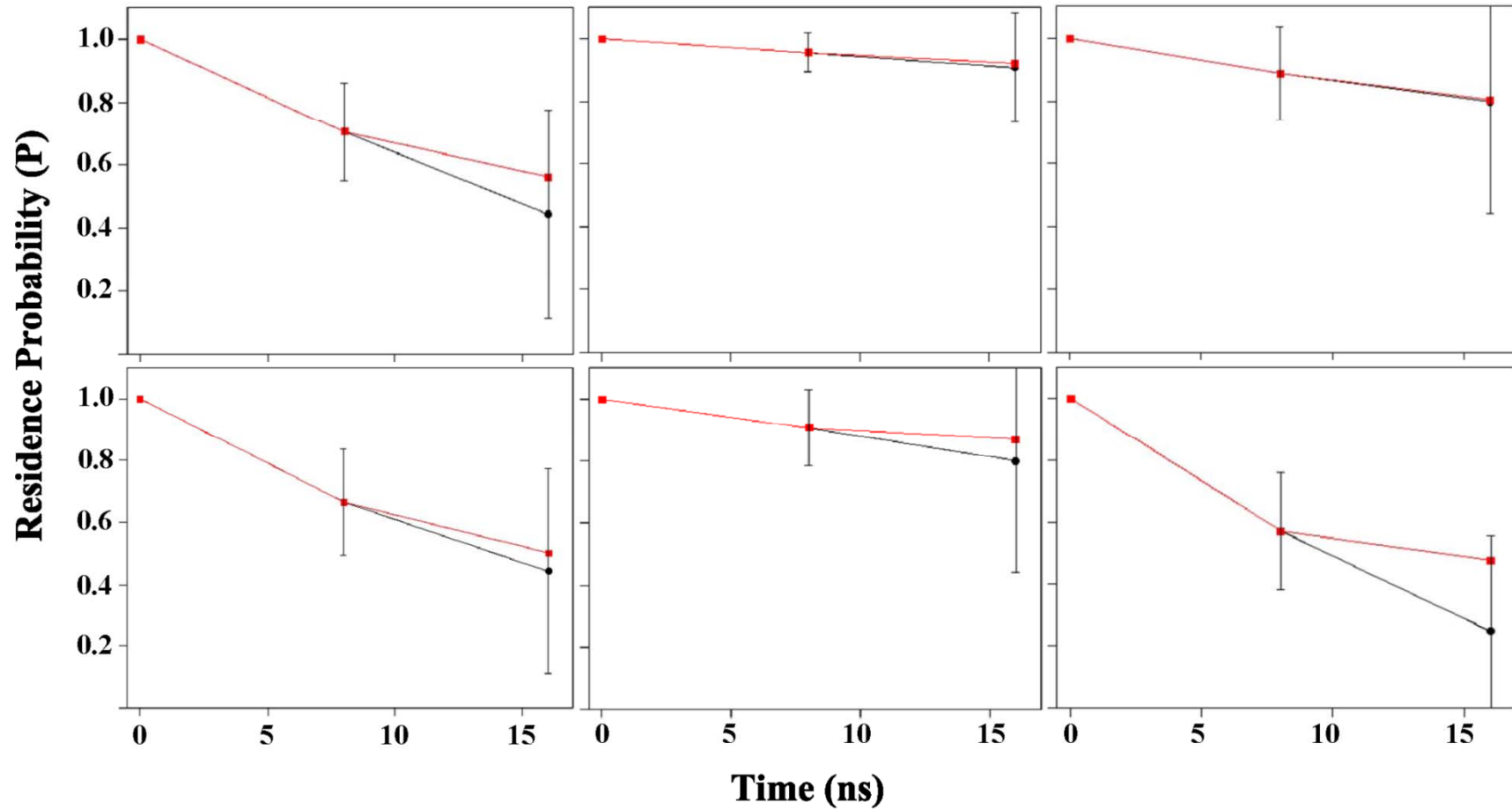
B



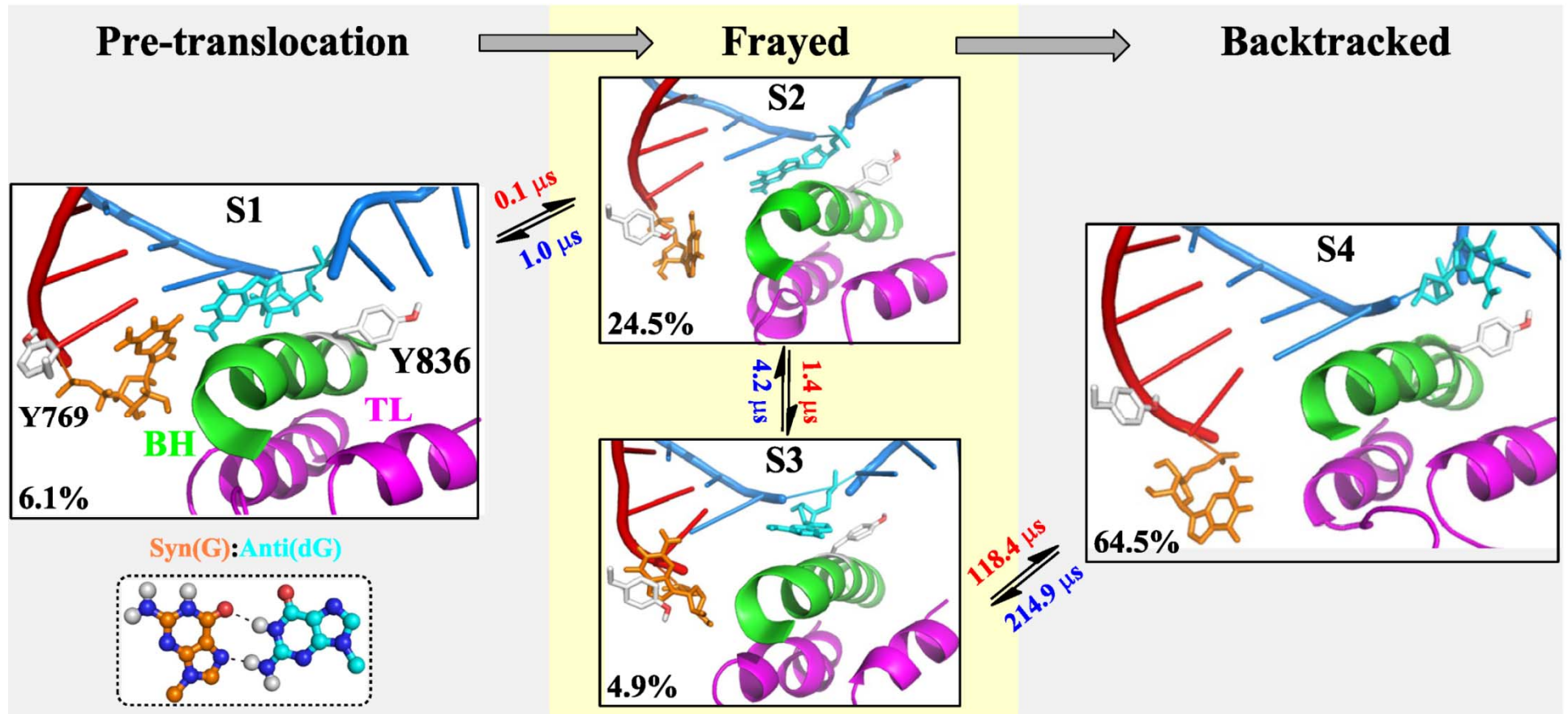
Chapman-Kolmogorov test

MSM

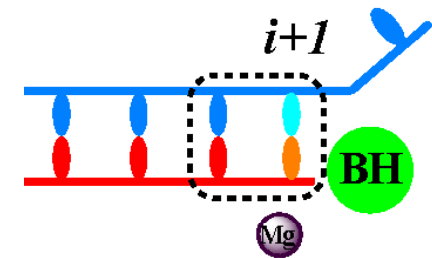
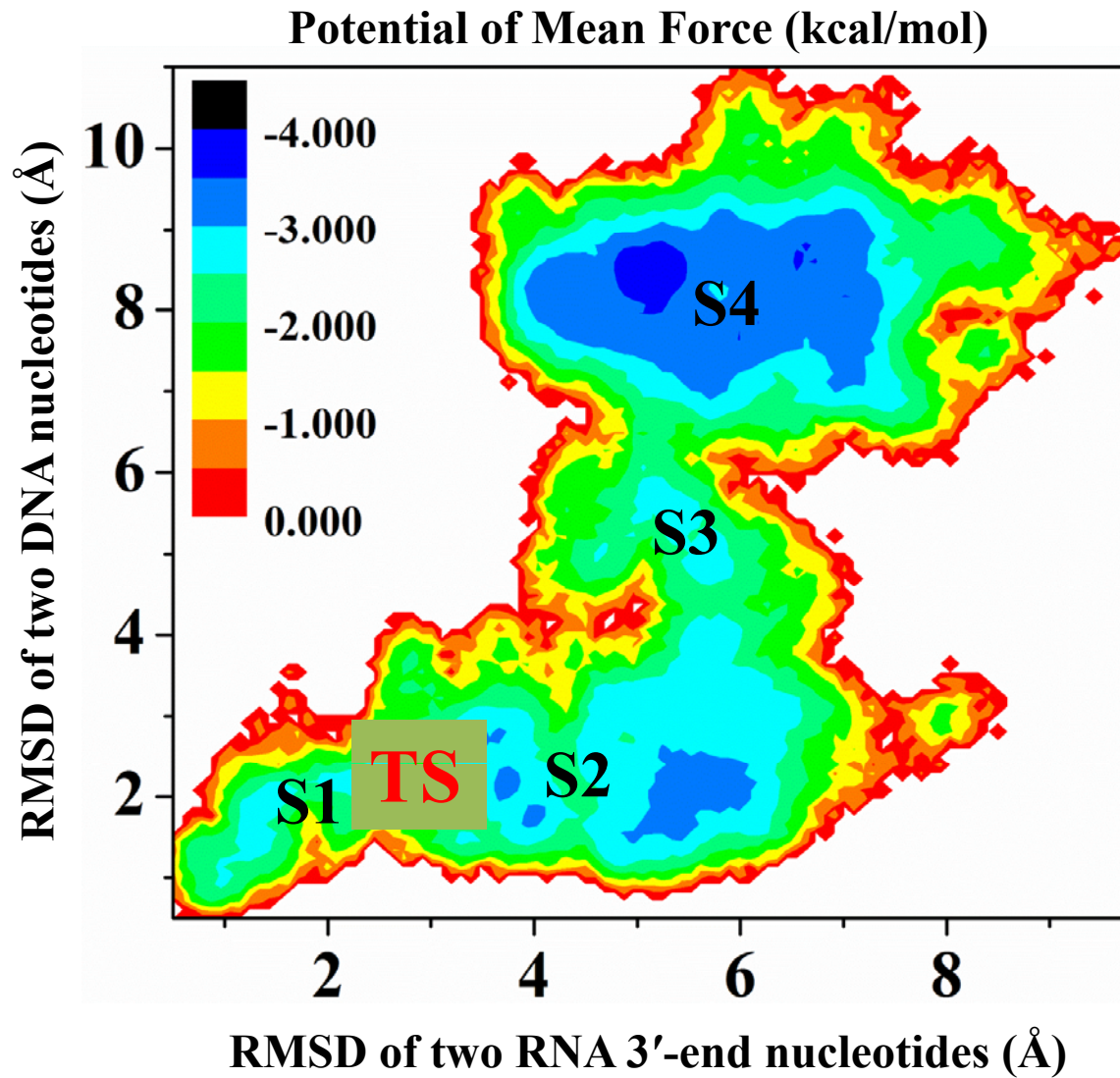
MD



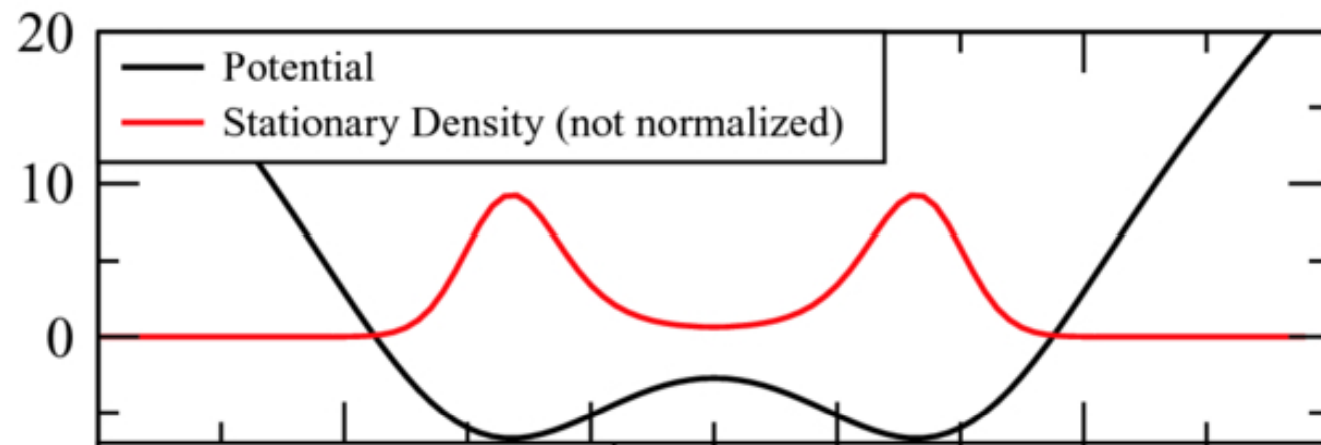
Four metastable states identified by MSM



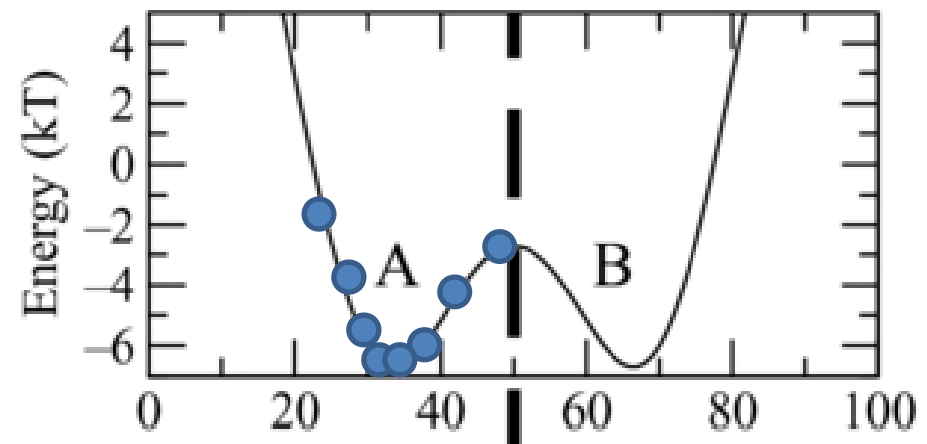
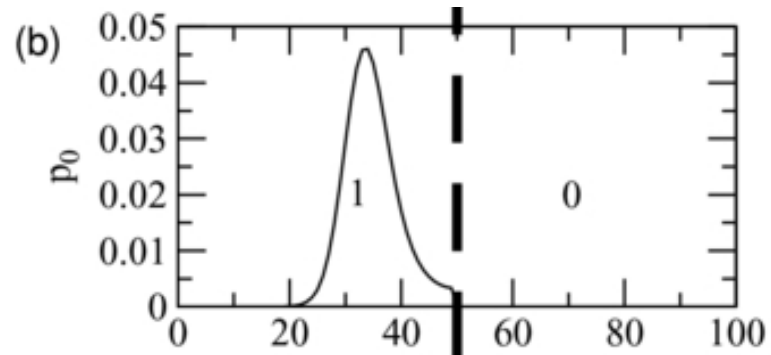
Backtracking preference



