MSMBuilder Requirements

1. Linux or OSX
2. Enthough Python Distribution or a Ubuntu VM
3. MSMBuilder (https://github.com/SimTk/msmbuilder)
4. CPU with SSE3 support.
5. GCC 4.2 or later (with OpenMP support)
6. pymol (optional for visualization)

Markov State Models for Simulation Analysis

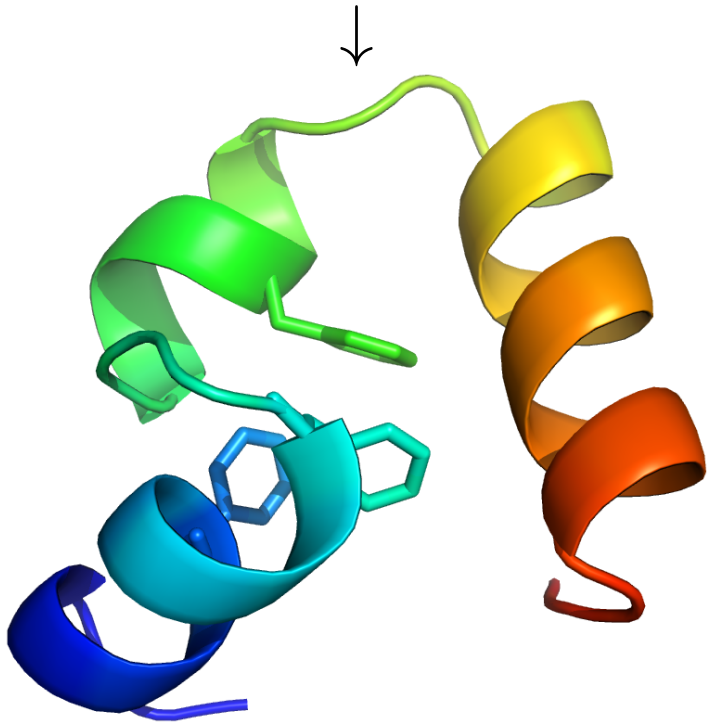
Kyle Beauchamp, Robert McGibbon

March 25, 2013

Conformational States of Biological Molecules

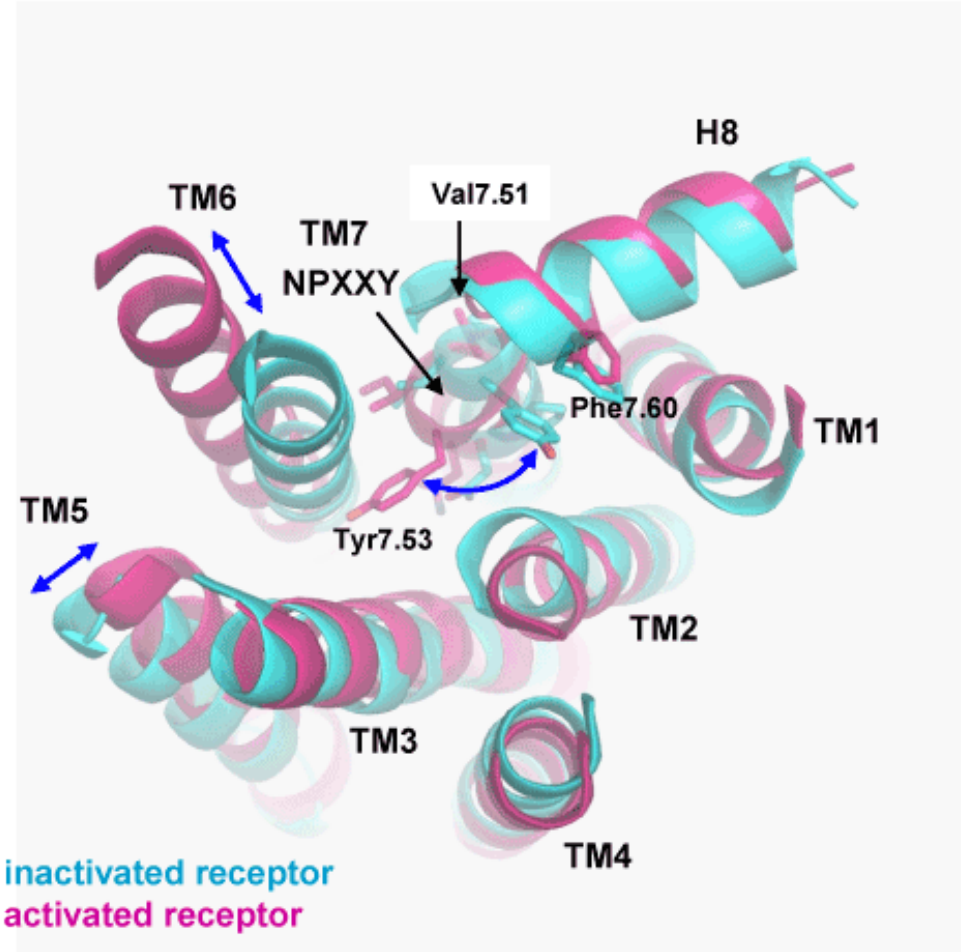
Protein Folding

MLSDEDFKAVFGMTRSAFANLPLWKQQNLKKEKGLF



Conformational States of Biological Molecules

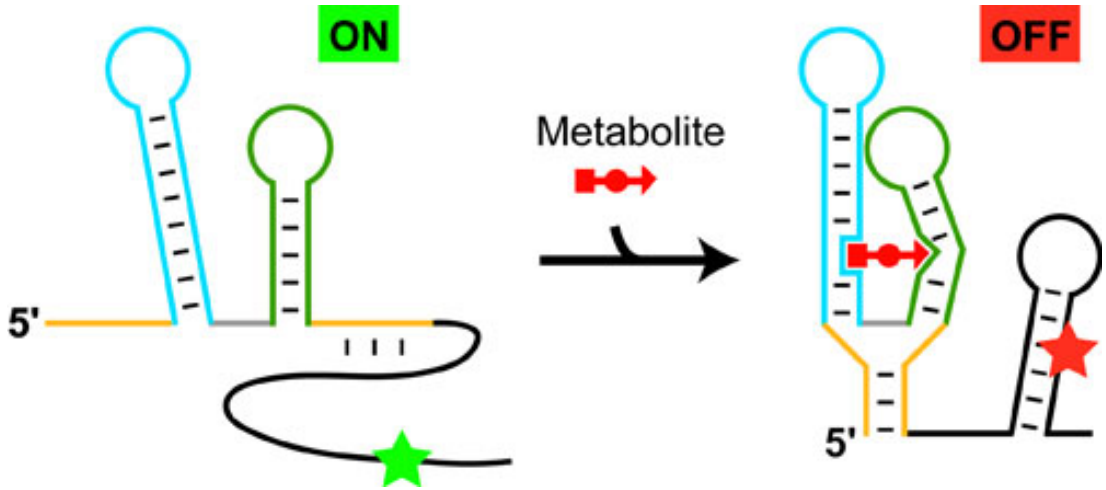
GPCR Dynamics



Jean-Francois Deleuze, 2010

Conformational States of Biological Molecules

Riboswitches



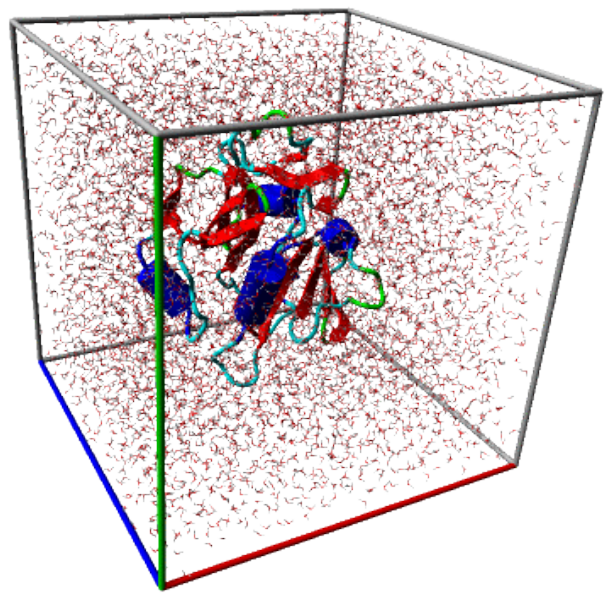
Breaker et al

Key Points:

