

Markov-state modeling of biomolecular systems, II



Toni Giorgino

National Research Council of Italy

toni.giorgino@cnr.it

www.giorginolab.it



@giorginolab

Thesis projects available

[github.com/giorginolab/
Markov-Tutorial-UniPd-2022](https://github.com/giorginolab/Markov-Tutorial-UniPd-2022)

University of Padova c/o Prof. Fuxreiter
Mar 23, 2022

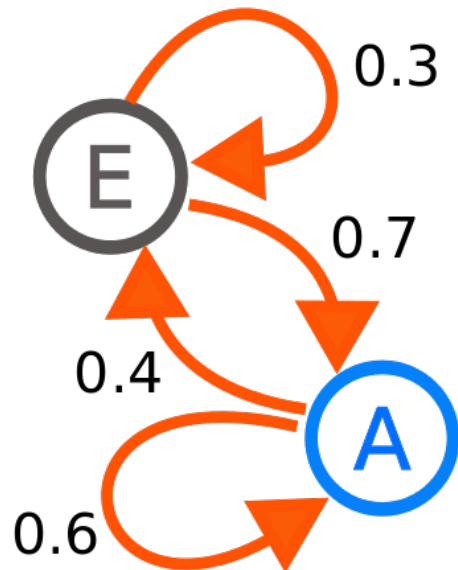
Discrete-time Markov chains



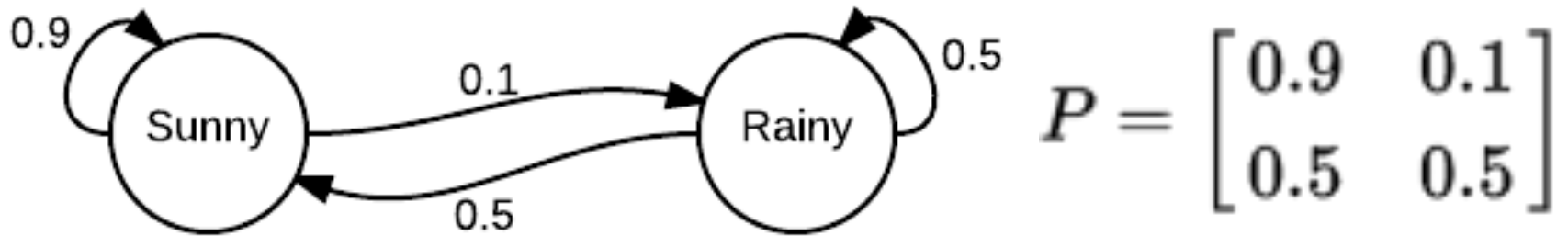
Andrei Markov
1856-1922

Discrete Time Markov Chains

- A **random** process.
- The system's state is a **discrete** variable.
- It undergoes transitions between states at uniformly-spaced (**discrete**) time points.
- Transition probabilities do not depend on the previous history of states (**memorylessness**).



First example



Assume a deterministic initial condition:

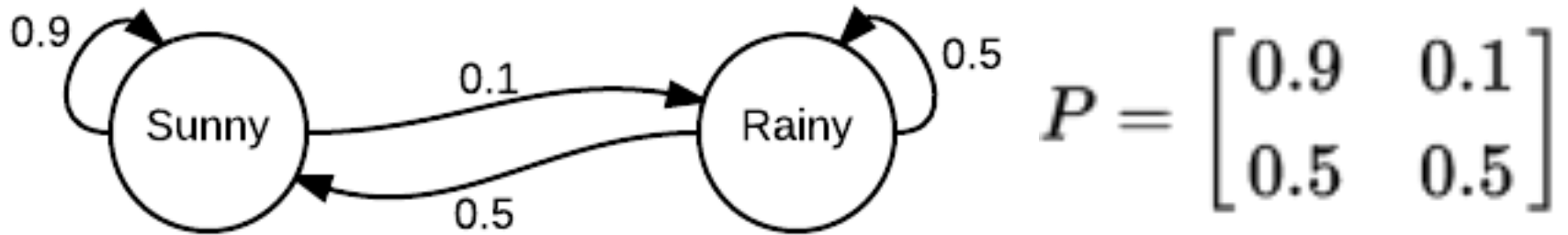
$X_0 = \text{Sunny}$ with certainty; i.e.,

$p(\text{Sunny} \mid t=0)=1$ and
 $p(\text{Rainy} \mid t=0)=0$ i.e.

$$s_0 = [1, 0]$$

...now, what is s_1 ?