A Two-well Potential

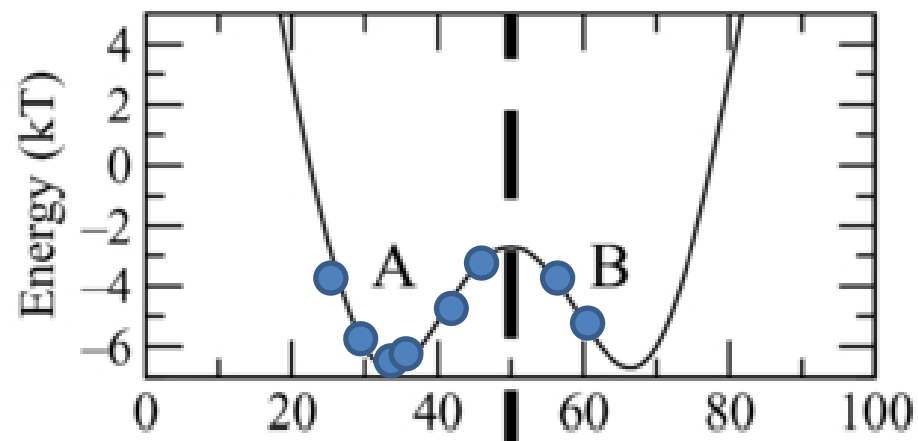
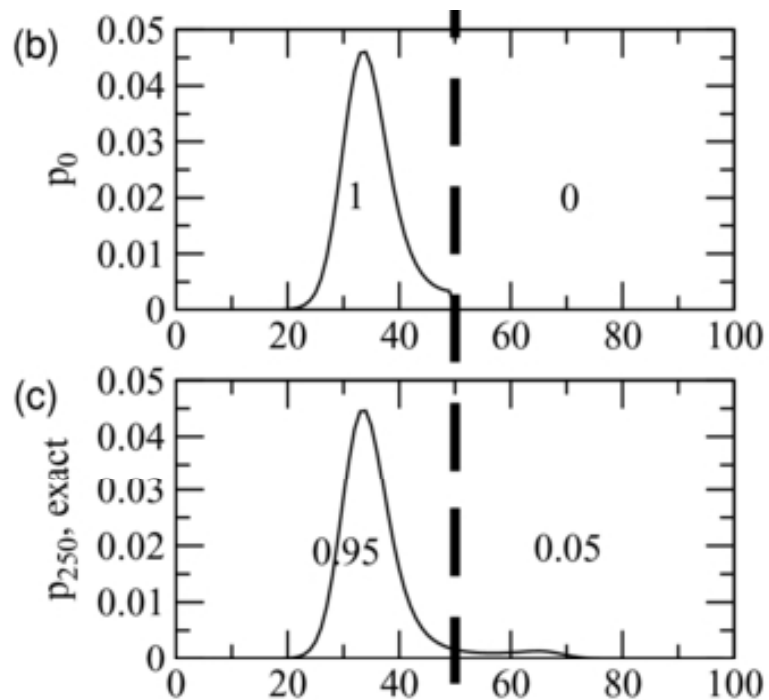


Propagation of Population

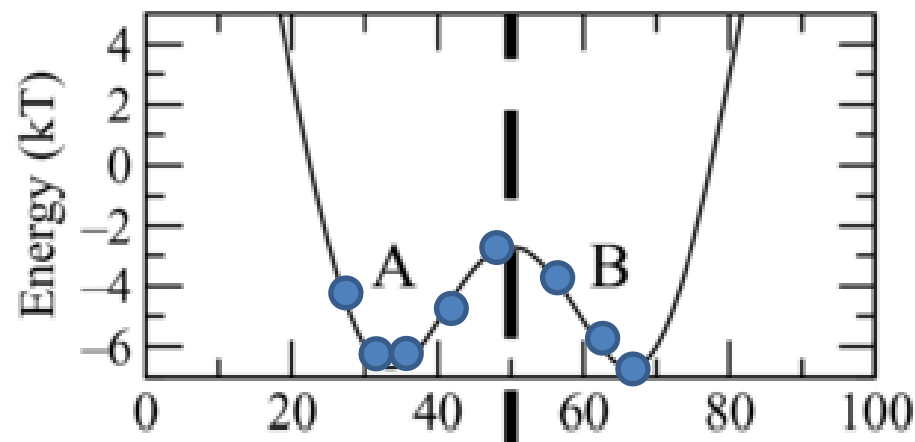
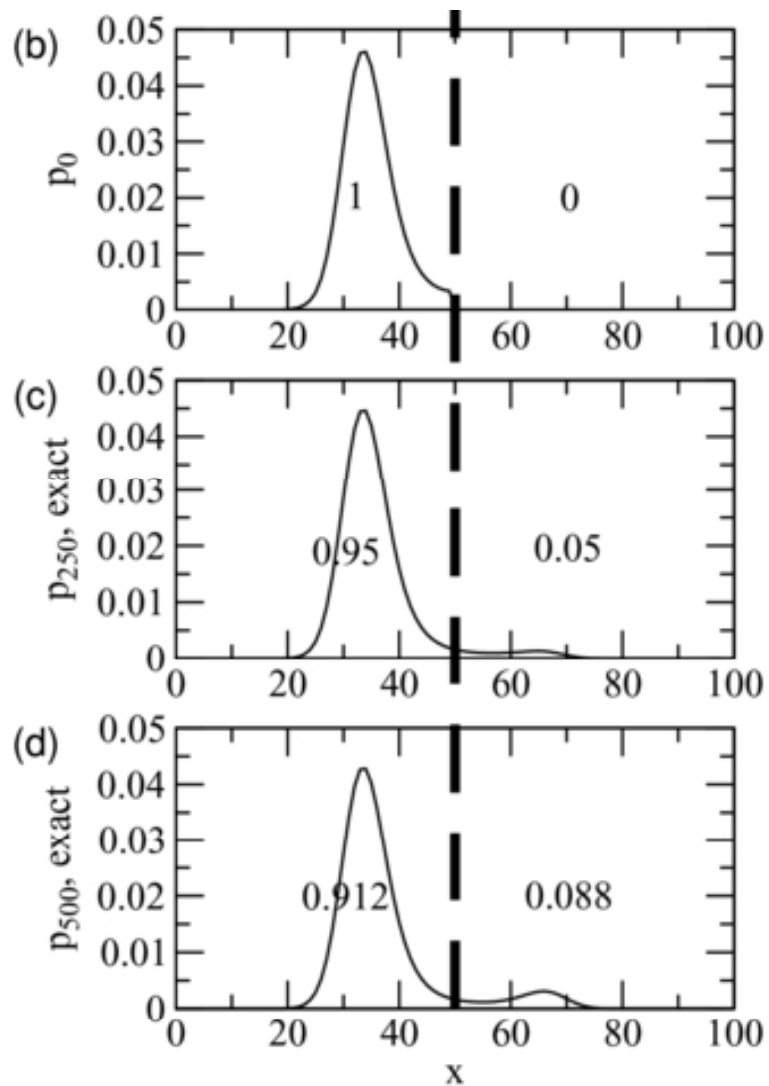


λ

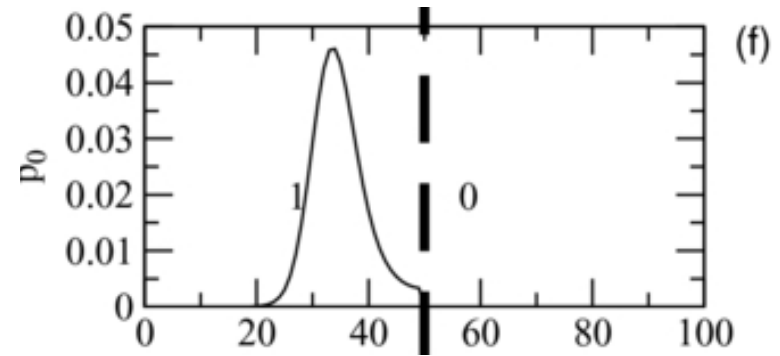
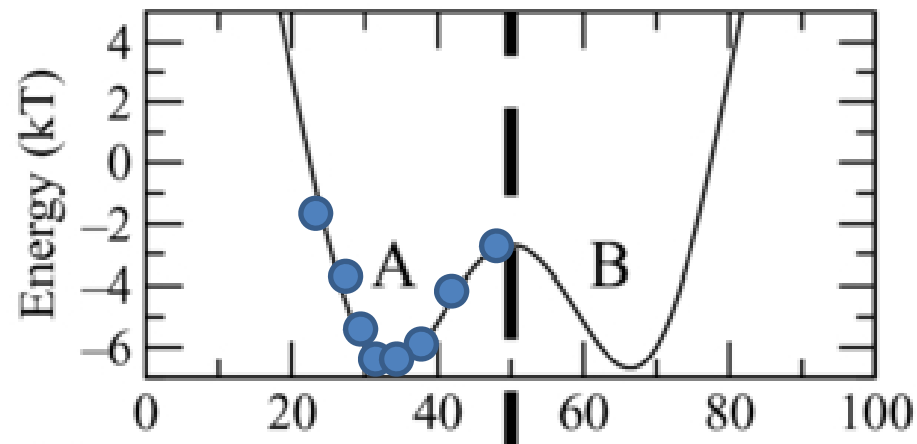
Propagation of Population



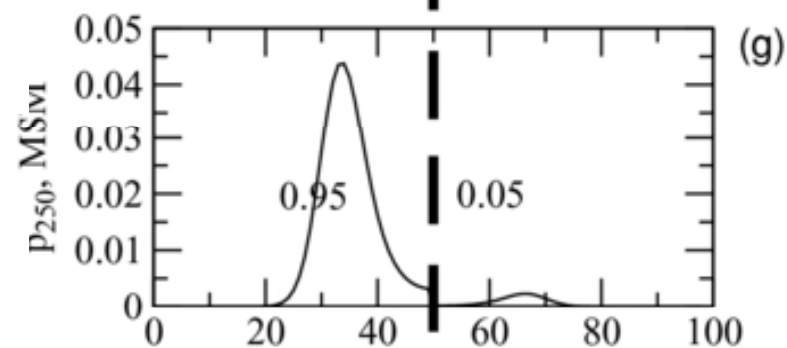
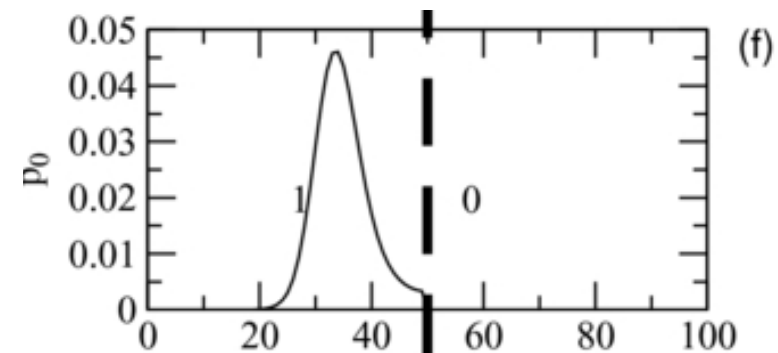
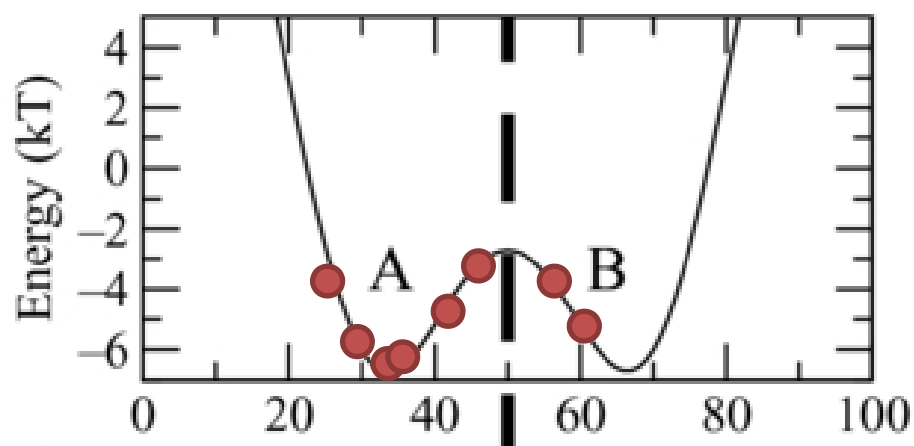
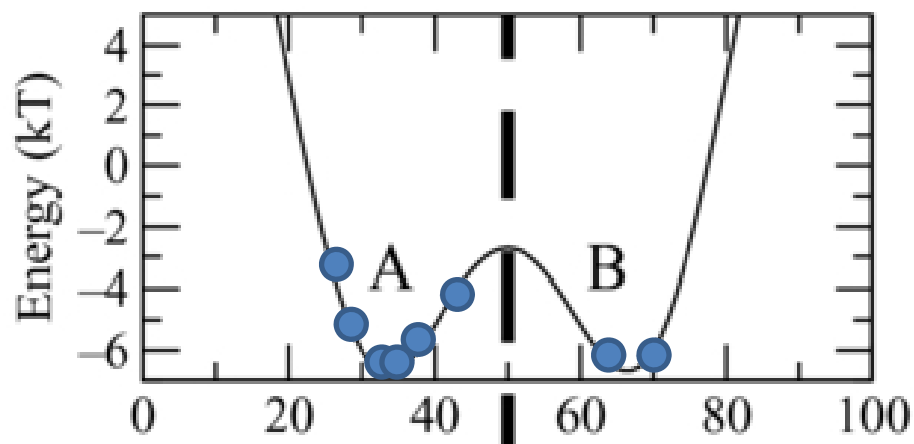
Propagation of Population



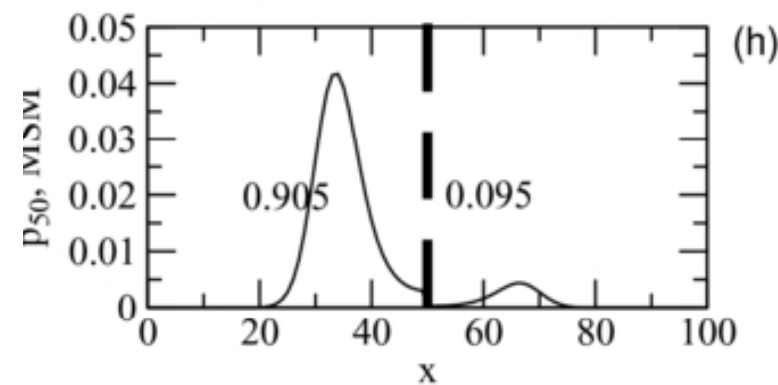
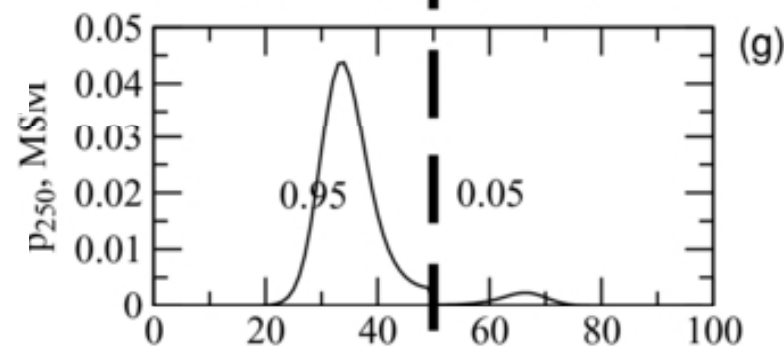
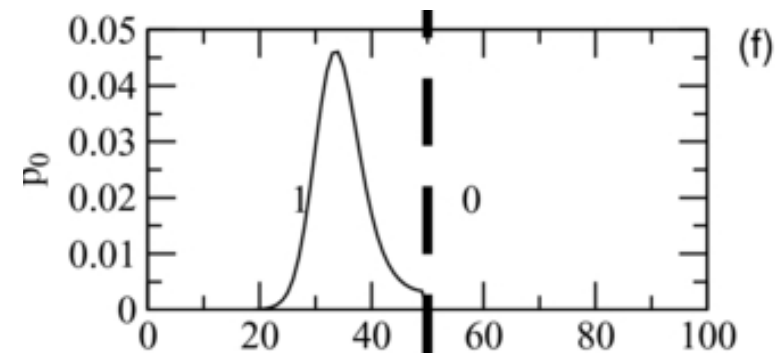
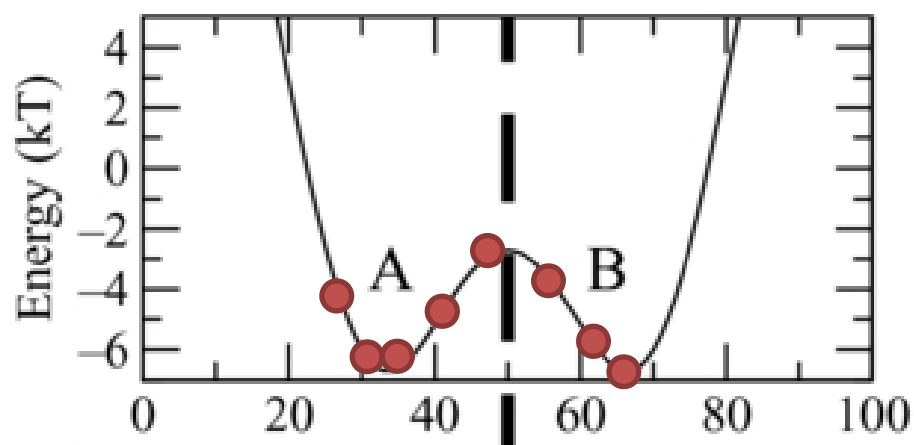
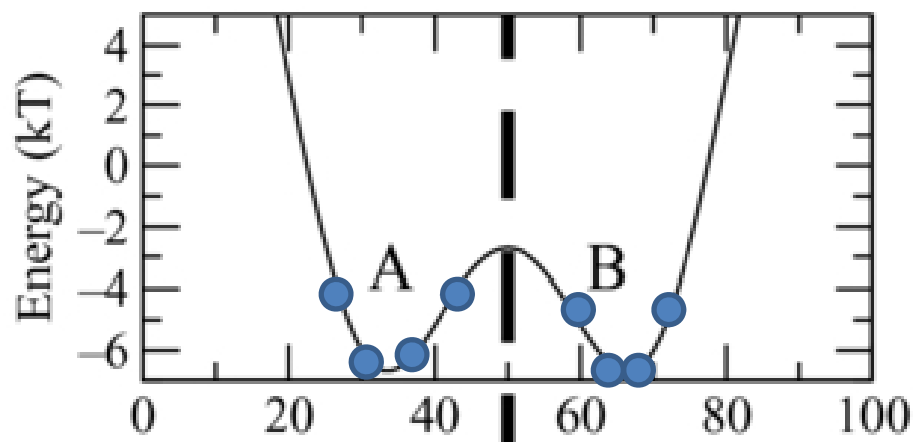
Propagation of Population