- Importance of conformation
- Multiple States
- Dynamics

Molecular Dynamics

Molecular dynamics simulations capture equilibrium and kinetic properties of biomolecules.



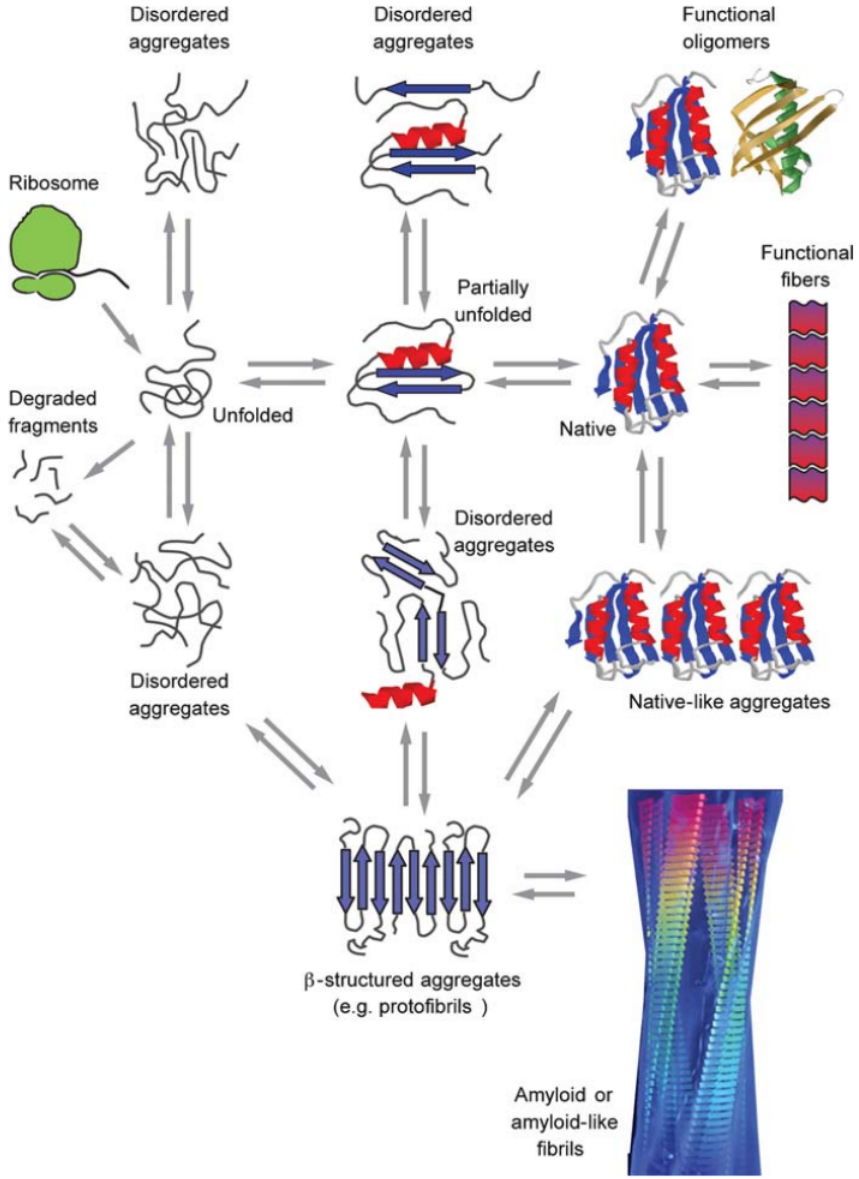
How should we analyze simulation?

1. Direct, quantitative connection to experimental observables
2. Intuitive explanation (coarse-graining)
3. Statistically optimal use of limited data
4. Computationally tractable and easy-to-use
5. Compatible with both many and few-state behavior

Markov State Models achieve these goals!

States and Rates

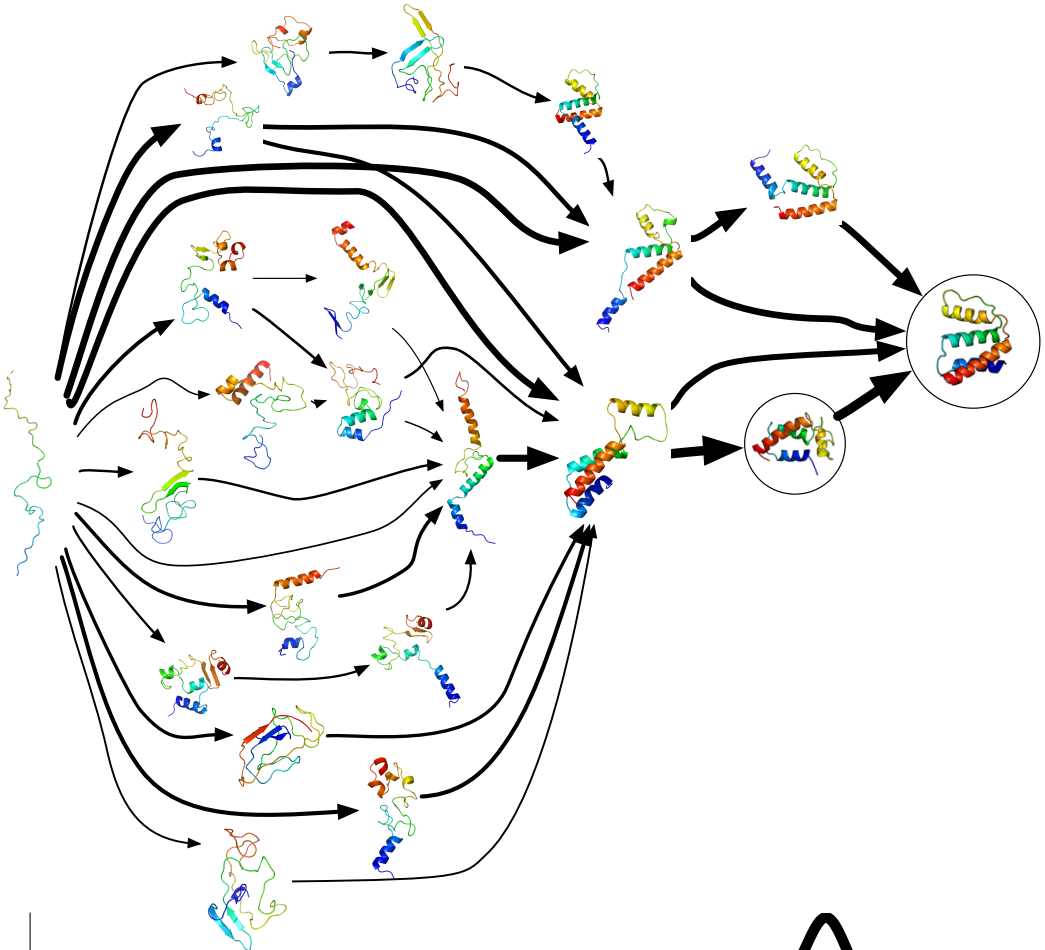
Experimentalists view biomolecules through the lens of "states and rates"



Dobson, 2006.