First example



What is s_1 ?

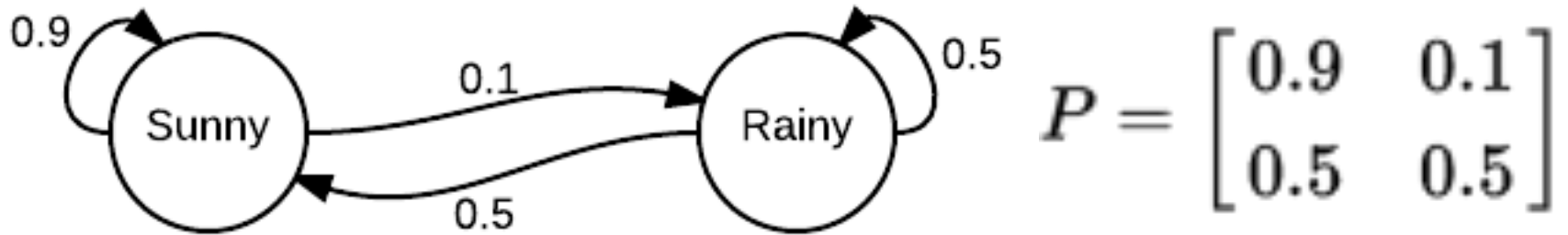
$$s_1 \left\{ \begin{array}{l} p(\text{Sunny} \mid t=1) = 0.9 \\ p(\text{Rain} \mid t=1) = 0.1 \end{array} \right.$$

In matrix form...

$$s_1 = s_0 P$$

...now, what is s_2 ?

First example



What is s_2 ?

$$s_2 \begin{cases} p(\text{Sunny} \mid t=2) = 0.9 p(\text{Sunny} \mid t=1) + 0.5 p(\text{Rainy} \mid t=1) = 0.86 \\ p(\text{Rainy} \mid t=2) = 0.1 p(\text{Sunny} \mid t=1) + 0.5 p(\text{Rainy} \mid t=1) = 0.14 \end{cases}$$

In matrix form...