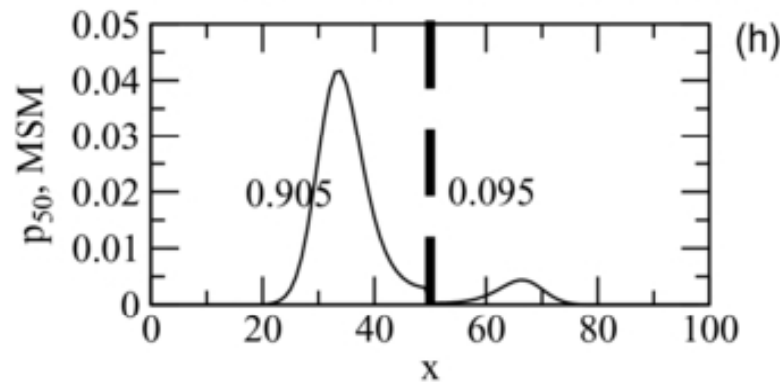
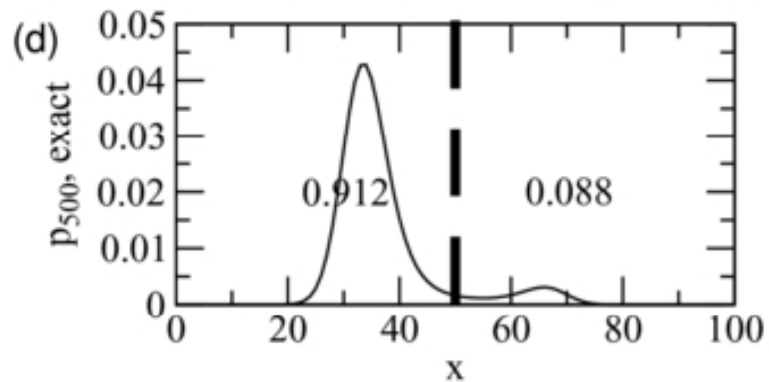
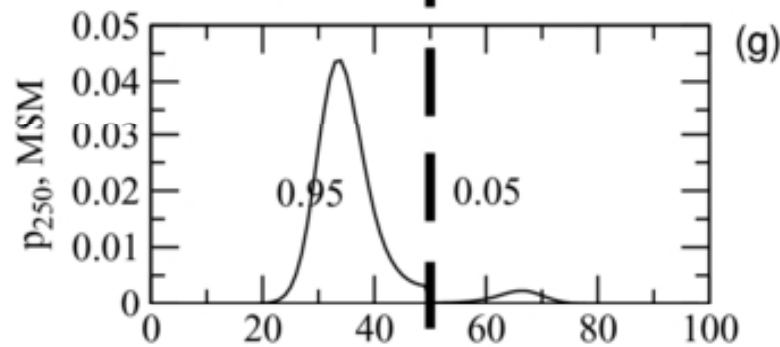
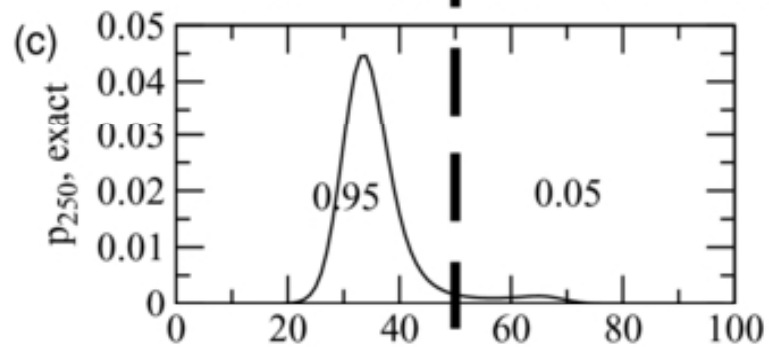
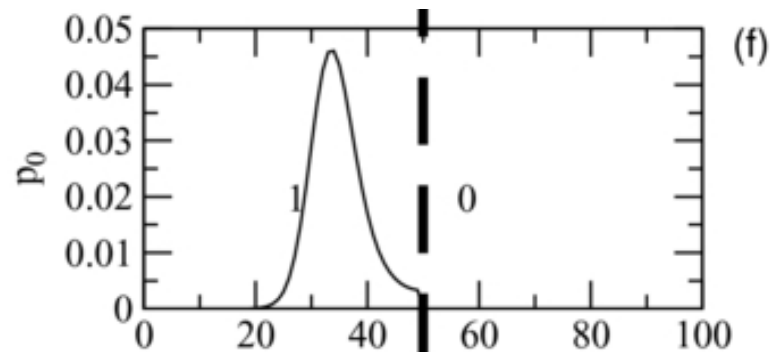
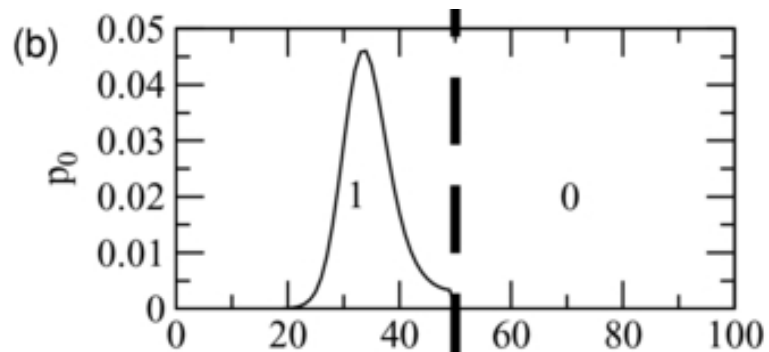
Propagation of Population



Propagation of Population



Propagation of Population



“Memory” vs Markovian Assumption

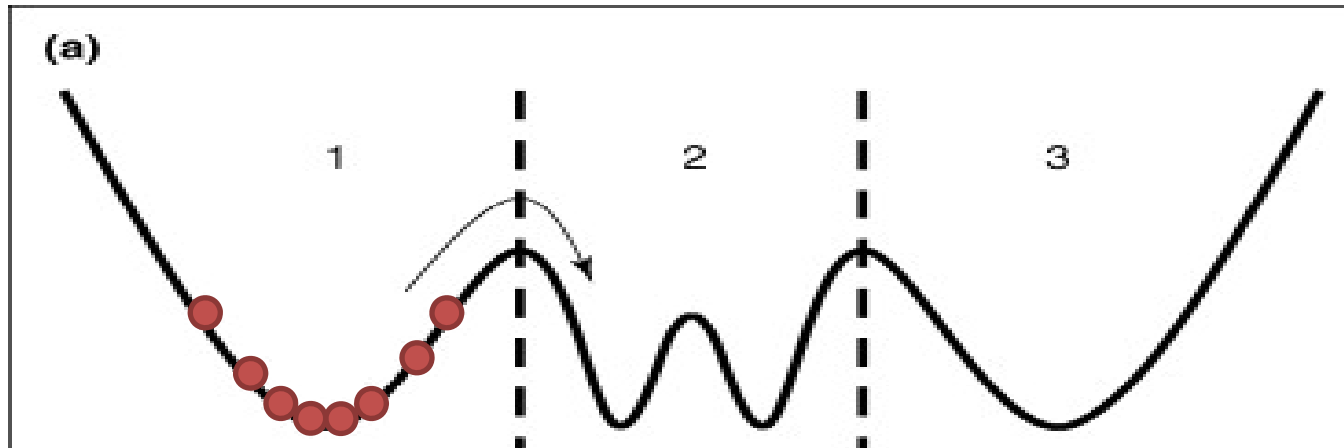
- Discretization introduces memory
- A Markov model is a “memoryless model” (or only have a finite amount of memory)
- The challenge of using MSM to analyze MD data is therefore minimizing the effect of “memory”

Problems with Very Long Lag Time

- Resources Issue
 - Recall: $n\tau \leq$ length of longest MD trajectory
 - Even for $n=1$, $\tau \leq$ length of longest MD trajectory
- Statistical Significance
 - Longer lag time give fewer transition counts
 - Problematic both for building TPM and validating the model

What else can we do?

- Consider:



Frank Noé, Stefan Fischer

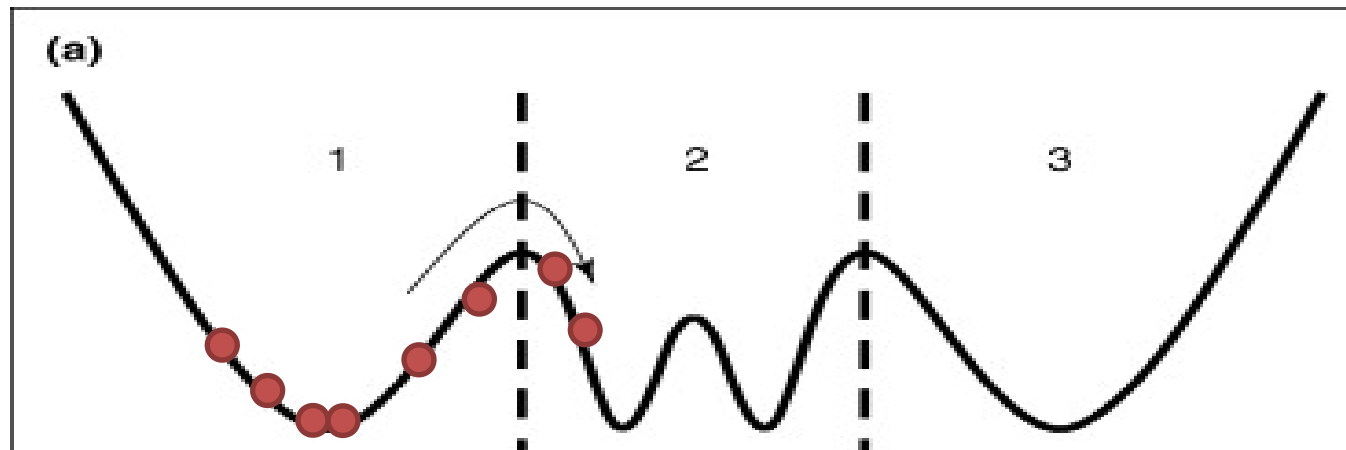
Transition networks for modeling the kinetics of conformational change in macromolecules

Current Opinion in Structural Biology Volume 18, Issue 2 2008 154 - 162

<http://dx.doi.org/10.1016/j.sbi.2008.01.008>

What else can we do?

- Consider:



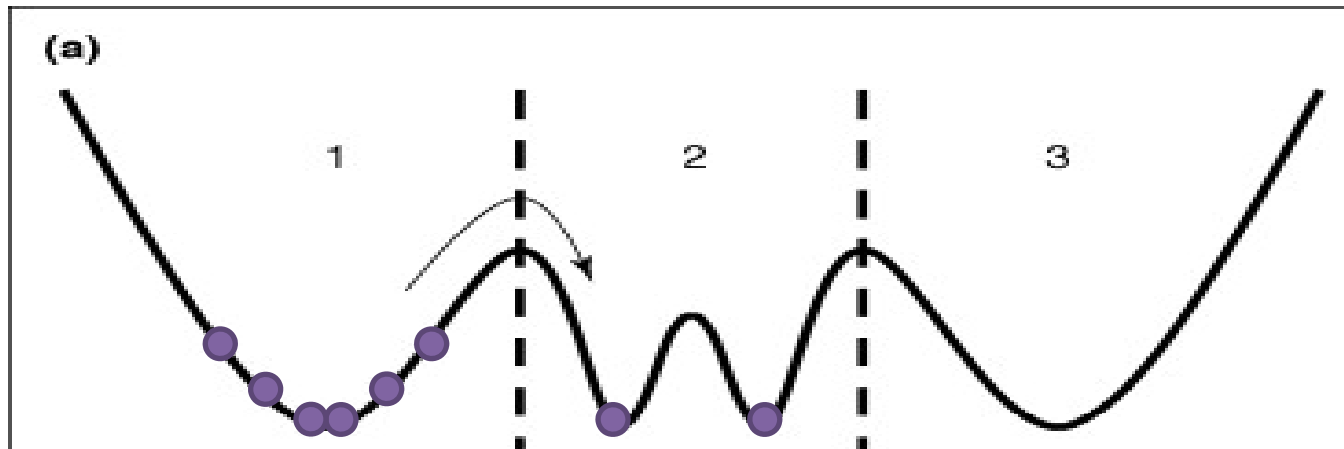
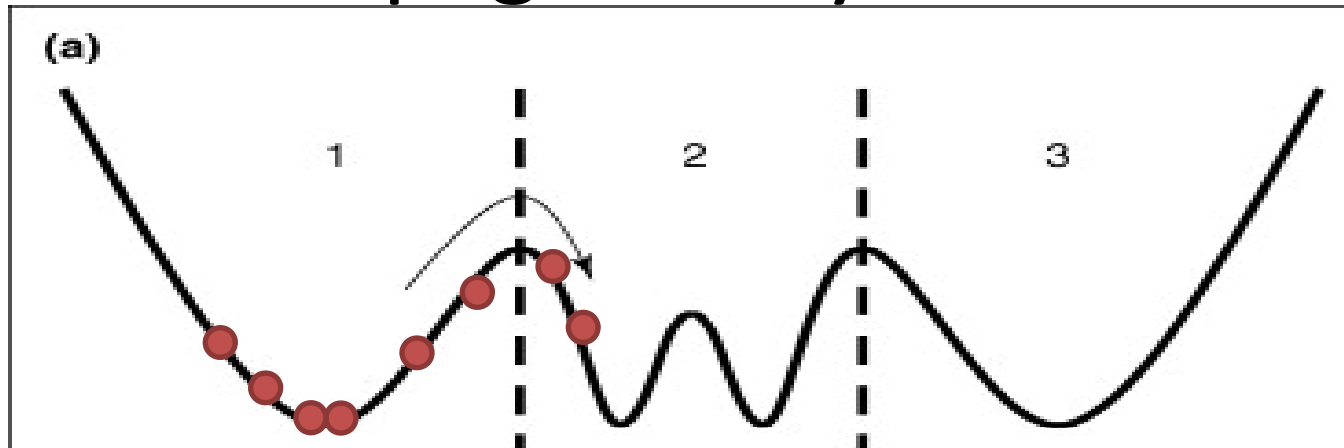
Frank Noé, Stefan Fischer

Transition networks for modeling the kinetics of conformational change in macromolecules

Current Opinion in Structural Biology Volume 18, Issue 2 2008 154 - 162

<http://dx.doi.org/10.1016/j.sbi.2008.01.008>

Difference Between Real and Propagated Dynamics



What else can we do?