States and Rates

Markov State Models provide a "states and rates" view on conformational dynamics



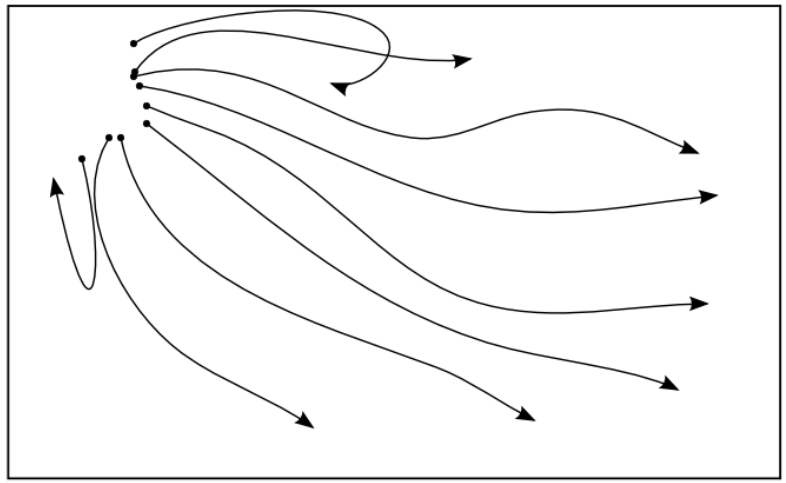
Voelz et al.

Markov State Models in a Nutshell

1. Define states by clustering.
2. Estimate rates between states.

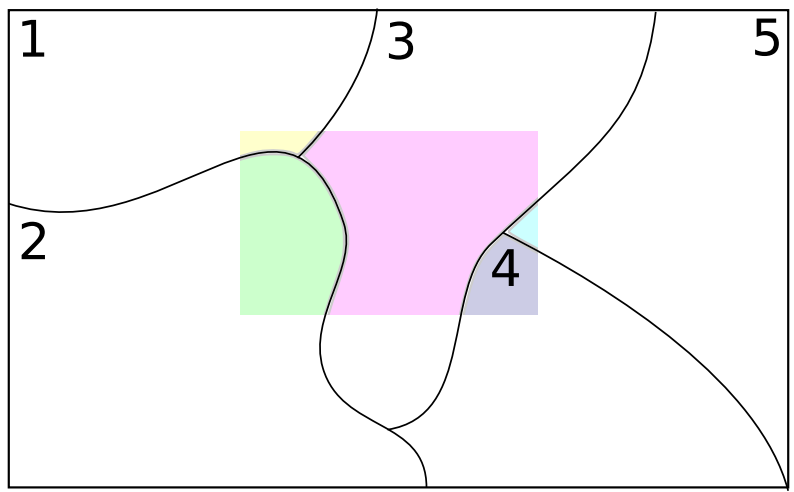
Markov State Models

Suppose we have an ensemble of molecular dynamics trajectories:



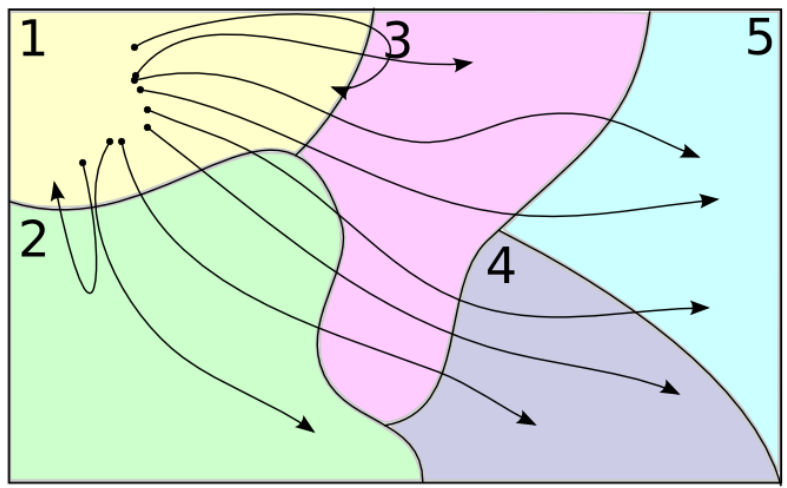
Markov State Models

Cluster the conformation space into disjoint states: $\{1, 2, 3, 4, 5\}$



Markov State Models

Estimate the transition probabilities by counting jumps between states:



Estimating Transition Probabilities

Suppose we slice our trajectories every Δt picoseconds (lagtime) and count the observed transitions:

$$C_{ij} = C_{i \to j}$$

Estimating Transition Probabilities

To get the transition probabilities, we simply "normalize" the counts:

$$T_{ij} = T_{i \to j} = \frac{C_{ij}}{\sum_k C_{ik}}$$

Dynamics in an MSM

Suppose that at time zero a protein sits in state i.

After lagtime Δt, jump to another state with probabilities T_{ij}.

Dynamics in an MSM

Suppose have an ensemble of proteins occupying different states–we describe this by a population vector $\mathbf{x}(\mathbf{0})$.

$$\mathbf{x}(\mathbf{t}) = \mathbf{T}\mathbf{x}(\mathbf{0})$$

Eigenvalues and Eigenvectors

Equilibrium

$$Tv = \lambda v$$

Setting $\lambda = 1$ gives us the equilibrium populations:

$$T\pi = 1\pi = \pi$$

At long times, the system approaches equilibrium populations:

$$x(t) \to \pi$$

Eigenvalues and Eigenvectors

Dynamics

$$Tv = \lambda_i v$$

For the remaining eigenvalues, $\lambda_i < 1$.
These eigenvalues correspond to characteristic timescales at which different populations approach equilibrium.