$$s_2 = s_1 P$$

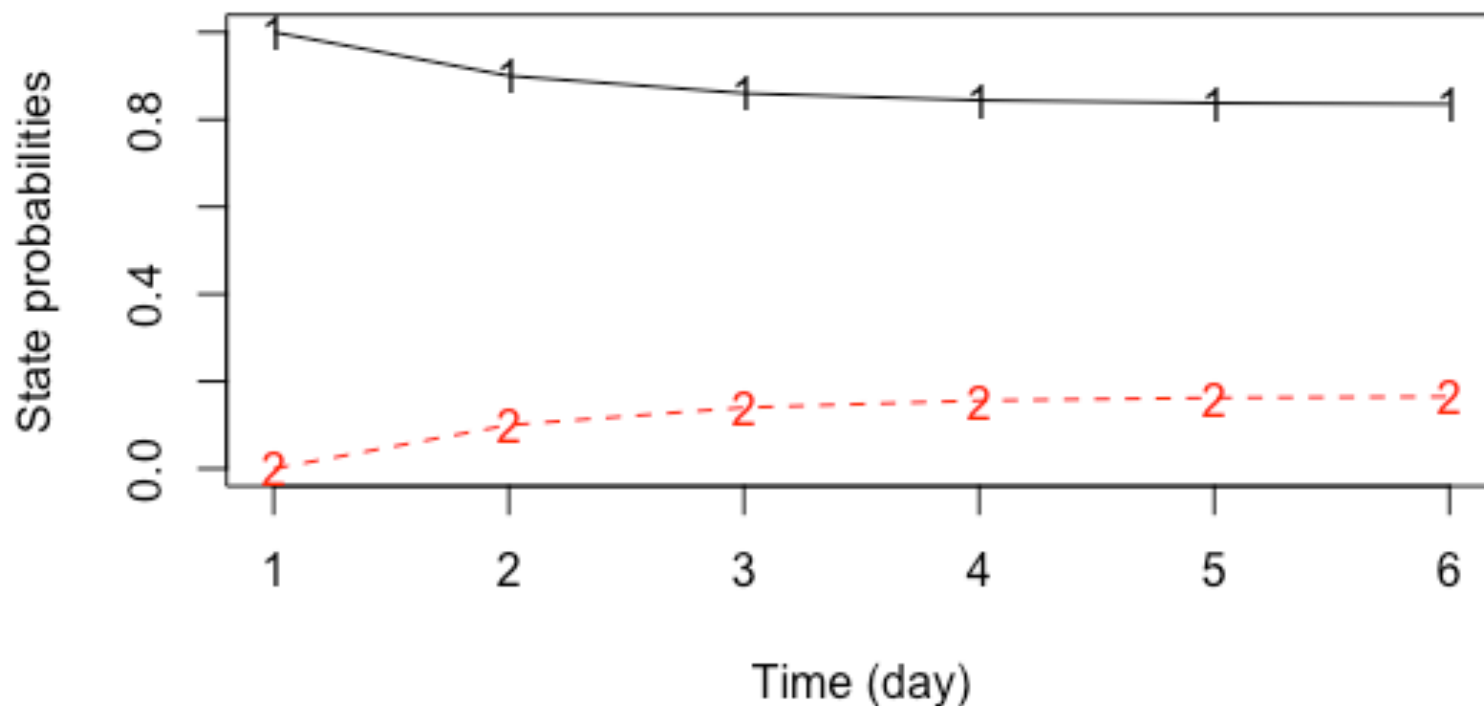
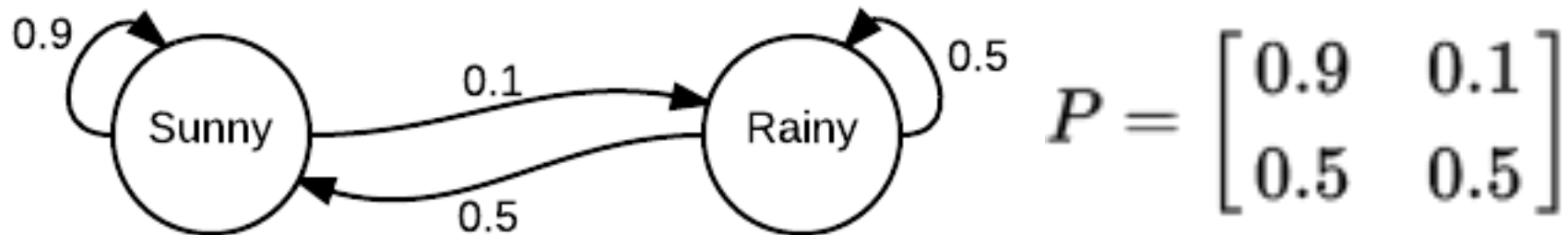
And in general...