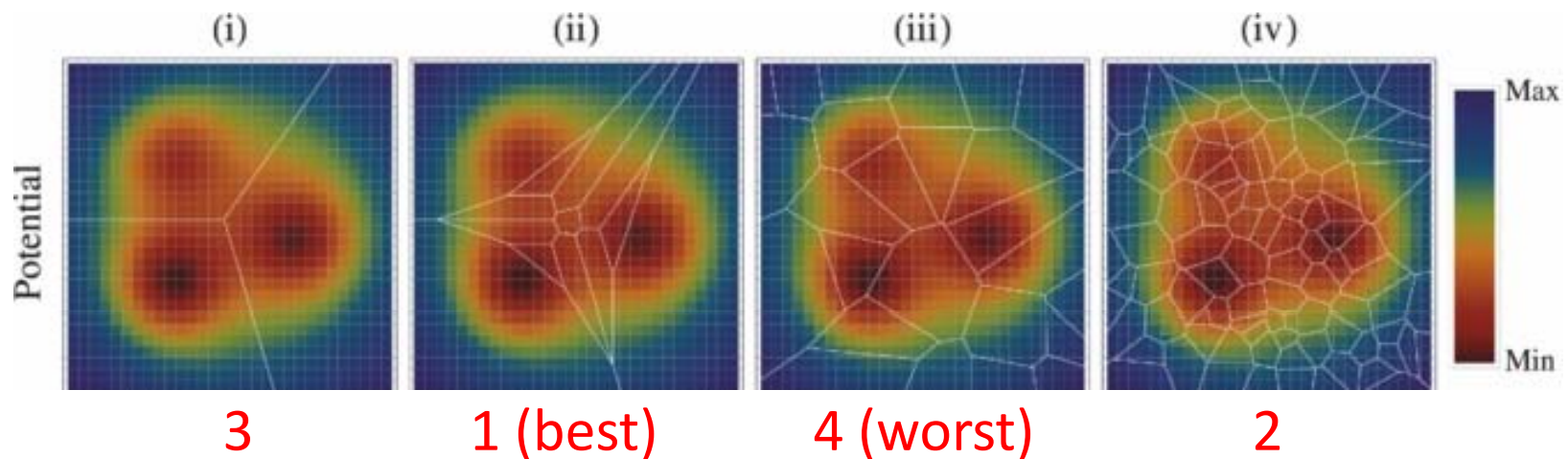
- Discretize the configuration space such that:
 - There is no internal barrier within all the states
 - In other words, always cut the very top of each barrier

Choose the Markovian Lag Time Under Our Commonly Used Framework

- Choosing the time such that the implied timescale plot of the slow modes plateaued
- The flattened implied timescale implies the populations are roughly “equilibrated” in such transition modes
- Validation via Chapman-Kolmogorov test is still necessary

Challenge Question

- Rank of the following discretization on how well each model represents the original energy landscape?



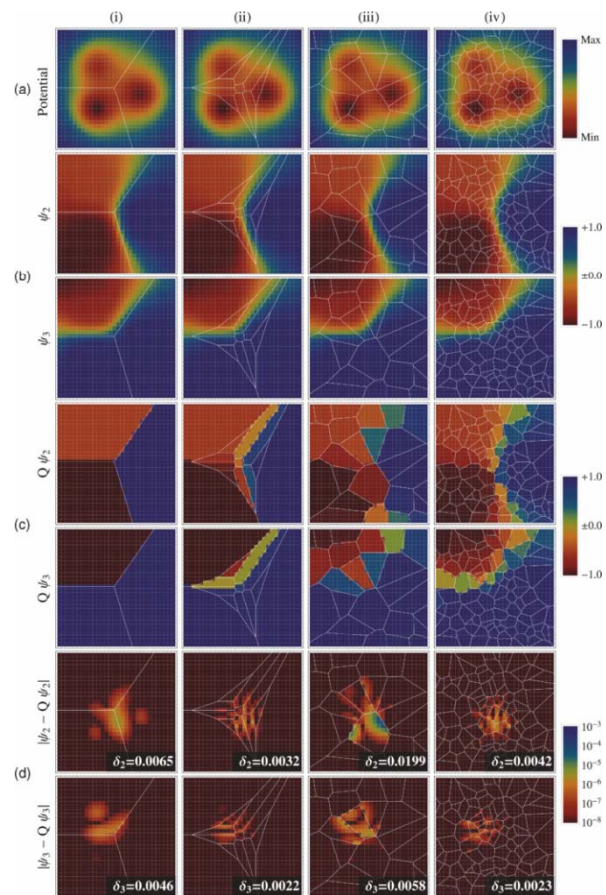


FIG. 6. Illustration of the eigenfunction approximation errors δ_2 and δ_3 on the two slowest processes in a two-dimensional three-well diffusion model [see supplementary material for model details (Ref.)]. The columns from left to right show different state space...

Observations

- The model with the most number of state is not the best
- The model with the least number of state is not the worst
- The model with highest metastability is not the best

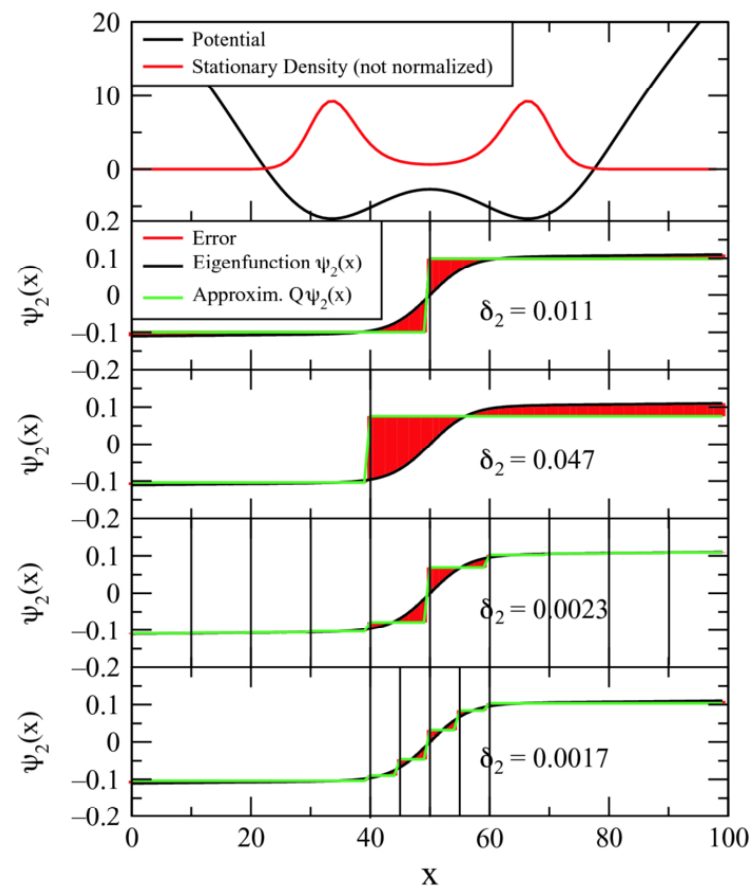
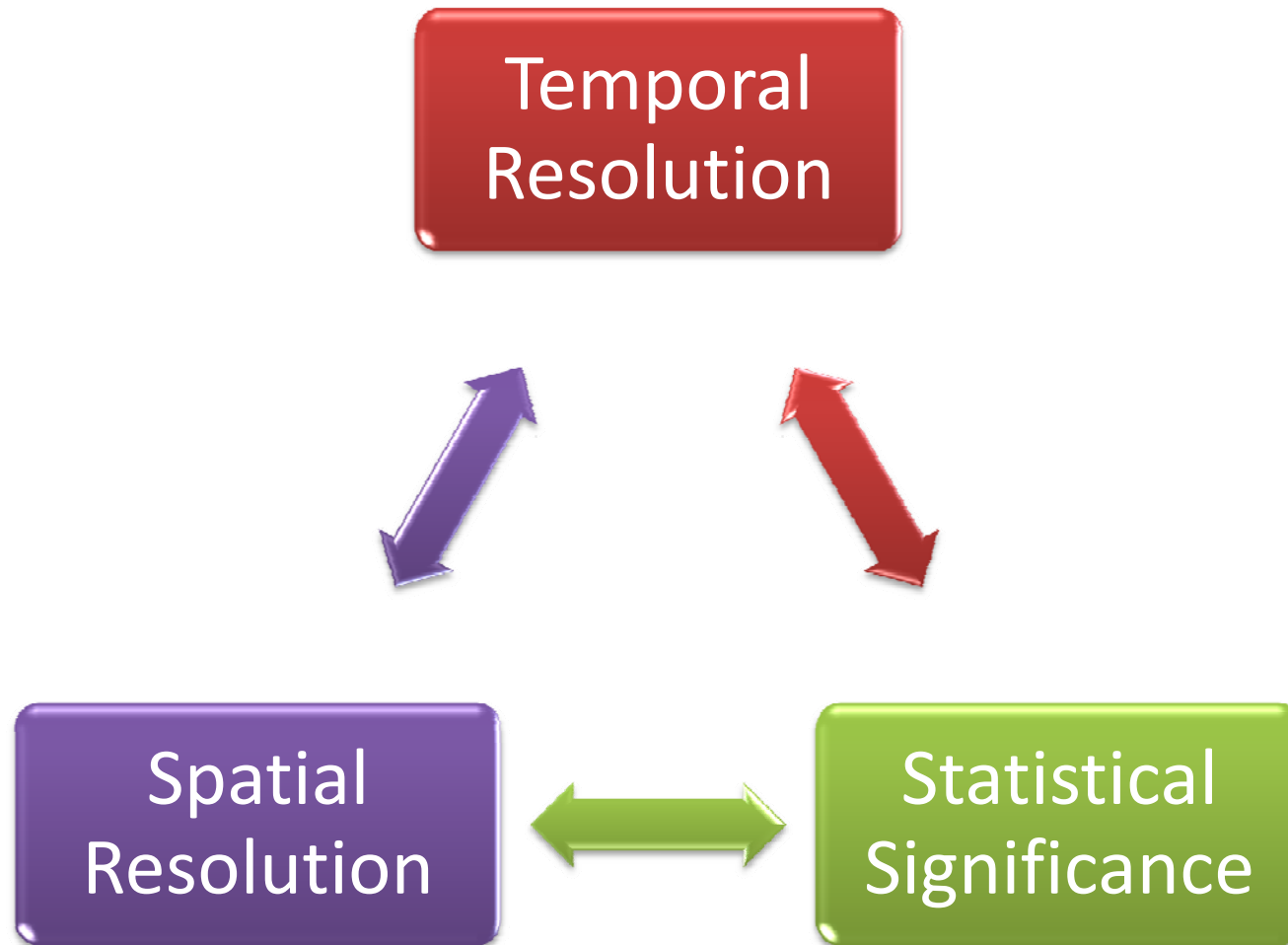


FIG. 5. Illustration of the eigenfunction approximation error δ_2 on the slow transition in the diffusion in a double well (top, black line). The slowest eigenfunction is shown in the lower four panels (black), along with the step approximations (green) of the par...

Compromise Between Factors



**Any questions on the
considerations we
have while building
an MSM?**

Flux Analysis

- What is the probability for a trajectory that starts from state A ends up in state B first before hitting state A again?

$$-q_i^+ + \sum_{k \in I} T_{ik} q_k^+ = - \sum_{k \in B} T_{ik}$$

$$f_{ij} = \pi_i q_i^- T_{ij} q_j^+$$

Flux Analysis

$$-q_i^+ + \sum_{k \in I} T_{ik} q_k^+ = - \sum_{k \in B} T_{ik}$$

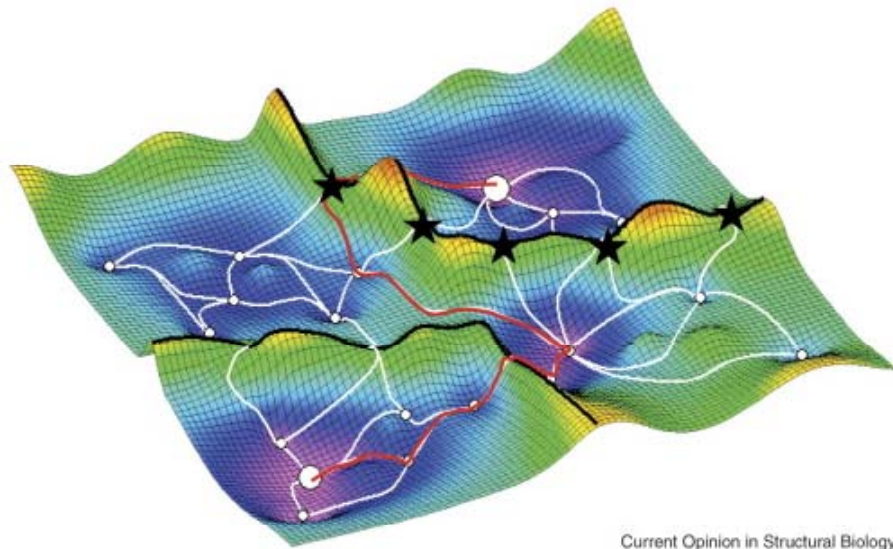
Forward committor probability
(Probability of hitting B first instead of A)

Transition probability from i to k

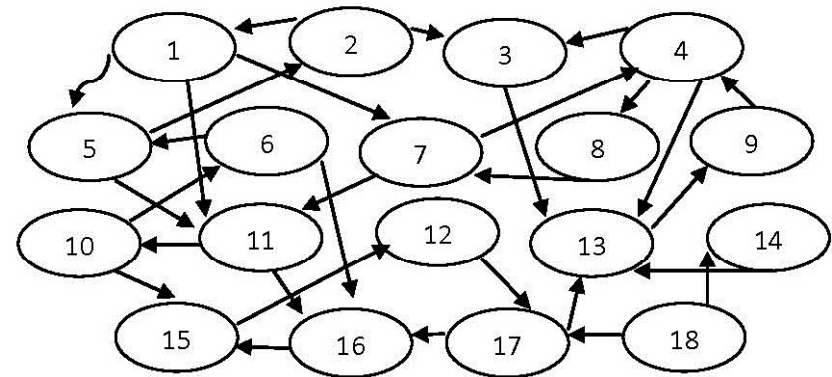
Sum only over states not belonging to A and B (intermediate states)

$$f_{ij} = \pi_i (1 - q_i^+) T_{ij} q_j^+$$

Approximating the Energy Landscape



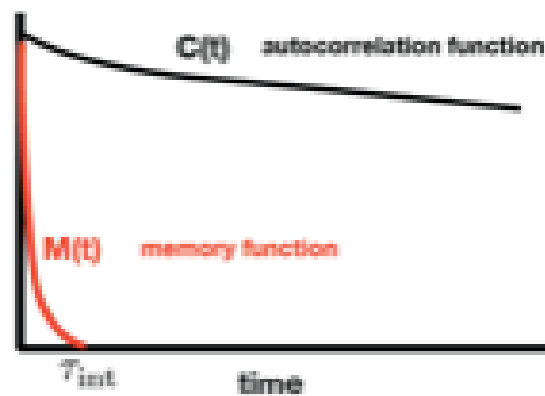
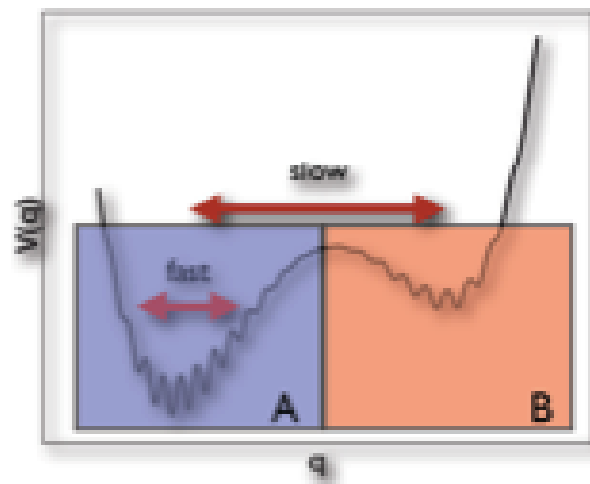
Current Opinion in Structural Biology



Minima on energy landscape represented as nodes in a kinetic scheme

F. Noe, S. Fischer, Transition networks for modeling the kinetics of conformational change in macromolecules. *Current opinion in structural biology* **18**, 154 (Apr, 2008).
http://upload.wikimedia.org/wikipedia/commons/f/fc/Kinetic_scheme.jpg

MSMs Can Describe Conformational dynamics by coarse-grained time and space



- *Coarse-graining of space* into discrete states will introduce memory:

$$\dot{C}(t) = - \int_0^t dt' M(t-t') C(t')$$

- *Coarse-graining of time* can make the memory appear short if there is **separation of timescales**

$$\dot{C}(t) = \left[- \int_0^\infty dt' M(t-t') \right] C(t) = \mathbf{K}C(t)$$

- This resembles a memoryless discrete-state discrete-time Markov model

$$P(n\tau) = [T(\tau)]^n P(0)$$

Figure Courtesy: John Chodera

Zwanzig, J Stat Phys 30:255, 1983.