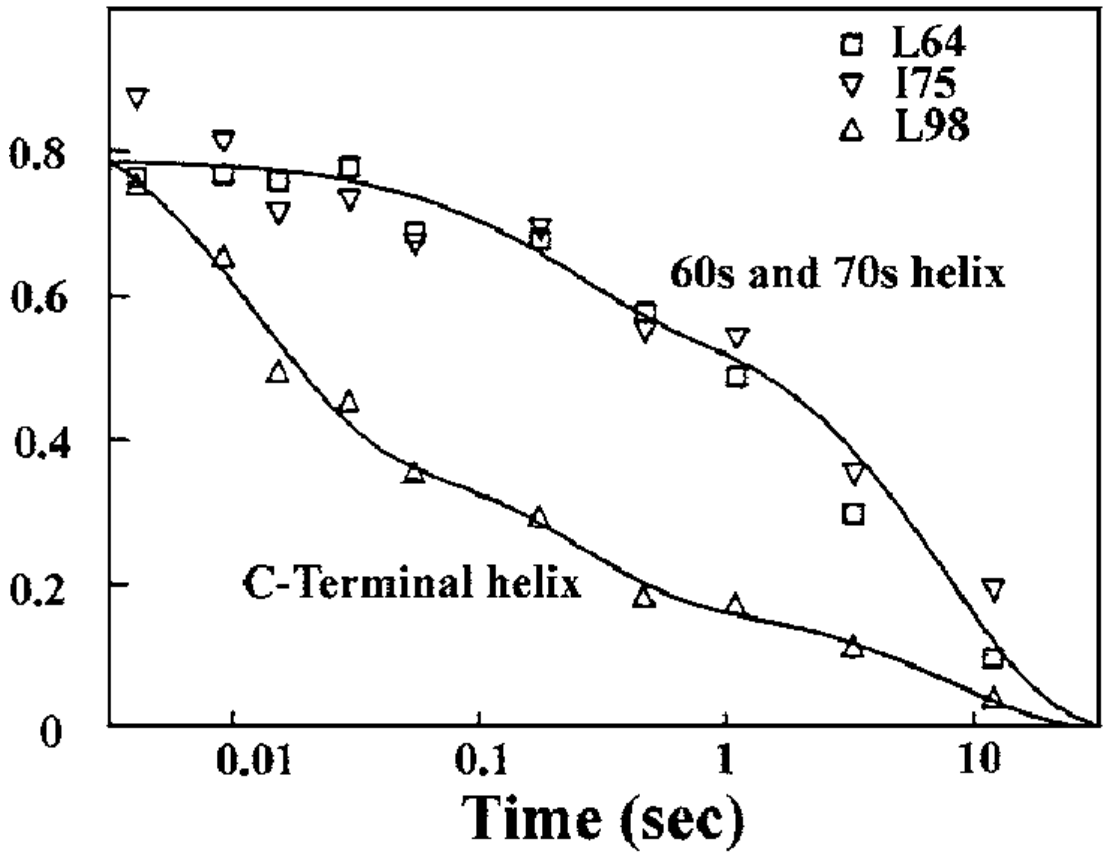
$$\tau_i = -\frac{\Delta t}{\log \lambda_i}$$

Eigenvalues and Eigenvectors

Projection

Suppose we have an experiment that monitors a single variable $y(t)$.

$$y(t) = \sum_i c_i \exp(-\frac{t}{t_i}) < \mathbf{v_i}, \mathbf{x}(0) >$$



Englander

What can you do with a Markov State Model?