$$s_{t+1} = s_t P$$

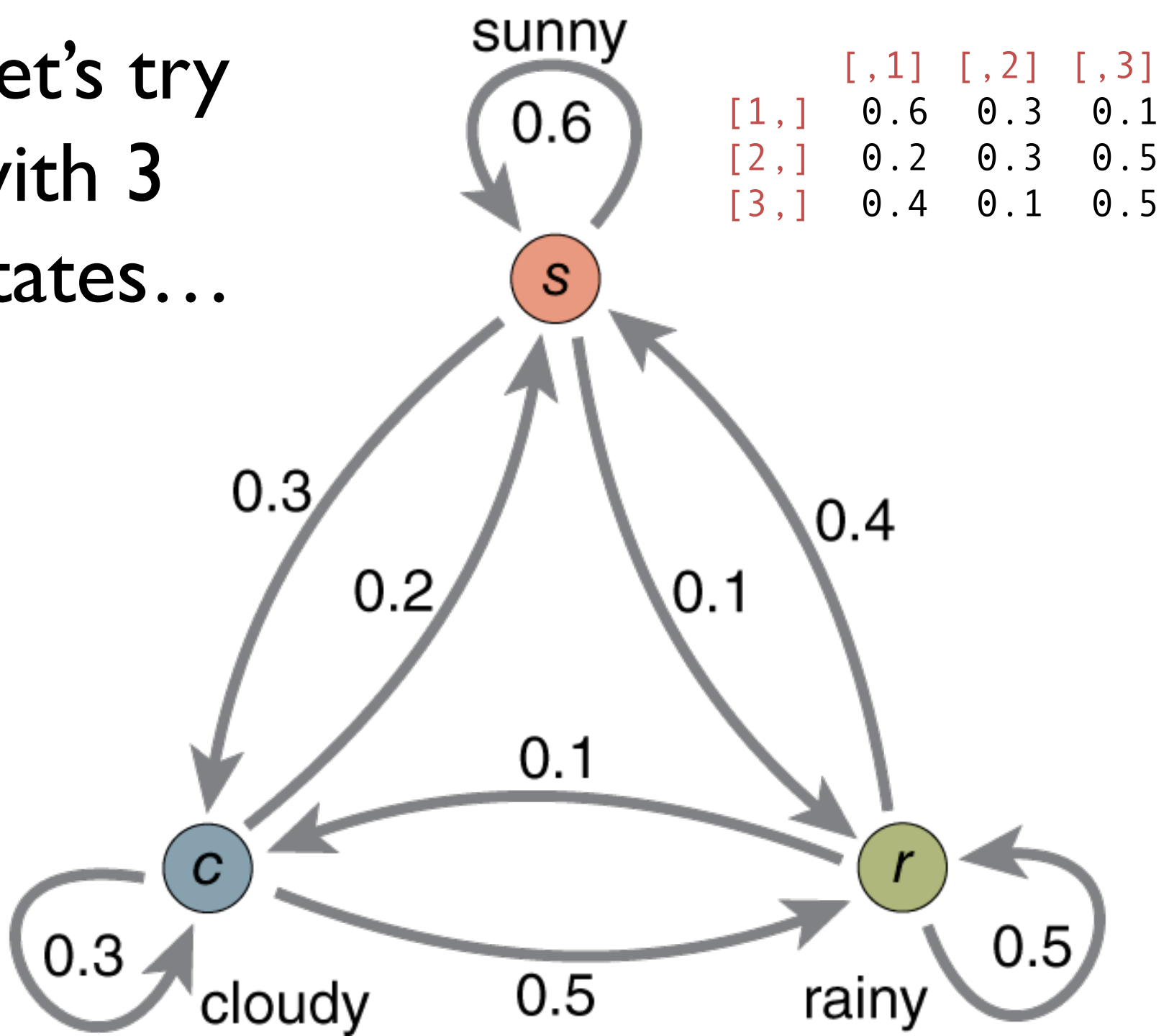
Meaning...

$$s_t = s_0 P^t$$

Let's do a numerical test...



Let's try
with 3
states...