Linear combination of the eigenvectors

$$P(n\tau) = [T(\tau)]^n P(0) = [T(\tau)]^n \sum_{i=1} c_i \Phi_i = \sum_{i=1} c_i [T(\tau)]^n \Phi_i = \sum_{i=1} c_i \lambda_i^n \Phi_i$$

Implied timescale

$$T(\tau) = LDL^{-1} \quad T(\tau)^n = LD^n L^{-1} \quad t = n\tau$$

$$\mu^n = \mu^{t/\tau} \quad \mu^{t/\tau} = e^{\ln \mu^{t/\tau}}$$

According to the exponential decay constant:

$$N(t) = N_0 e^{-\lambda t}$$

$$\text{Mean lifetime: } \tau_{decay} = \frac{1}{\lambda} = -\frac{\tau}{\ln \mu}$$

Mean first passage time (MFPT): the mean time it takes to reach a given metastable state m for the first time when starting from another state i .

$$\begin{aligned}
 f_{1m} &= \tau + T_{11}f_{1m} + T_{12}f_{2m} + \cdots + T_{1m}f_{mm} & -\tau &= -f_{1m} + T_{11}f_{1m} + T_{12}f_{2m} + \cdots + T_{1m}f_{mm} \\
 f_{2m} &= \tau + T_{21}f_{1m} + T_{22}f_{2m} + \cdots + T_{2m}f_{mm} & -\tau &= -f_{2m} + T_{21}f_{1m} + T_{22}f_{2m} + \cdots + T_{2m}f_{mm}
 \end{aligned}$$

$$\begin{bmatrix}
 T_{11}^{-1} & & \cdots & & T_{1m} \\
 \vdots & T_{22}^{-1} & & & T_{2m} \\
 & & \ddots & & \\
 T_{m1} & \cdots & & T_{m-1,m-1}^{-1} & T_{m-1,m} \\
 0 & 0 & \cdots & 0 & 1
 \end{bmatrix}
 \times
 \begin{bmatrix}
 f_{1m} \\
 f_{2m} \\
 \vdots \\
 f_{m-1} \\
 f_{mm}
 \end{bmatrix}
 =
 \begin{bmatrix}
 -\tau \\
 -\tau \\
 \vdots \\
 -\tau \\
 -\tau
 \end{bmatrix}$$

Chapman-Kolmogorov Equation

- For a discrete-state time-homogenous model that assumes Markovian property, we have the Chapman-Kolmogorov Equation:

$$P(t + s) = P(t)P(s) \quad \begin{array}{l} P(\tau + \tau) = [P(\tau)]^2 \\ P(n\tau) = [P(\tau)]^n \end{array}$$

- Thus, for a Markovian model that well-represents the original system, we should have:

$$P_{MD}(n\tau) = P_{MSM}(n\tau) = [P_{MSM}(\tau)]^n$$

well-represents Markovian

Chapman-Kolmogorov Test

- Chapman-Kolmogorov Test

$$P_{MD}(n\tau) = [P_{MSM}(\tau)]^n$$

–holds within the margin of error

A “Markovian”
model that “well-
represents” MD



- Any tests that follows such procedure can be considered as “Chapman-Kolmogorov test”

A Commonly Used Implementation

$$P_{MD}(n\tau) = [P_{MSM}(\tau)]^n$$

$$\begin{bmatrix} P_{11,MD}(n\tau) & P_{12,MD}(n\tau) & \dots & P_{1k,MD}(n\tau) \\ P_{21,MD}(n\tau) & P_{22,MD}(n\tau) & \dots & P_{2k,MD}(n\tau) \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MD}(n\tau) & P_{k2,MD}(n\tau) & \dots & P_{kk,MD}(n\tau) \end{bmatrix} = \begin{bmatrix} P_{11,MSM}(\tau) & P_{12,MSM}(\tau) & \dots & P_{1k,MSM}(\tau) \\ P_{21,MSM}(\tau) & P_{22,MSM}(\tau) & \dots & P_{2k,MSM}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MSM}(\tau) & P_{k2,MSM}(\tau) & \dots & P_{kk,MSM}(\tau) \end{bmatrix}^n$$

Transition probabilities
at time $n\tau$ counted
from MD trajectories

**Note: $n\tau \leq$ length of
longest MD trajectory**

(TPM of the MSM)ⁿ

If a complete dataset is used, the equality should hold strictly at $n = 0$ and $n = 1$

Testing the Dataset

$$P_{MD}(n\tau) = [P_{MSM}(\tau)]^n$$

$$IP_{MD}(n\tau) = I[P_{MSM}(\tau)]^n$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} P_{11,MD} & P_{12,MD} & \dots & P_{1k,MD} \\ P_{21,MD} & P_{22,MD} & \dots & P_{2k,MD} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MD} & P_{k2,MD} & \dots & P_{kk,MD} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} P_{11,MSM} & P_{12,MSM} & \dots & P_{1k,MSM} \\ P_{21,MSM} & P_{22,MSM} & \dots & P_{2k,MSM} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MSM} & P_{k2,MSM} & \dots & P_{kk,MSM} \end{bmatrix}^n$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} P_{11,MD} & P_{12,MD} & \dots & P_{1k,MD} \\ P_{21,MD} & P_{22,MD} & \dots & P_{2k,MD} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MD} & P_{k2,MD} & \dots & P_{kk,MD} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} P_{11,MSM} & P_{12,MSM} & \dots & P_{1k,MSM} \\ P_{21,MSM} & P_{22,MSM} & \dots & P_{2k,MSM} \\ \vdots & \vdots & \ddots & \vdots \\ P_{k1,MSM} & P_{k2,MSM} & \dots & P_{kk,MSM} \end{bmatrix}^n$$

And so is other (k-1) equalities

Accounting for Uncertainties in Simulation

- One-sigma Standard Error

$$\epsilon_{\text{MD}}(A, A; k\tau) = \sqrt{k \frac{p_{\text{MD}}(A, A; k\tau) - [p_{\text{MD}}(A, A; k\tau)]^2}{\sum_{i \in A} \sum_{j=1}^n c_{ij}^{\text{obs}}(k\tau)}}$$

- Bootstrapping

Example: a 5-state model

- State 1 • State 2 • State 3 • State 4 • State 5

Lagtime $\tau = 100\text{ps}$

For each t there are $2k$ points:

- k with error bars
- k on solid lines

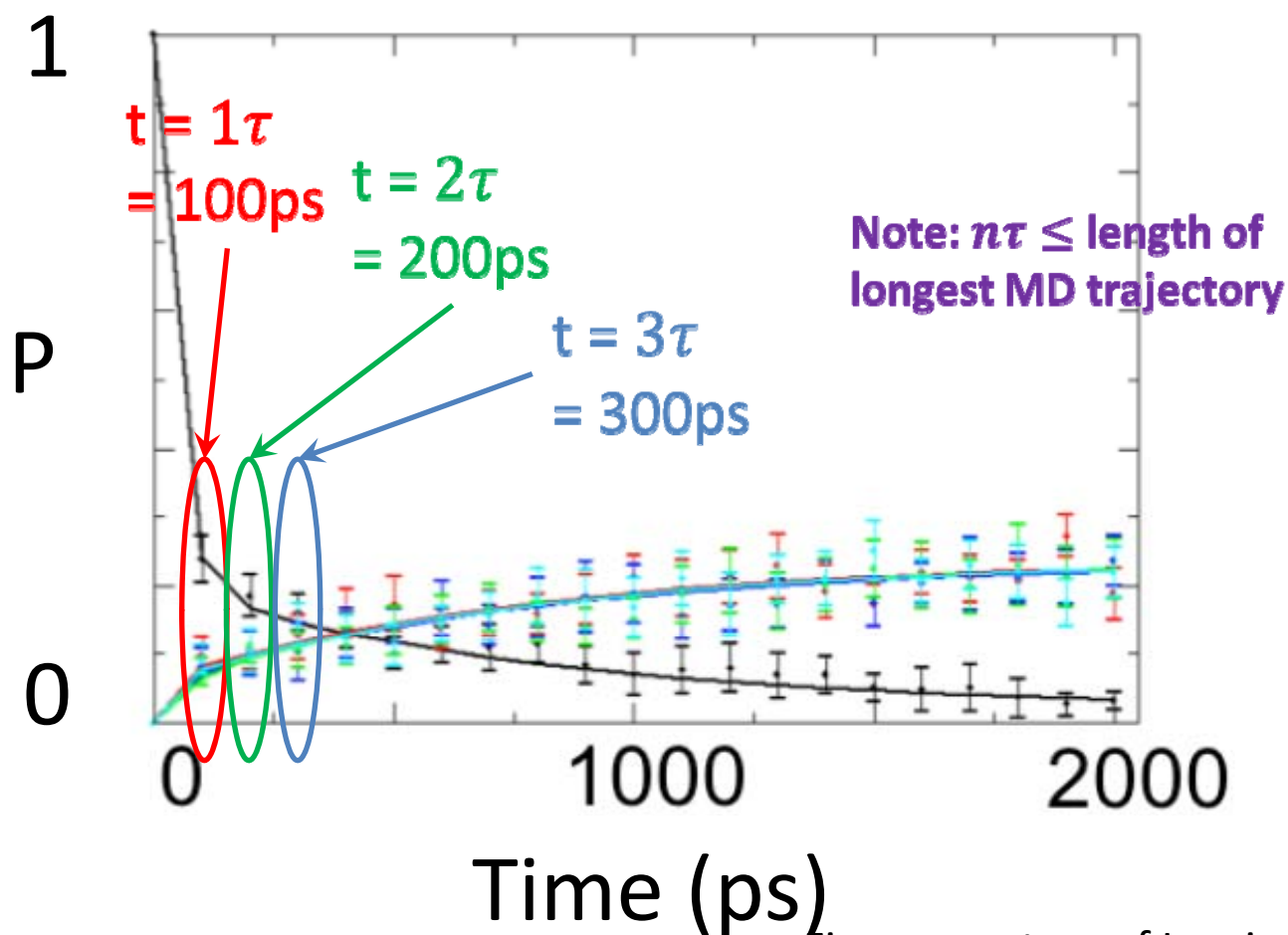


Figure courtesy of Luming

Example: a 5-state model

- State 1 • State 2 • State 3 • State 4 • State 5

$$IP_{MD}(n\tau) = I[P_{MSM}(\tau)]^n$$

Points with Error Bar:

$$[0 \ 0 \ 0 \ 0 \ 1] \begin{bmatrix} P_{11,MD} & P_{12,MD} & \dots & P_{15,MD} \\ P_{21,MD} & P_{22,MD} & \dots & P_{25,MD} \\ \vdots & \vdots & \ddots & \vdots \\ P_{51,MD} & P_{52,MD} & \dots & P_{55,MD} \end{bmatrix}$$

Points on solid line:

$$[0 \ 0 \ 0 \ 0 \ 1] \begin{bmatrix} P_{11,MSM} & P_{12,MSM} & \dots & P_{15,MSM} \\ P_{21,MSM} & P_{22,MSM} & \dots & P_{25,MSM} \\ \vdots & \vdots & \ddots & \vdots \\ P_{51,MSM} & P_{52,MSM} & \dots & P_{55,MSM} \end{bmatrix}$$

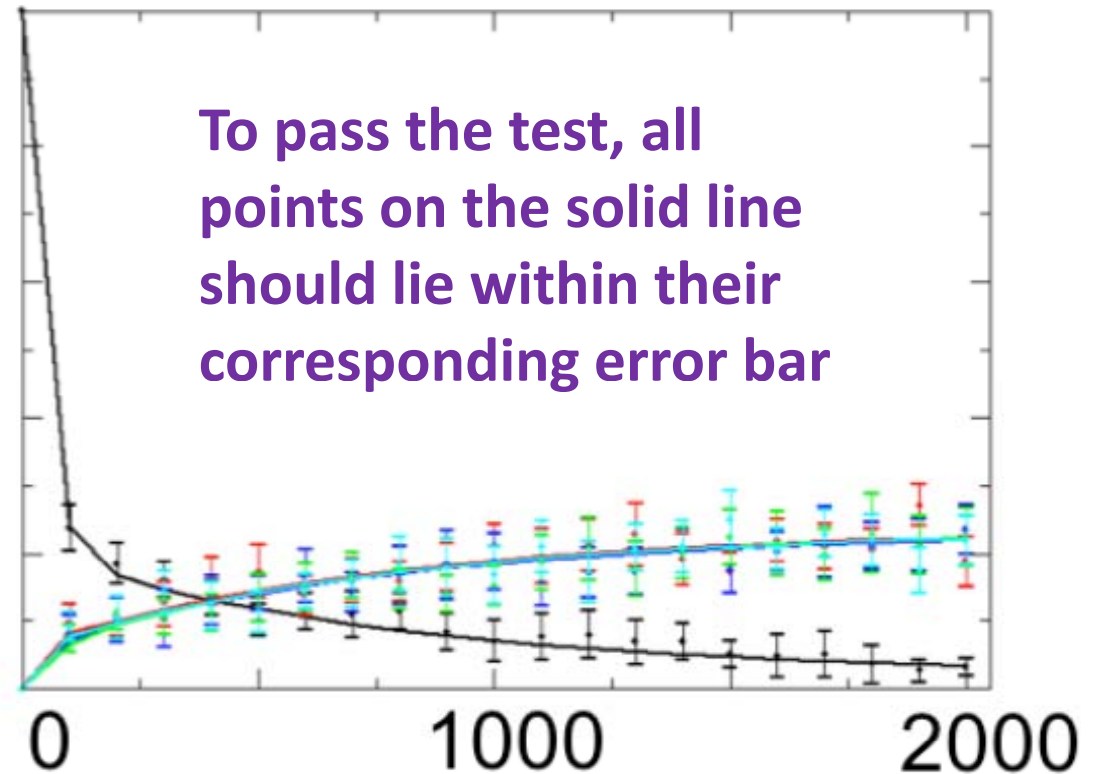


Figure courtesy of Luming

Pass or Fail?