Ligand Binding

Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations

Ignasi Buch *, Toni Giorgino *, and Gianni De Fabritiis *

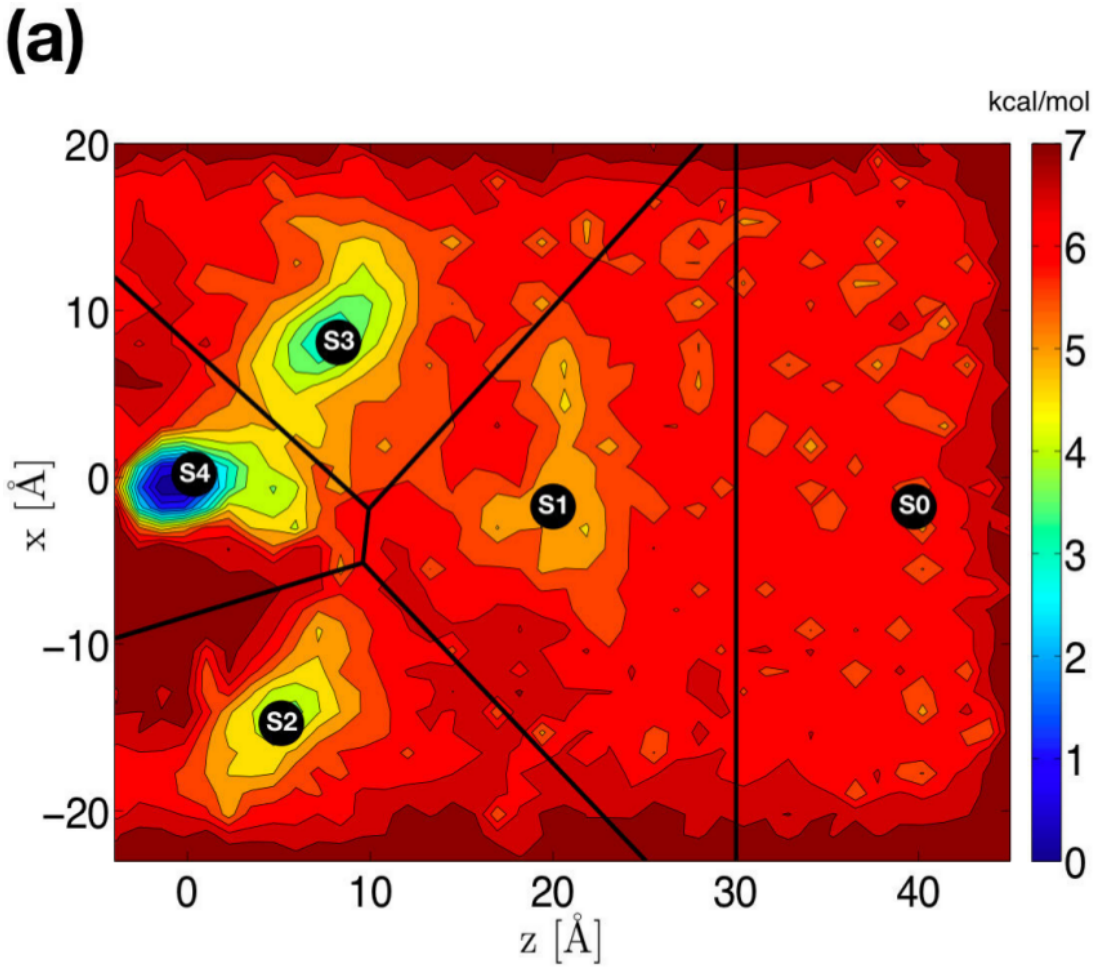


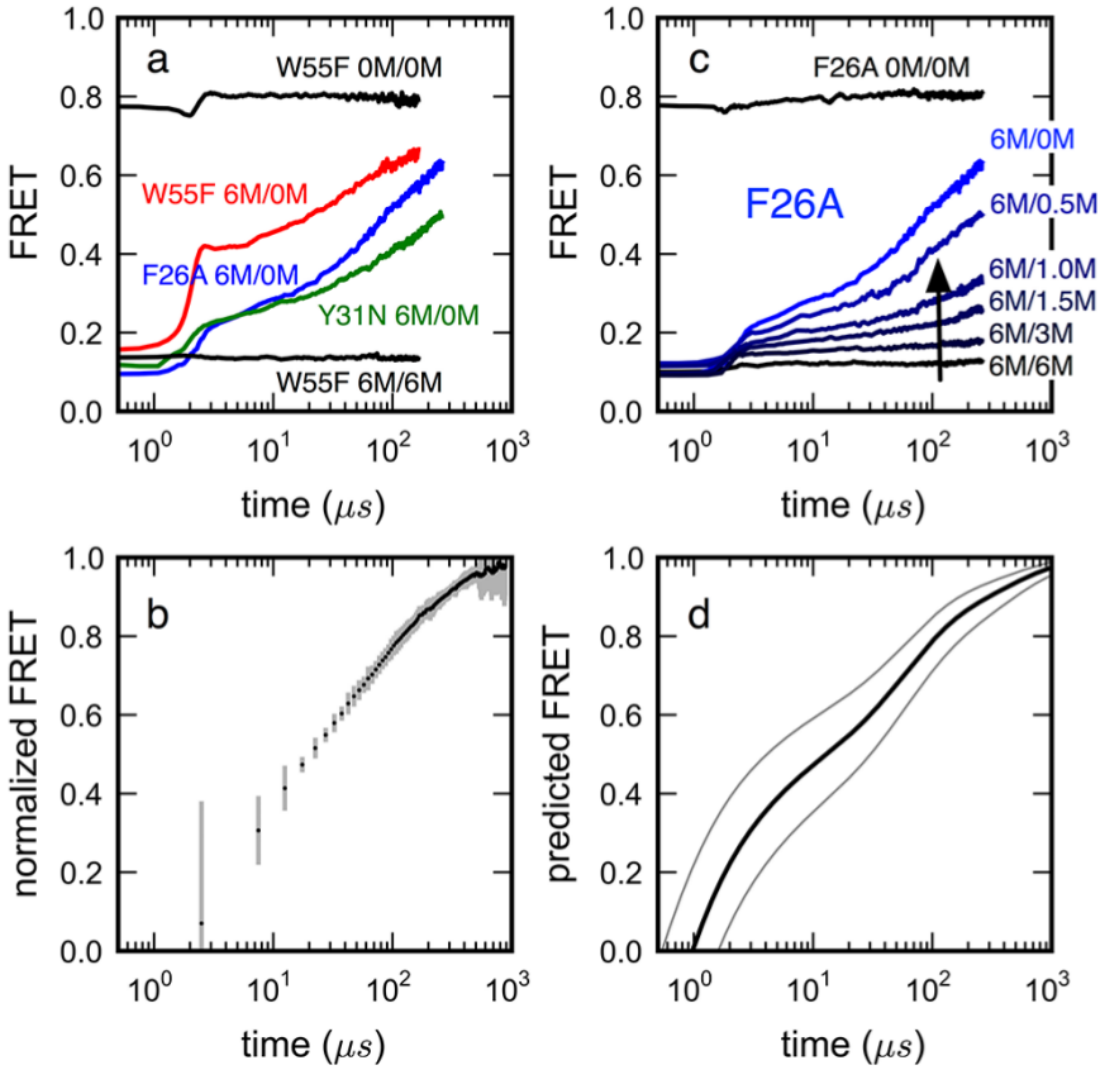
Figure 3. Identification of metastable states. (a) Potential of mean force

What can you do with a Markov State Model?

Predict Experiments

Slow Unfolded-State Structuring in Acyl-CoA Binding Protein Folding Revealed by Simulation and Experiment

Vincent A. Voelz,[⊥,†] Marcus Jäger,[¶,‡] Shuhuai Yao,[□] Yujie Chen,[○,#] Li Zhu,[○,△] Steven A. Waldauer,[○,§] Gregory R. Bowman,[⊥,∥] Mark Friedrichs,[▽] Olgica Bakajin,[▲] Lisa J. Lapidus,[○] Shimon Weiss,*[,▼] and Vijay S. Pande*[,■]

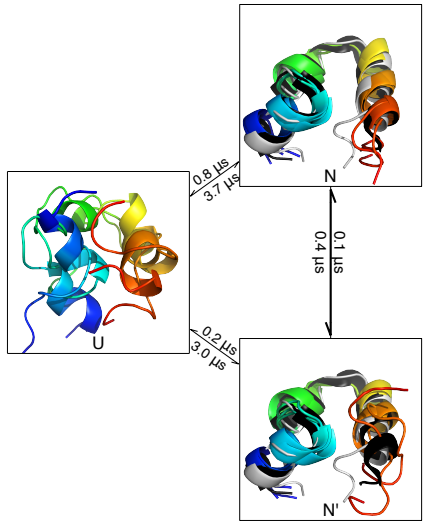


What can you do with a Markov State Model?

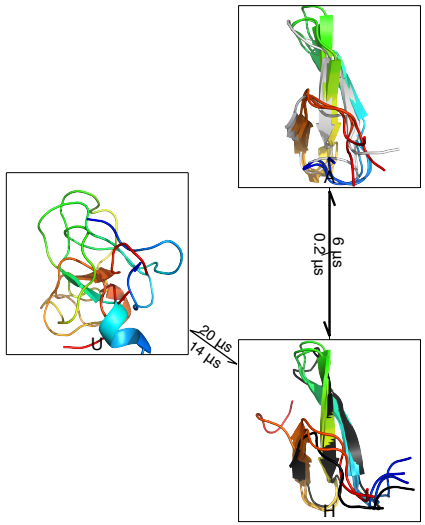
Construct Simple Models

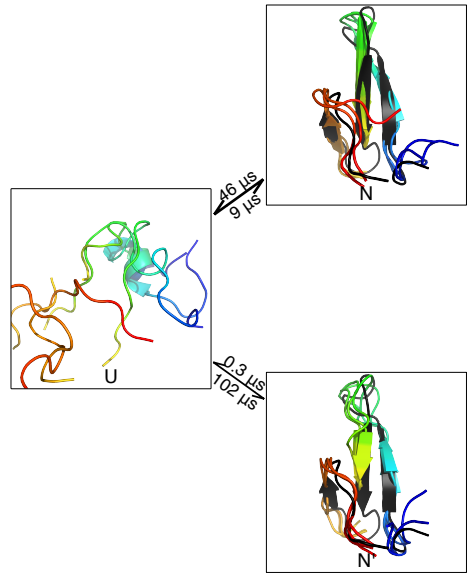
Simple few-state models reveal hidden complexity in protein folding

Kyle A. Beauchamp[a], Robert McGibbon[b], Yu-Shan Lin[b], and Vijay S. Pande[b,1]



(a)