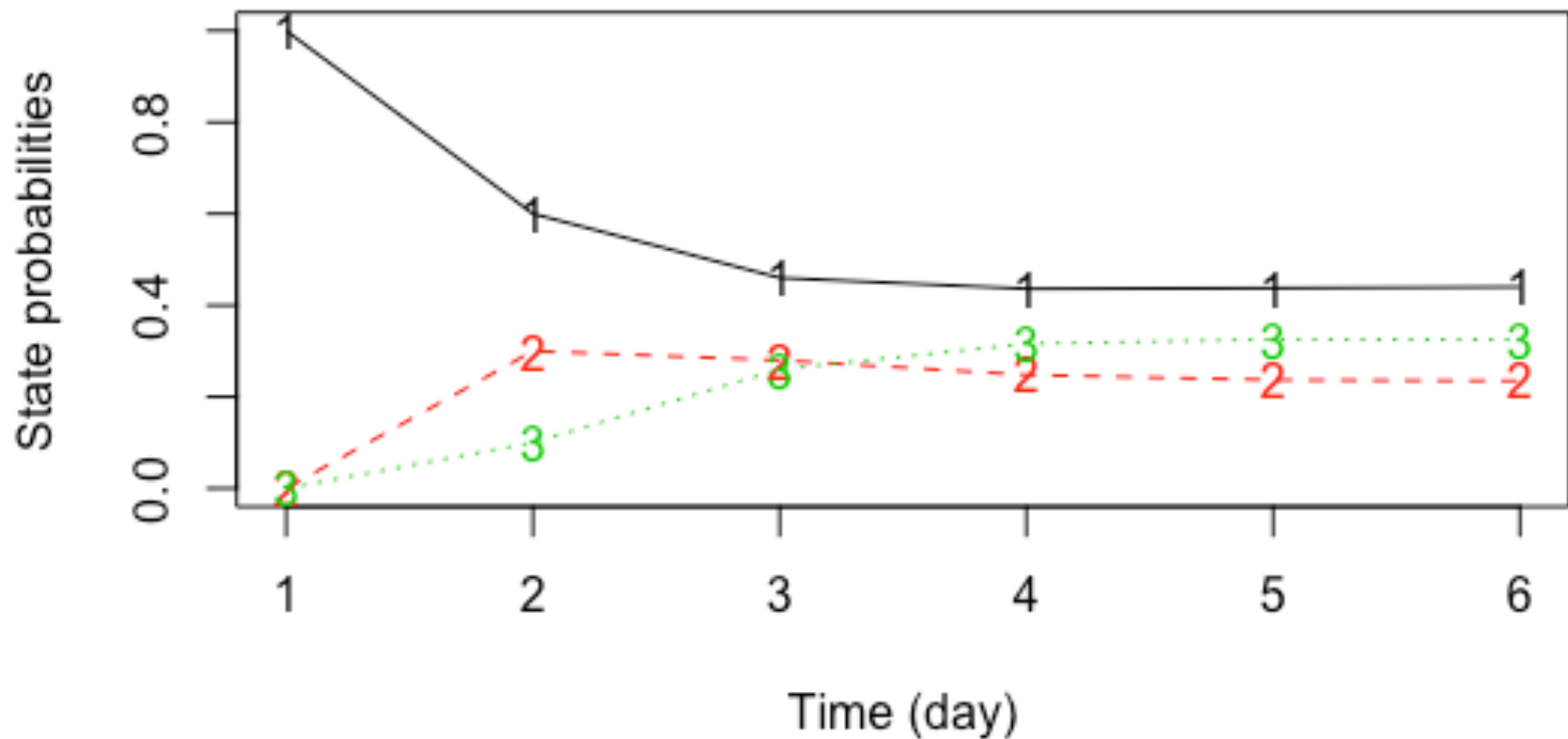


Starting from
Sunny...