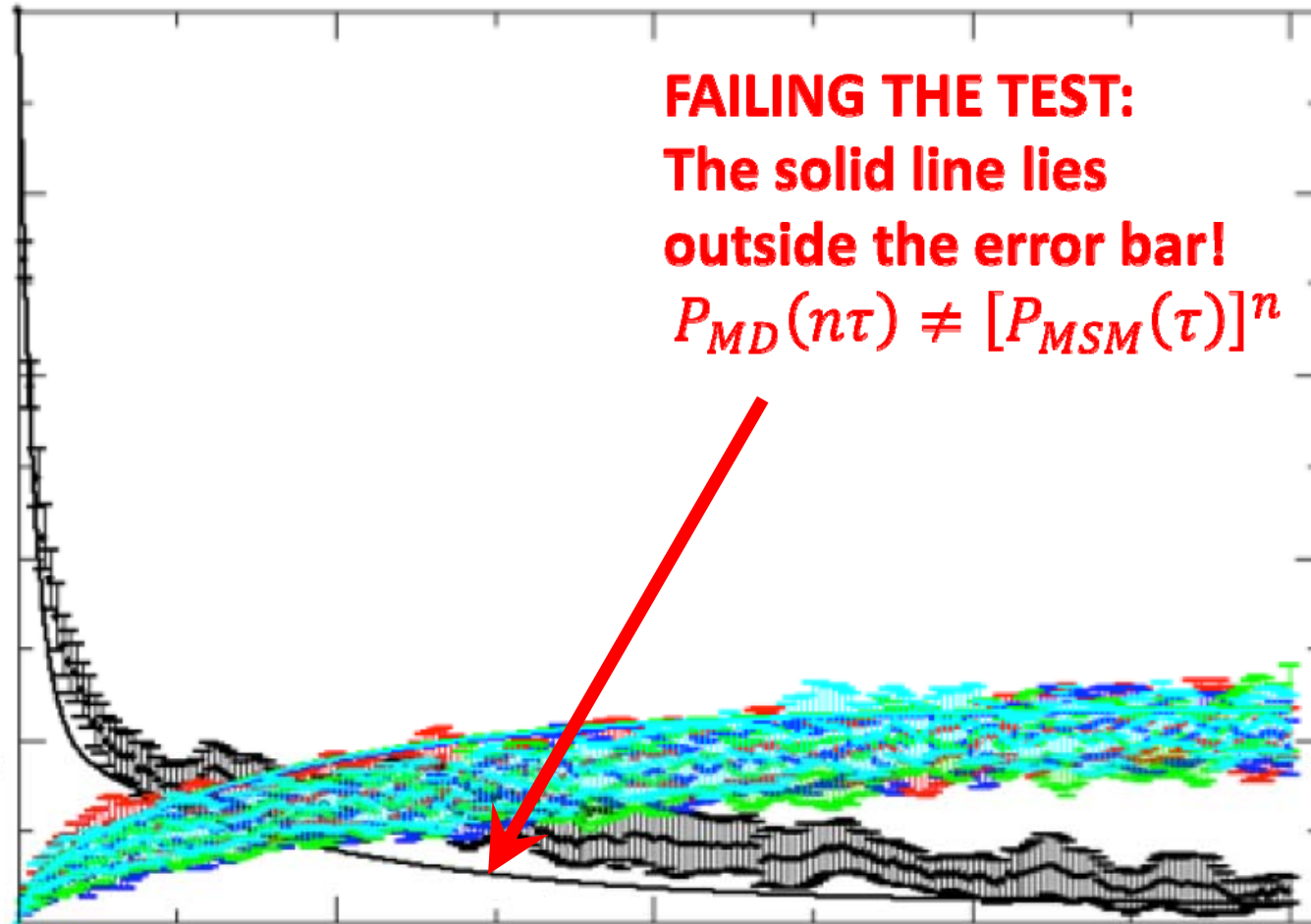


Figure courtesy of Luming

Example: a 5-state model

- State 1 • State 2 • State 3 • State 4 • State 5

$$IP_{MD}(n\tau) = I[P_{MSM}(\tau)]^n$$

Points with Error Bar:

$$[0 \ 0 \ 0 \ 0 \ 1] \begin{bmatrix} P_{11,MD} & P_{12,MD} & \dots & P_{15,MD} \\ P_{21,MD} & P_{22,MD} & \dots & P_{25,MD} \\ \vdots & \vdots & \ddots & \vdots \\ P_{51,MD} & P_{52,MD} & \dots & P_{55,MD} \end{bmatrix}$$

Points on solid line:

$$[0 \ 0 \ 0 \ 0 \ 1] \begin{bmatrix} P_{11,MSM} & P_{12,MSM} & \dots & P_{15,MSM} \\ P_{21,MSM} & P_{22,MSM} & \dots & P_{25,MSM} \\ \vdots & \vdots & \ddots & \vdots \\ P_{51,MSM} & P_{52,MSM} & \dots & P_{55,MSM} \end{bmatrix}$$

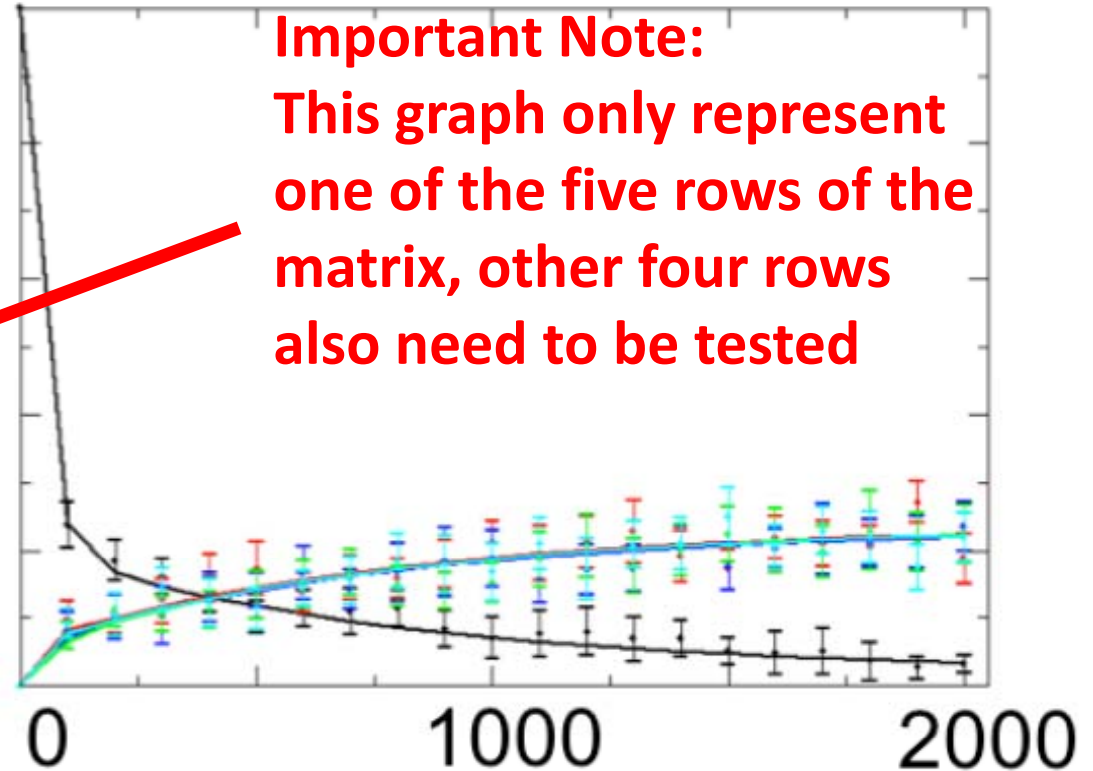


Figure courtesy of Luming

Varying the Lag Time

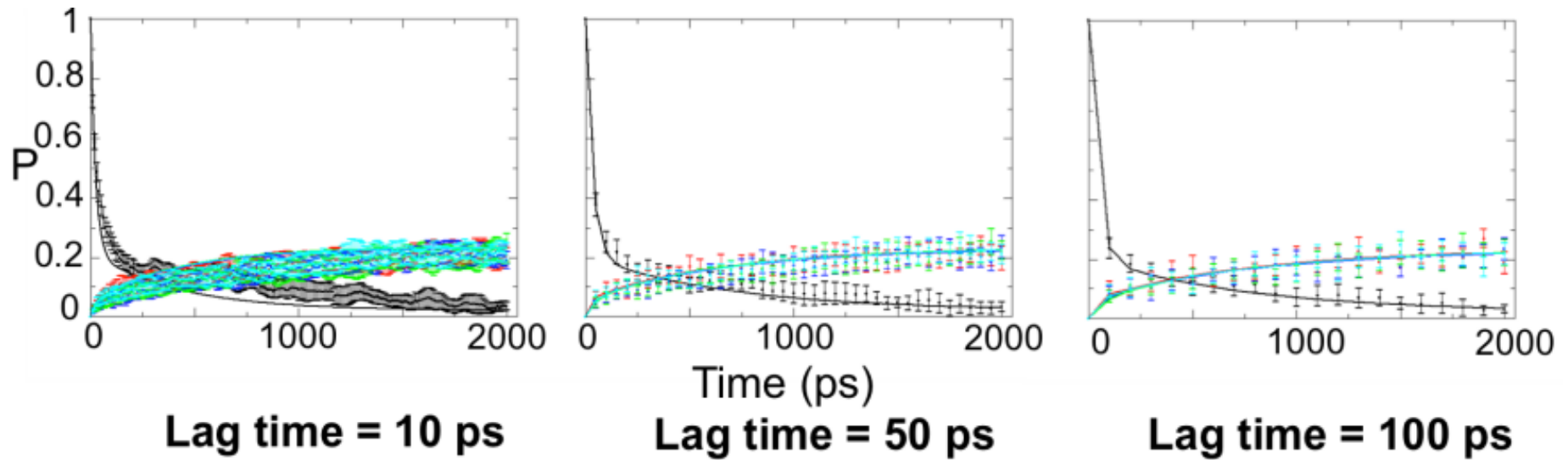


Figure courtesy of Luming

Frank Noé's Implementation

- A simplified version of Chapman-Kolmogorov test
- Considers only the equality of the diagonal term (i.e. self-transition probability) of:

$$P_{MD}(n\tau) = [P_{MSM}(\tau)]^n$$

Noé's Implementation Example: a 5-state model

- State 1 • State 2 • State 3 • State 4 • State 5

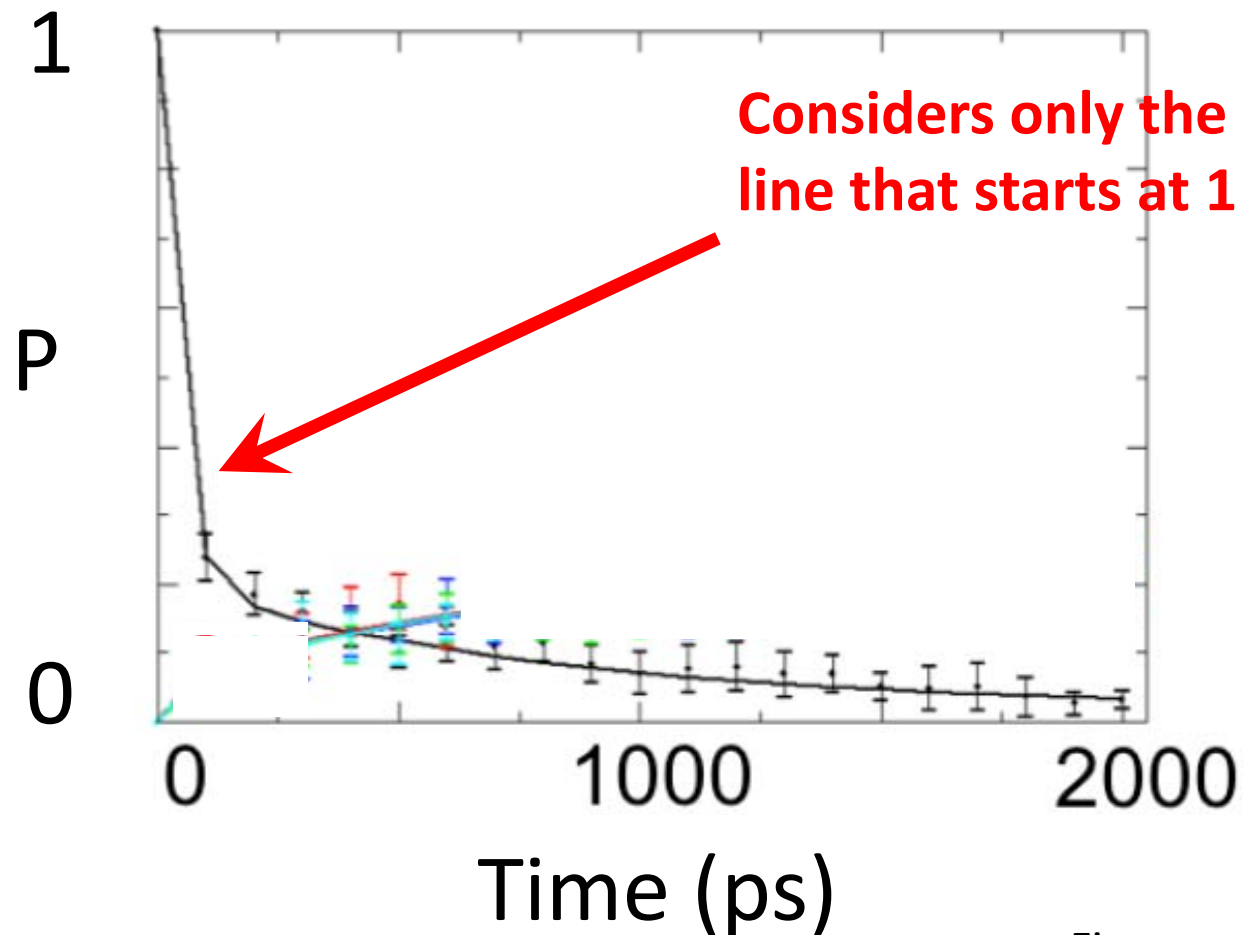
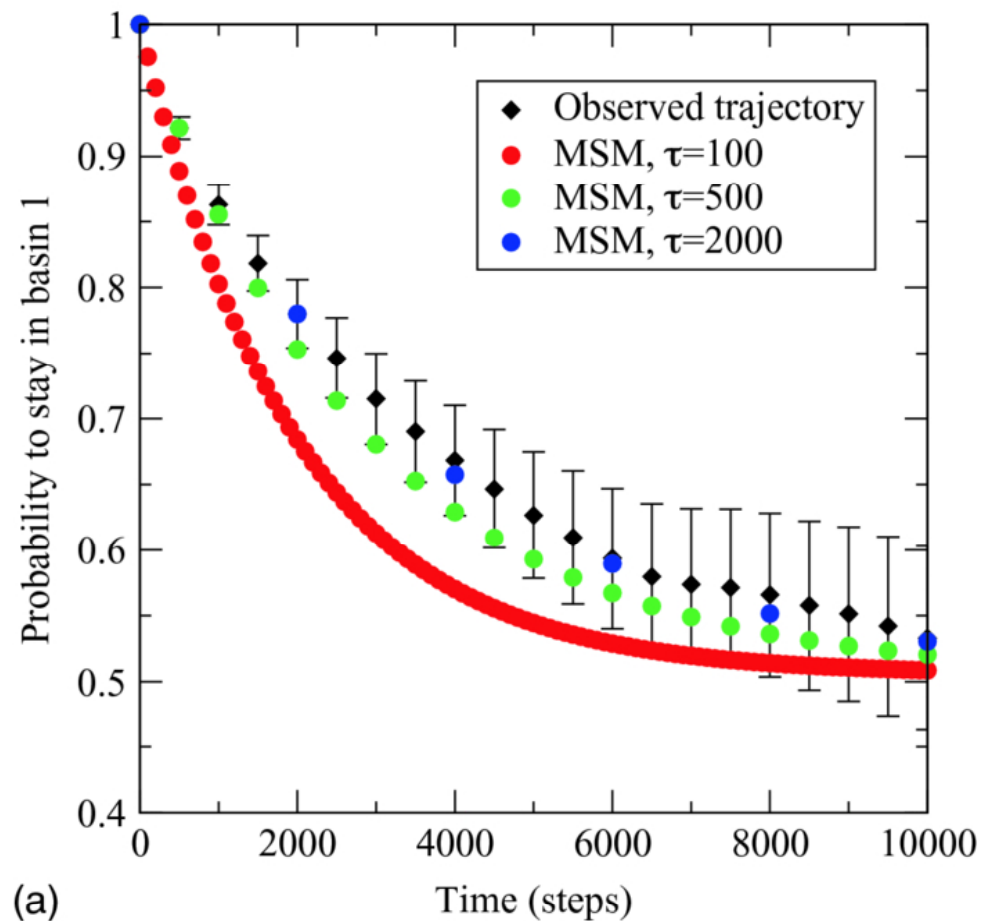