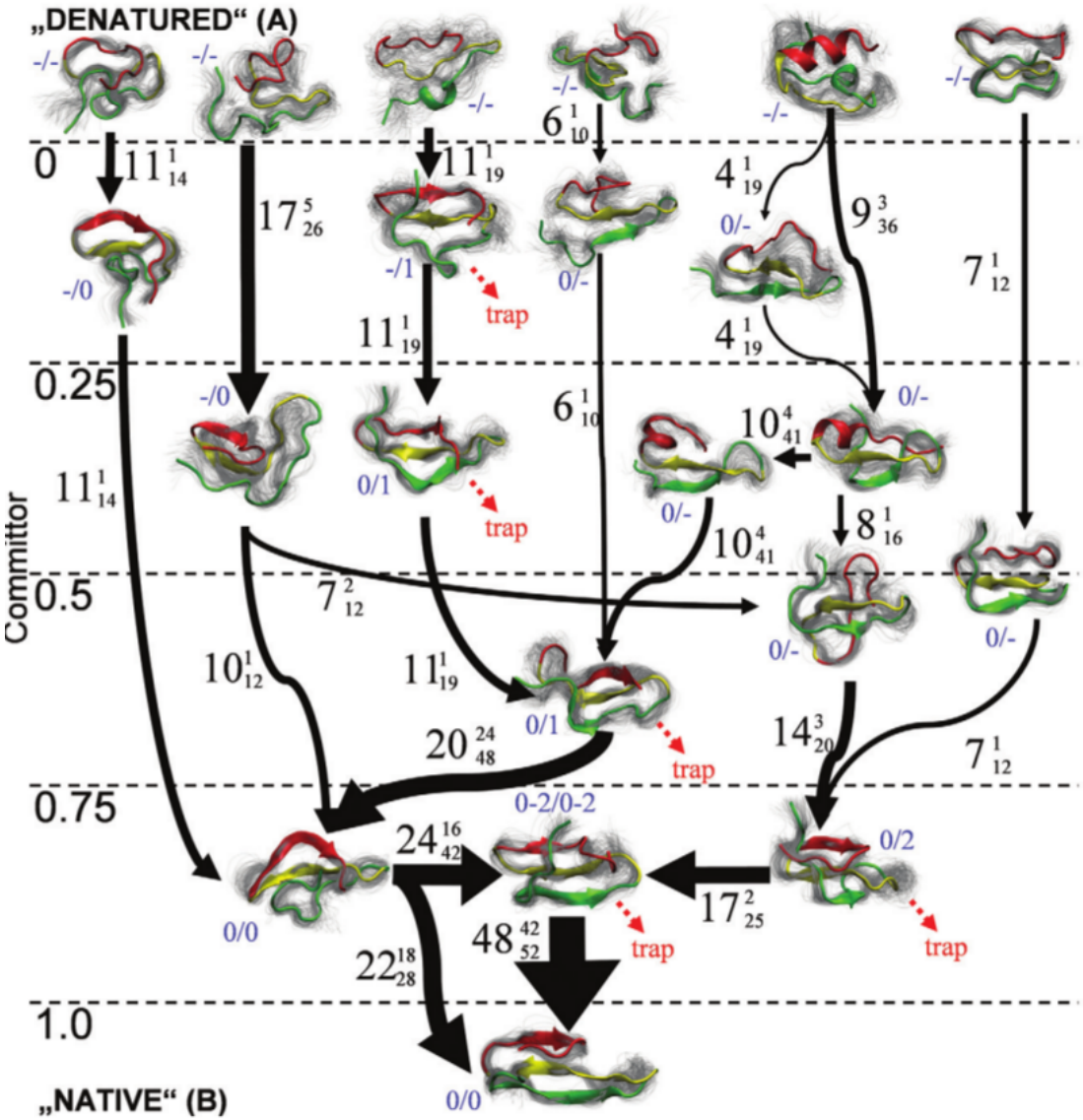


What can you do with a Markov State Model?

Extract Pathways

Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations

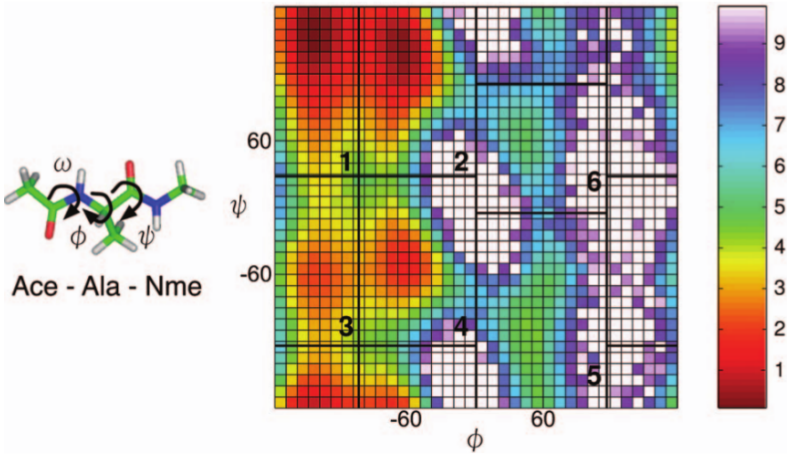
Frank Noé[a,1], Christof Schütte[a], Eric Vanden-Eijnden[b], Lothar Reich[c], and Thomas R. Weikl[c]



Introduction to MSMBuilder

Typical MSMBuilder Workflow

1. Data preparation
2. Build microstate model (clustering)
3. Build macrostate model using lumping
4. Investigate macrostates



Data Preparation

Convert XTC trajectories into MSMBuilder (HDF: .lh5) files:

```
cd ~/msmbuilder/Tutorial
ConvertDataToHDF.py  -s native.pdb -i XTC
```

MSMBuilder can read XTC, PDB, DCD, and other formats.

Cluster your data

To define microstates, clusters your data using the RMSD metric with hybrid k-centers k-medoids clustering.

```
Cluster.py rmsd hybrid -d 0.045 -l 50
```

1. Use the "rmsd" distance metric
2. Use the hybrid k-centers k-medoids clustering algorithm
3. Stop clustering when cluster radii are less than 0.045 nm
4. Refine clusters with 50 iterations of hybrid k-medoids

How to get help

Access help at command line with "-h" option.
Some help menus are context dependent:

- Cluster.py -h
- Cluster.py rmsd -h
- Cluster.py rmsd hybrid -h

Clustering Output Files

By default, Cluster.py will produce three files:

Data/Assignments.h5 is the set of state assignments.

Data/Assignments.h5.distances is the set of distances from each frame to its assigned cluster.

Data/Gens.lh5 is the set of cluster centers.

Choosing a lagtime

What is a lagtime?

MSM calculations require the user to pick a fixed lagtime.

Lagtime = the time window used when counting transitions.

The lagtime can any integer multiple of the trajectory output frequency.

Implied Timescales

The eigenvalues of the transition matrix provide the "implied timescales" of the model:

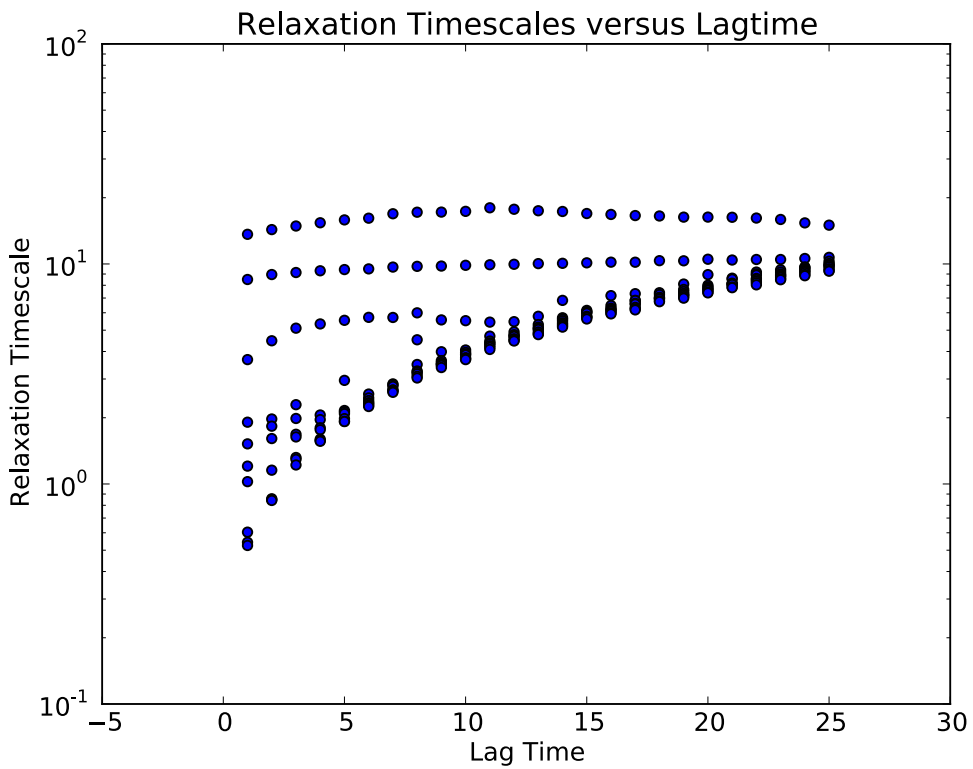
Implied timescales serve three roles:

1. Choose the number of macrostates via the "spectral gap"
2. Choose the macrostate lagtime via "leveling-off"
3. Experimental observables decay via a sum of exponentials with these timescales.