	[, 1]	[, 2]	[, 3]
[1,]	0.6	0.3	0.1
[2,]	0.2	0.3	0.5
[3,]	0.4	0.1	0.5

Initial state: $s_0 = [1, 0, 0]$

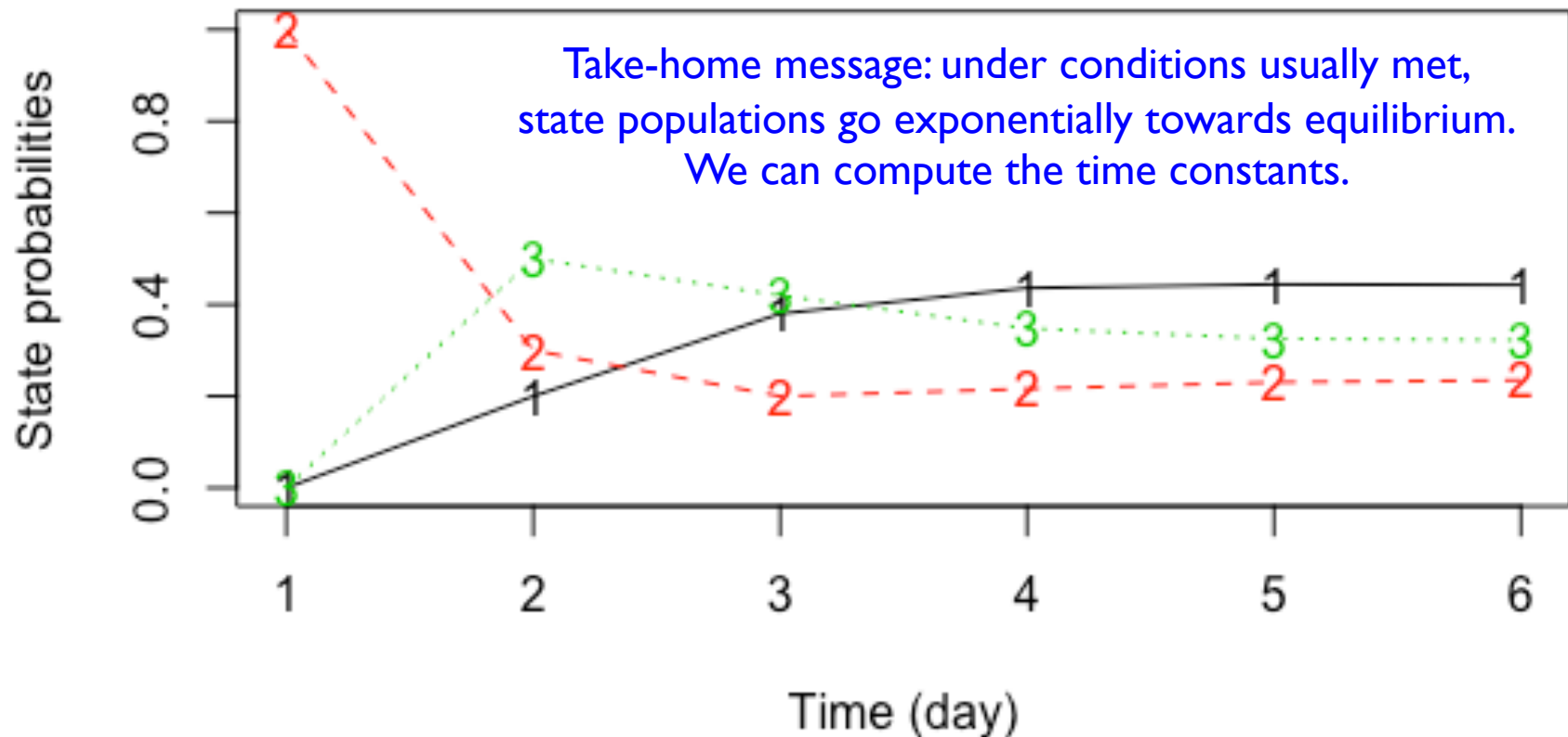


Starting from
Cloudy...



	[, 1]	[, 2]	[, 3]
[1,]	0.6	0.3	0.1
[2,]	0.2	0.3	0.5
[3,]	0.4	0.1	0.5

Initial state: $s_0 = [0, 1, 0]$



Homogeneity

- If the transition probabilities do not change with time, we have an *homogeneous* Markov chain
- Example of non-homogeneous MC: weather transition probabilities Markovian, but dependent on the season.

Important quantities we can compute

- Stationary distribution (\rightarrow eq. probabilities)
- Relaxation times
- Mean-first passage times (\rightarrow kinetic rates)
- And others we won't discuss:
 - Committor probabilities
 - Fluxes
 - ...

**Learning matrices
from trajectories
(of discrete states)**

Markov models

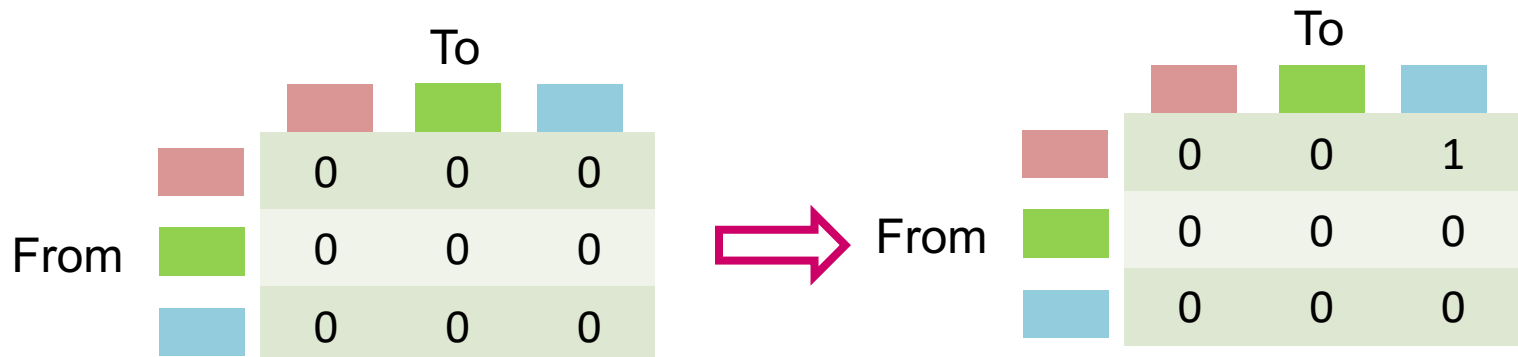
Total sampled time (e.g. 100 μs)

Define the *discrete* state of the system in *discrete time* (e.g. via reaction coordinates)



Lag time τ (here: 4 time units)