Figure courtesy of Luming

Noé's Implementation Example: a 2-state model

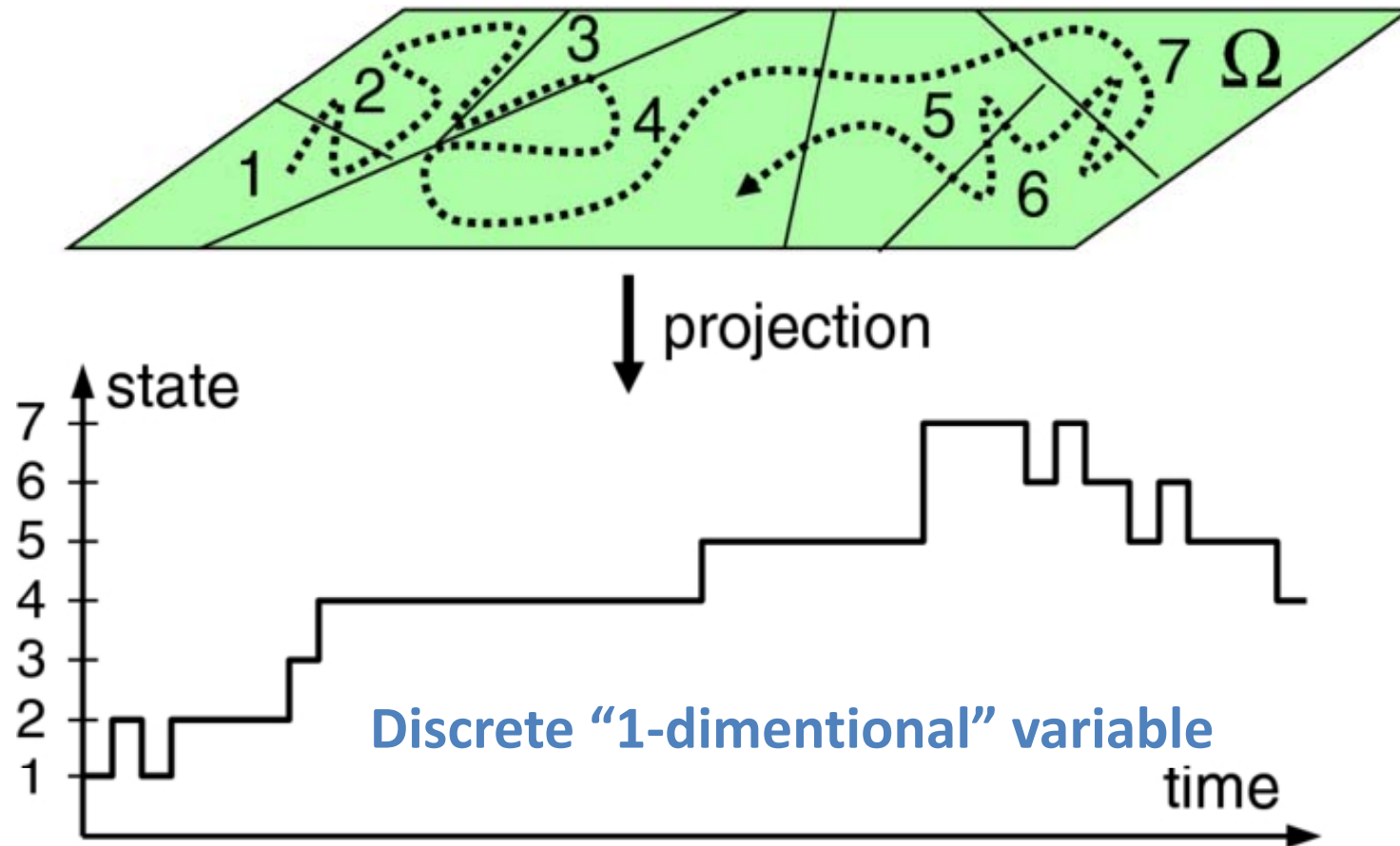


**Model built at $\tau = 100$
failed the test**

**Model at $\tau = 500$
marginally passed the test**

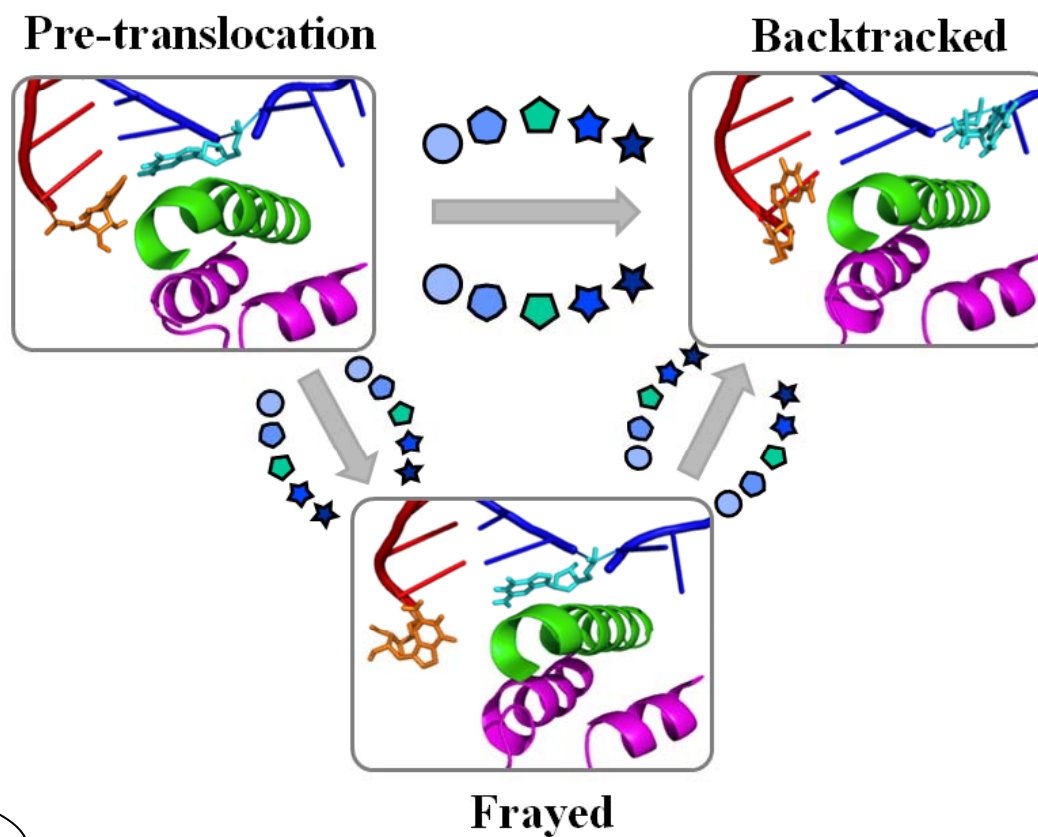
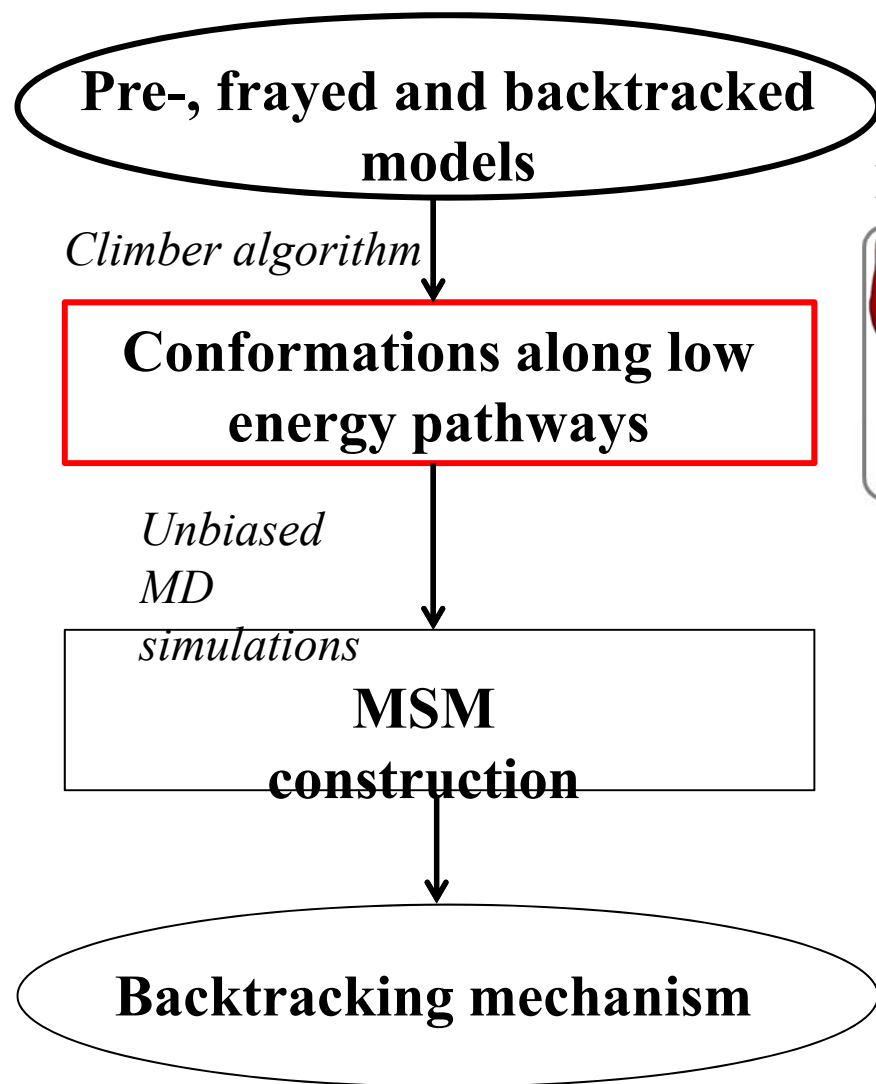
**Model at $\tau = 2000$
passed the test**

Continuous n-dimensional variable



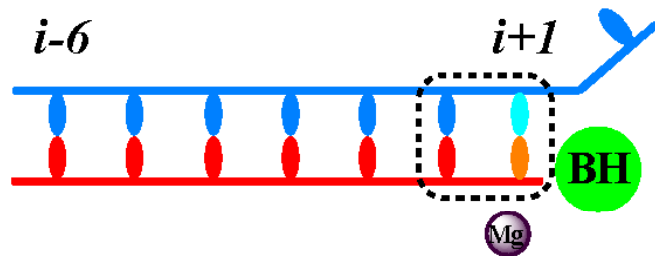
State: [1,2,1,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,7,7,7,6,7,6,5,6,5,5,5,4, ...]