Calculating Implied Timescales

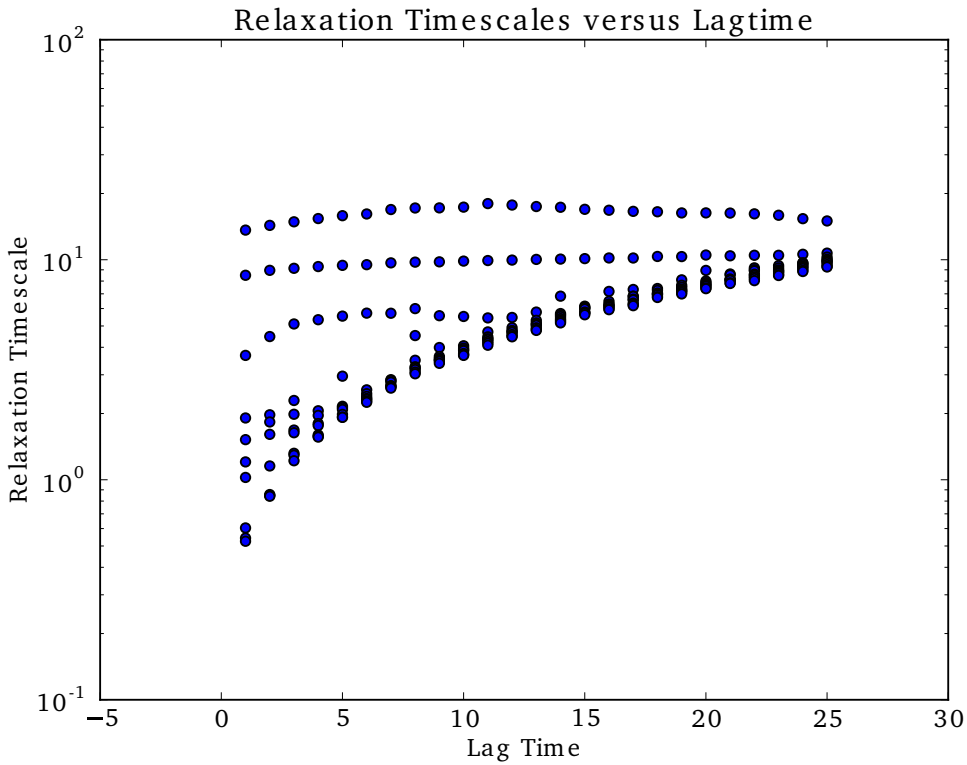
```
CalculateImpliedTimescales.py -l 1,25 -i 1 -o \
Data/ImpliedTimescales.dat

PlotImpliedTimescales.py -d 1. -i Data/ImpliedTimescales.dat
```



Determine the number of macrostates

The top 3 timescales are separated by a spectral gap.



Three slow timescales suggests building a four macrostate model.

Build a Macrostate Model

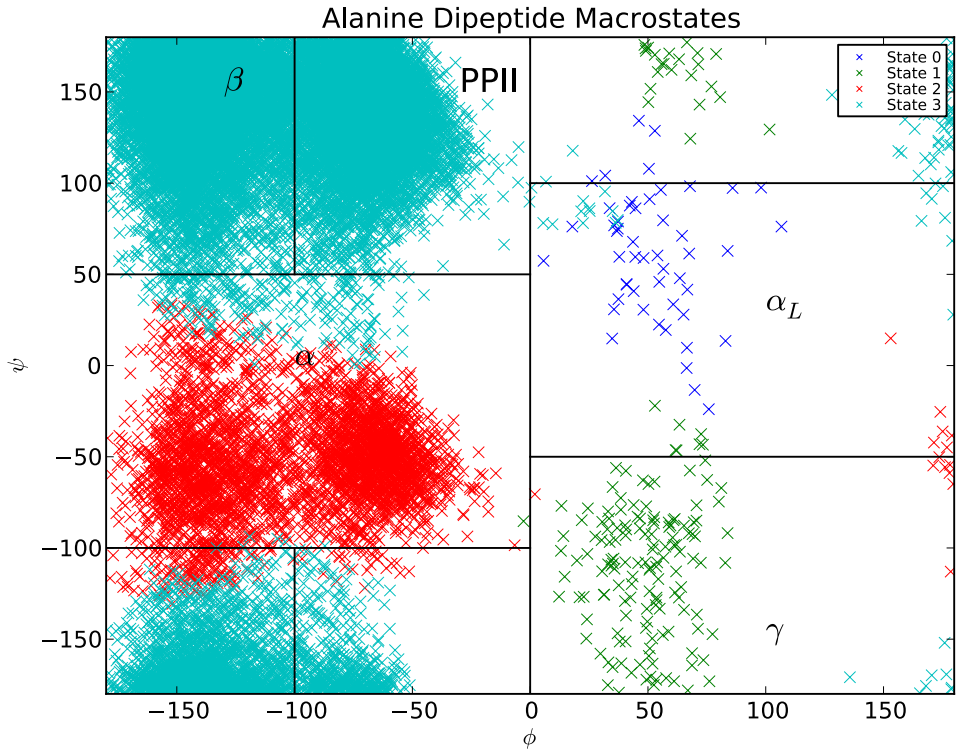
1. Build a transition matrix with lagtime of 1 ps
2. Use the transition matrix as input to PCCA+ algorithm

```
BuildMSM.py -l 1 -o L1

PCCA.py -n 4 -a L1/Assignments.Fixed.h5 -t L1/tProb.mtx \
-o Macro4/ -A PCCA+
```

Visualize Macrostates

```
python PlotDihedrals.py Macro4/MacroAssignments.h5
```



Note the agreement with a manual state decomposition from Tobin Sosnick.

Errors in Markov State Models

MSM modeling requires that the data be Markovian, or memoryless.

We can use implied timescales to check that data is truly Markovian:

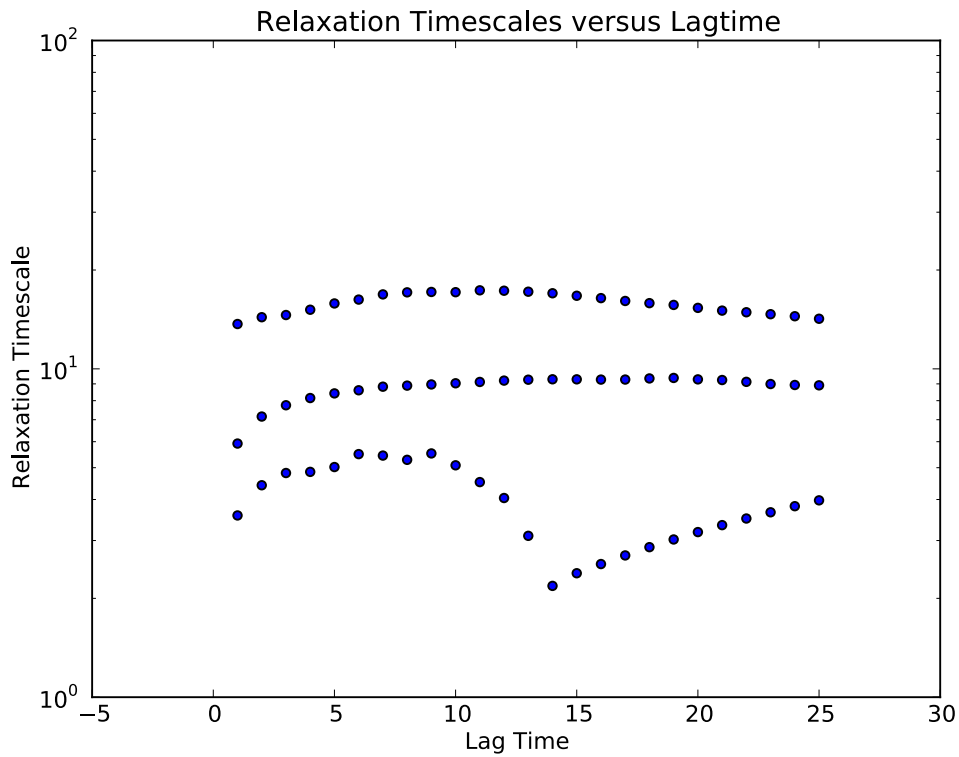
$$\tau = -\frac{\Delta t}{\log(\lambda)}$$

This *implied timescale* should be independent of the lagtime (Δt) used to slice your trajectories!

Validate Macrostate MSM

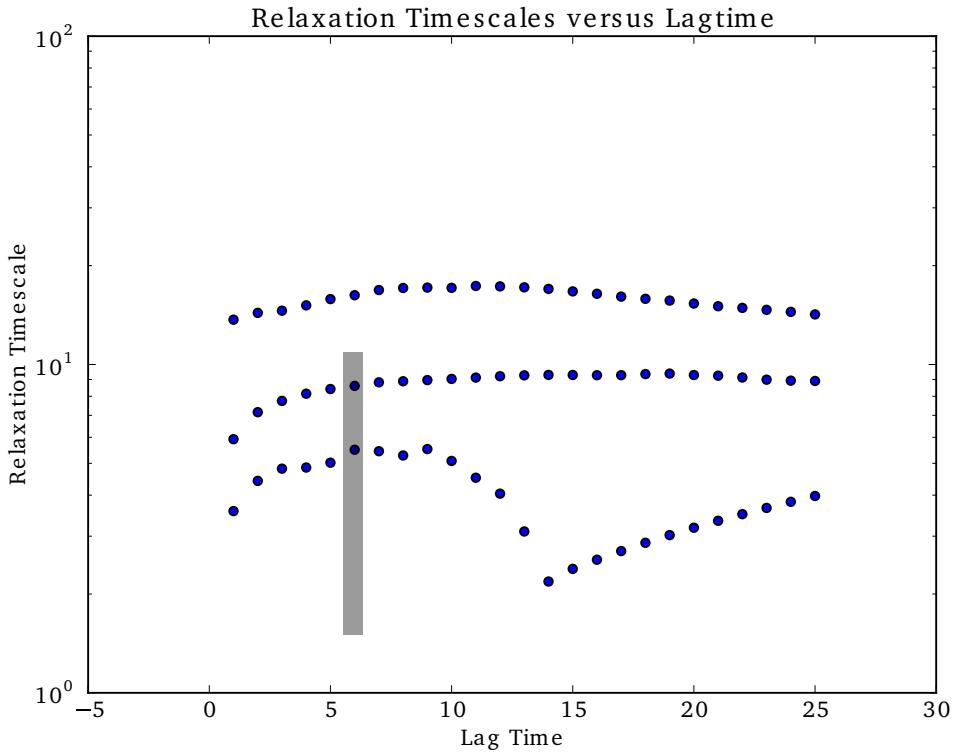
```
  CalculateImpliedTimescales.py -l 1,25 -i 1 \
-o Macro4/ImpliedTimescales.dat \
-a Macro4/MacroAssignments.h5 -e 3

PlotImpliedTimescales.py -i Macro4/ImpliedTimescales.dat -d 1
```



Build a "converged" model

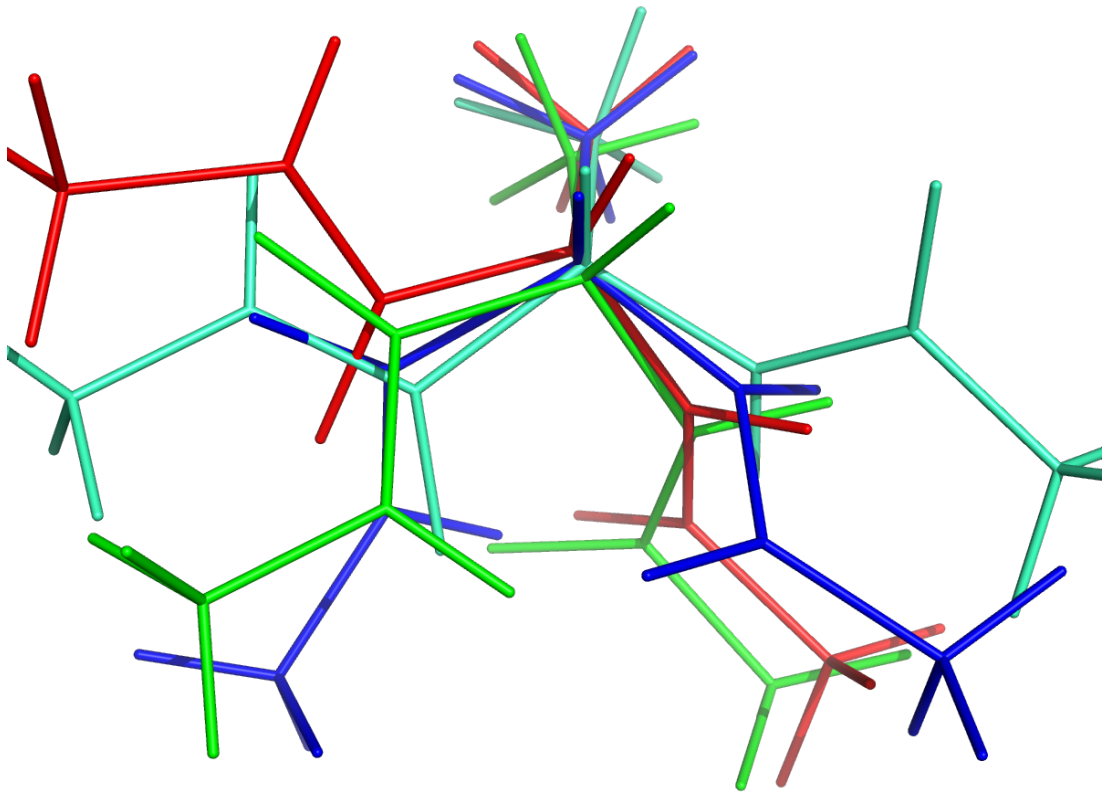
Note the convergence at 6 ps.




```
BuildMSM.py -l 6 -a Macro4/MacroAssignments.h5 -o Macro4/
```

Save PDBs

```
SaveStructures.py -s -1 -a Macro4/MacroAssignments.h5 \
-c 1 -S sep
pymol PDBs/State0-0.pdb PDBs/State1-0.pdb PDBs/State2-0.pdb \
PDBs/State3-0.pdb
```



Visit our GitHub page to report any issues!

https://github.com/SimTk/msmbuilder/