Compute the transition probability matrix
sliding a window of lag time τ



Markov models

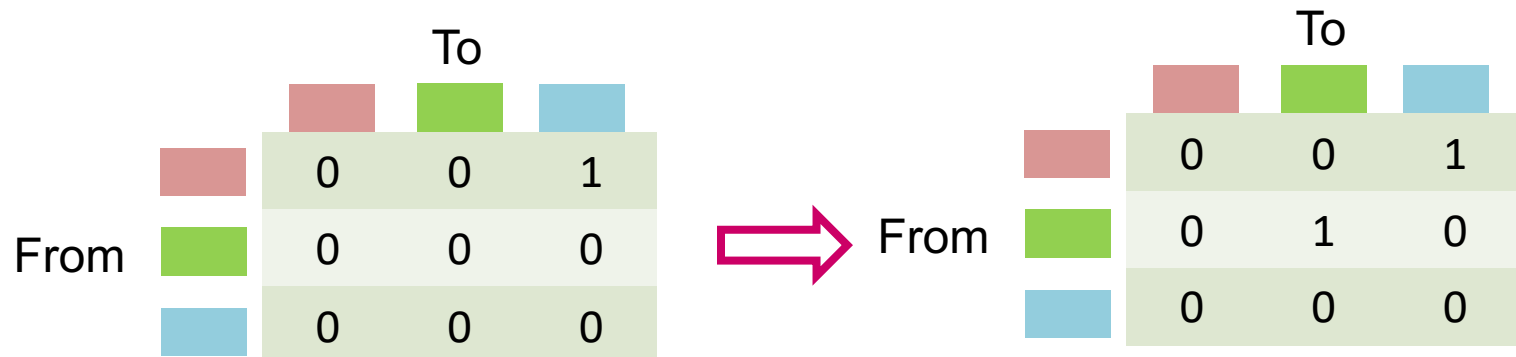
Total sampled time (e.g. 100 μs)

Define the *discrete* state of the system in *discrete time* (e.g. via reaction coordinates)



Lag time τ (here: 4 time units)

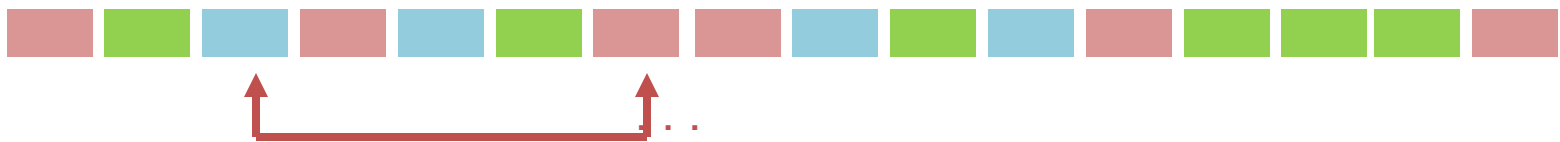
Compute the transition probability matrix
sliding a window of lag time τ



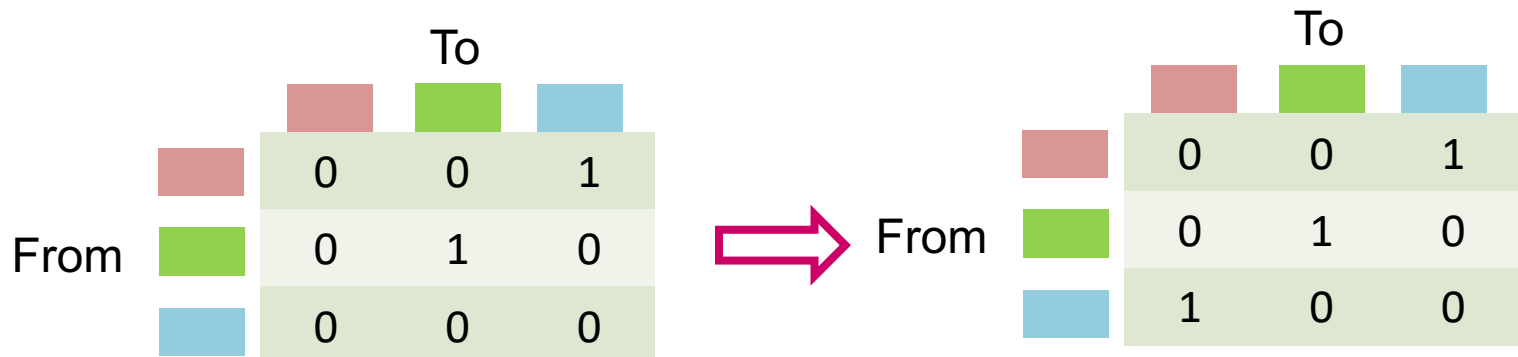
Markov models

Total sampled time (e.g. 100 μs)