Initial low-energy backtracking pathways

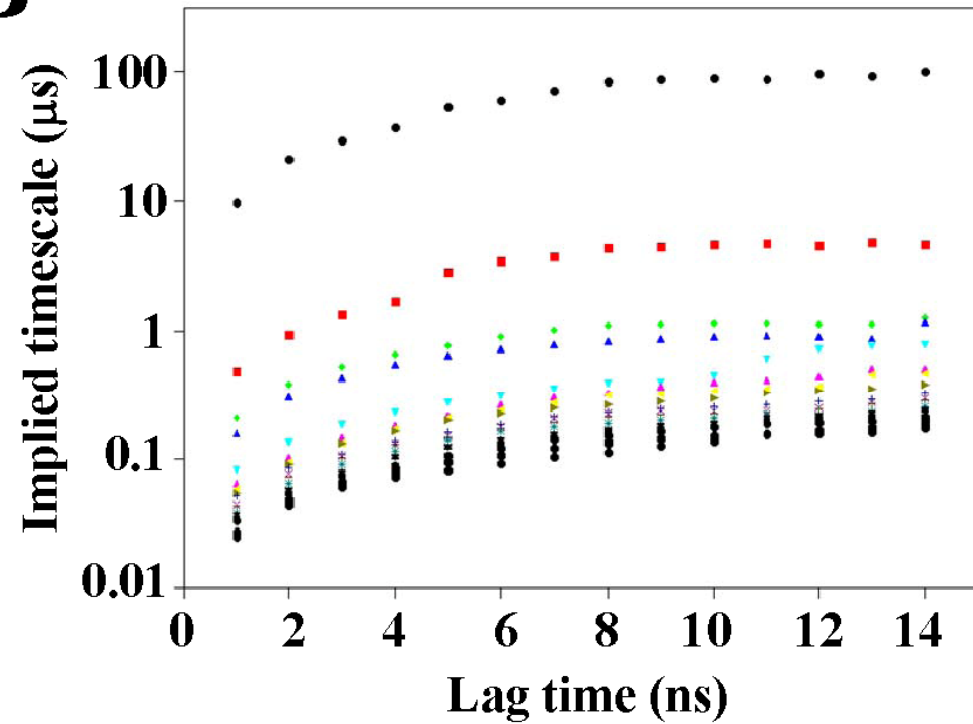


Validating the MSM

A



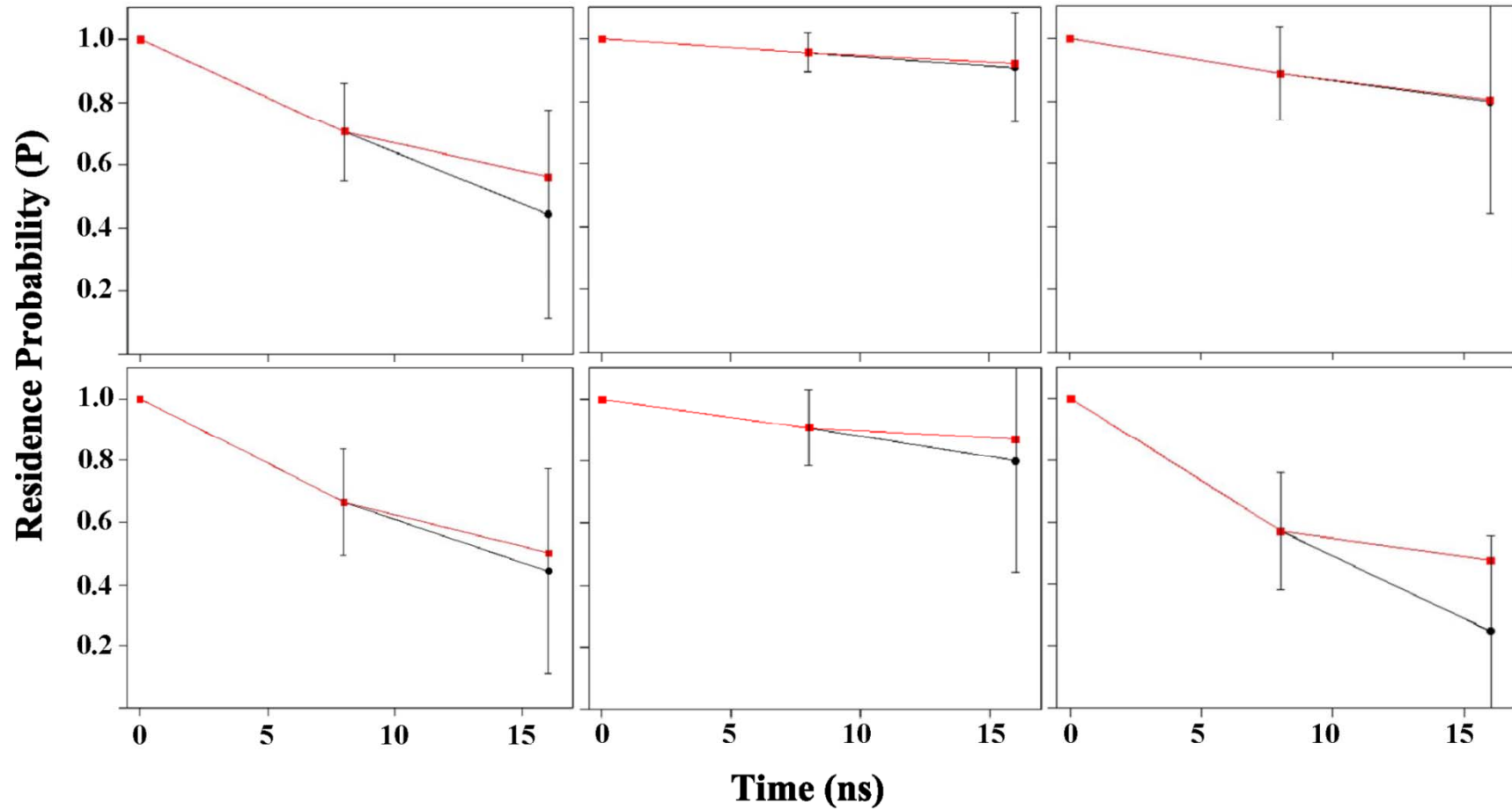
B



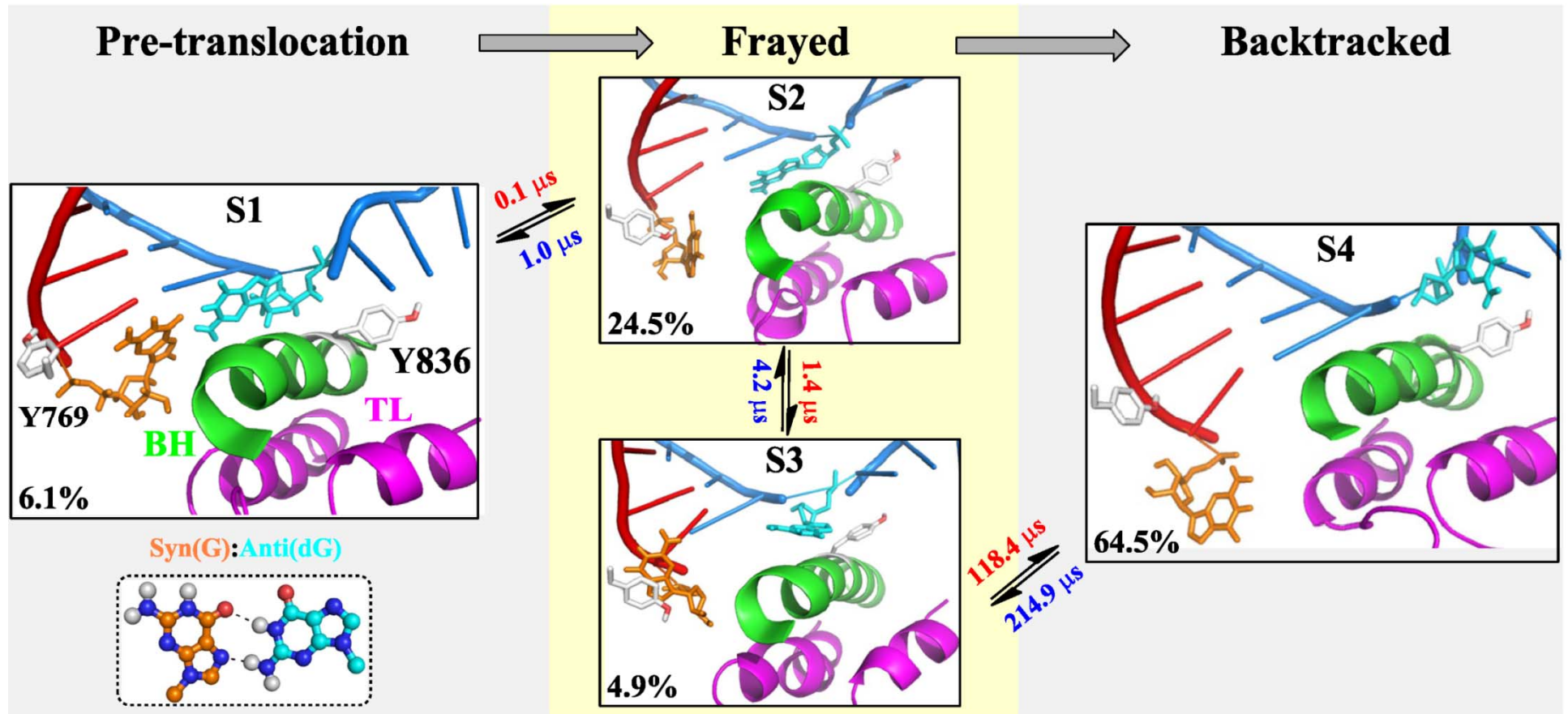
Chapman-Kolmogorov test

MSM

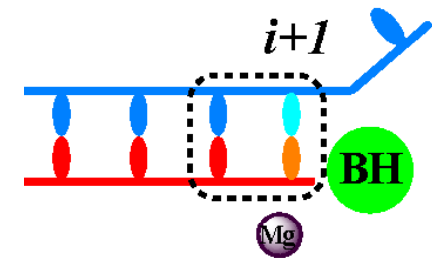
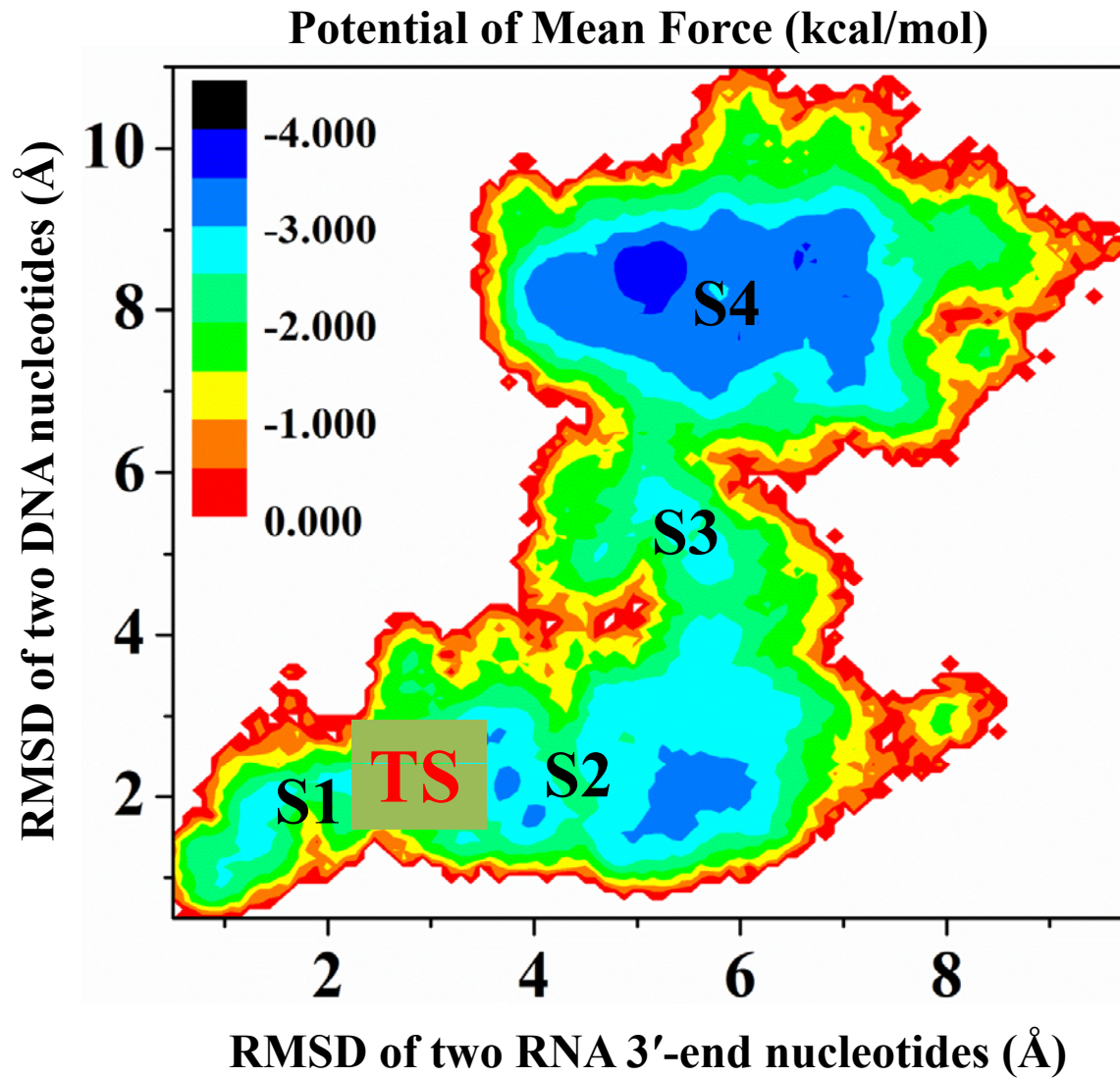
MD



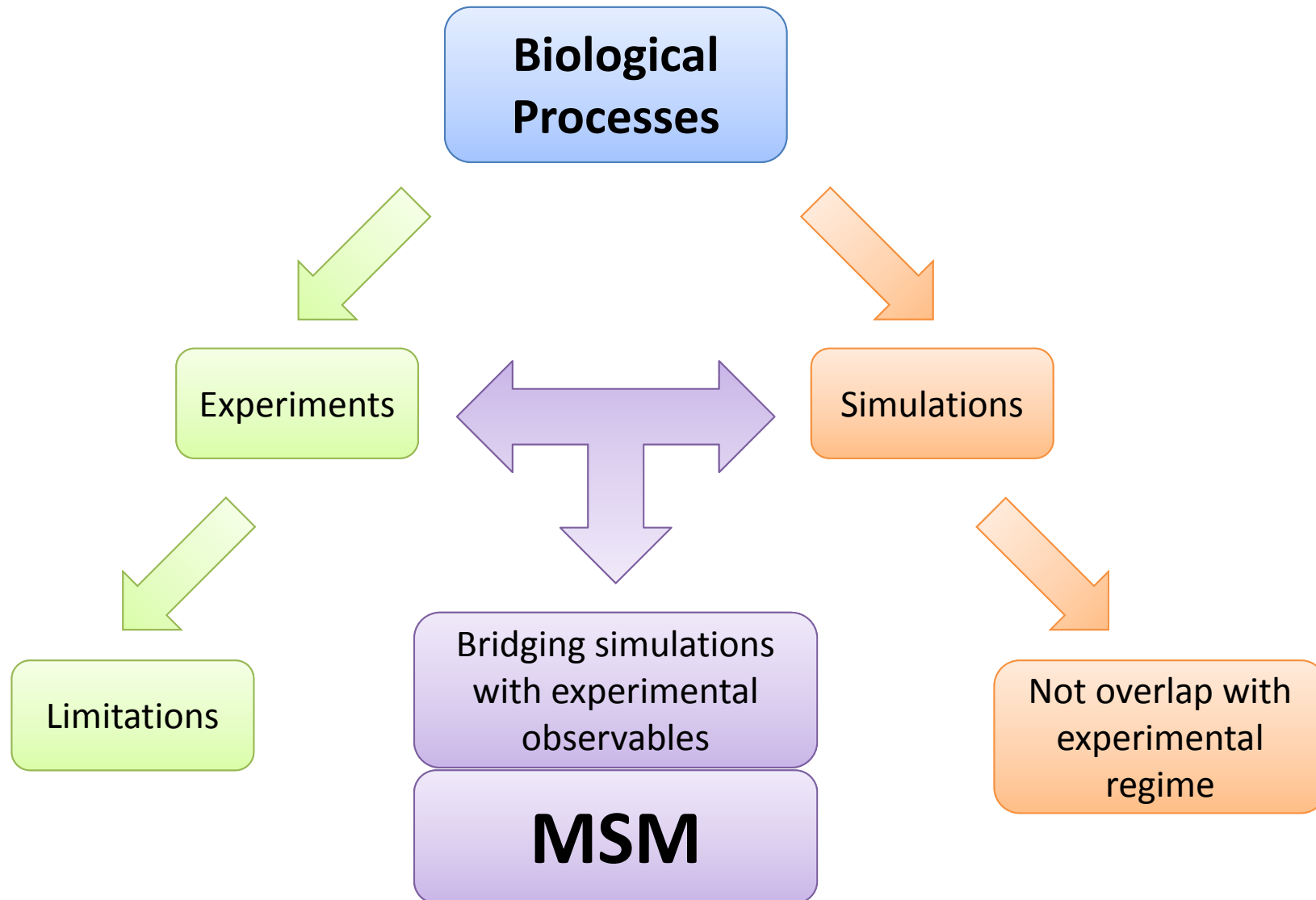
Four metastable states identified by MSM



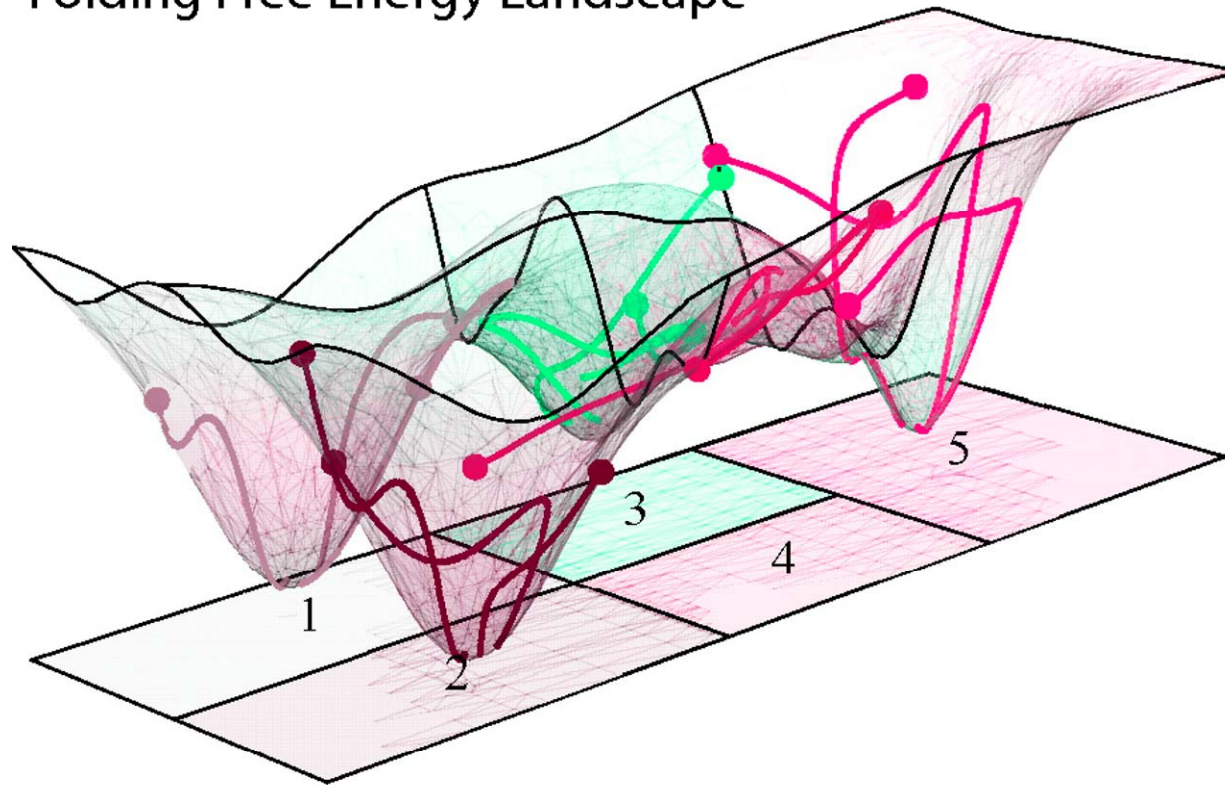
Backtracking preference



Reasons for Using MSM



Folding Free Energy Landscape



Metastable States
from Markov State Models