Define the *discrete* state of the system in *discrete time* (e.g. via reaction coordinates)



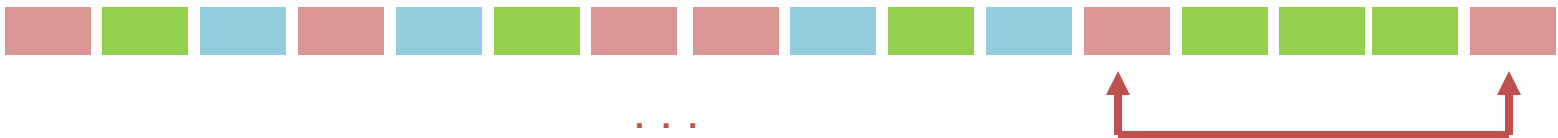
Compute the transition probability matrix
sliding a window of lag time τ



Markov models


Total sampled time (e.g. 100 μs)

Define the *discrete* state of the system in *discrete time* (e.g. via reaction coordinates)



Lag time τ (here: 4 time units)

Compute the transition probability matrix
sliding a window of lag time τ



		To		
From	red	3	0	2
	green	1	3	0
	blue	1	2	1

Transition counts

		To		
From				

Normalize
by rows

Transition probabilities

		To			Σ_j
From					

P_{ij}

Probability vector

x

		To		
From				

P_{ij}

Evolved (after τ) state

s'_j

Probability vector



s_i

x

$$\begin{bmatrix} 3/5 & 0 & 2/5 \\ 1/4 & 3/4 & 0 \\ 1/4 & 2/4 & 1/4 \end{bmatrix}$$

P_{ij}

Evolved state

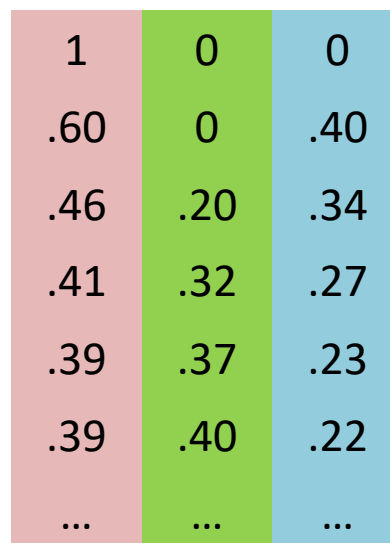


s'_j

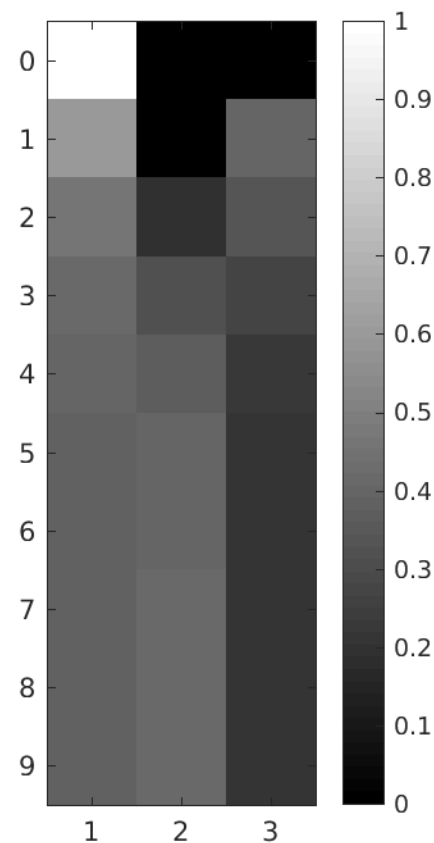
$$s' = sP$$

$$s'' = (sP)P = sP^2$$

$$s^{(n)} = sP^n$$



time



Probability vector

1	0	0
---	---	---

s_i

x

3/5	0	2/5
1/4	3/4	0
1/4	2/4	1/4

P_{ij}

Evolved state (after tau)

3/5	0	2/5
-----	---	-----

s'_j

$$s^\infty P = s^\infty$$

$$s' = sP$$

$$s'' = (sP)P = sP^2$$

$$s^{(n)} = sP^n$$

...

$$s^\infty = ?$$

Left eigenvector of P (eigenvalue 1)

Is the stationary state
= equilibrium probabilities
= the free energy surface

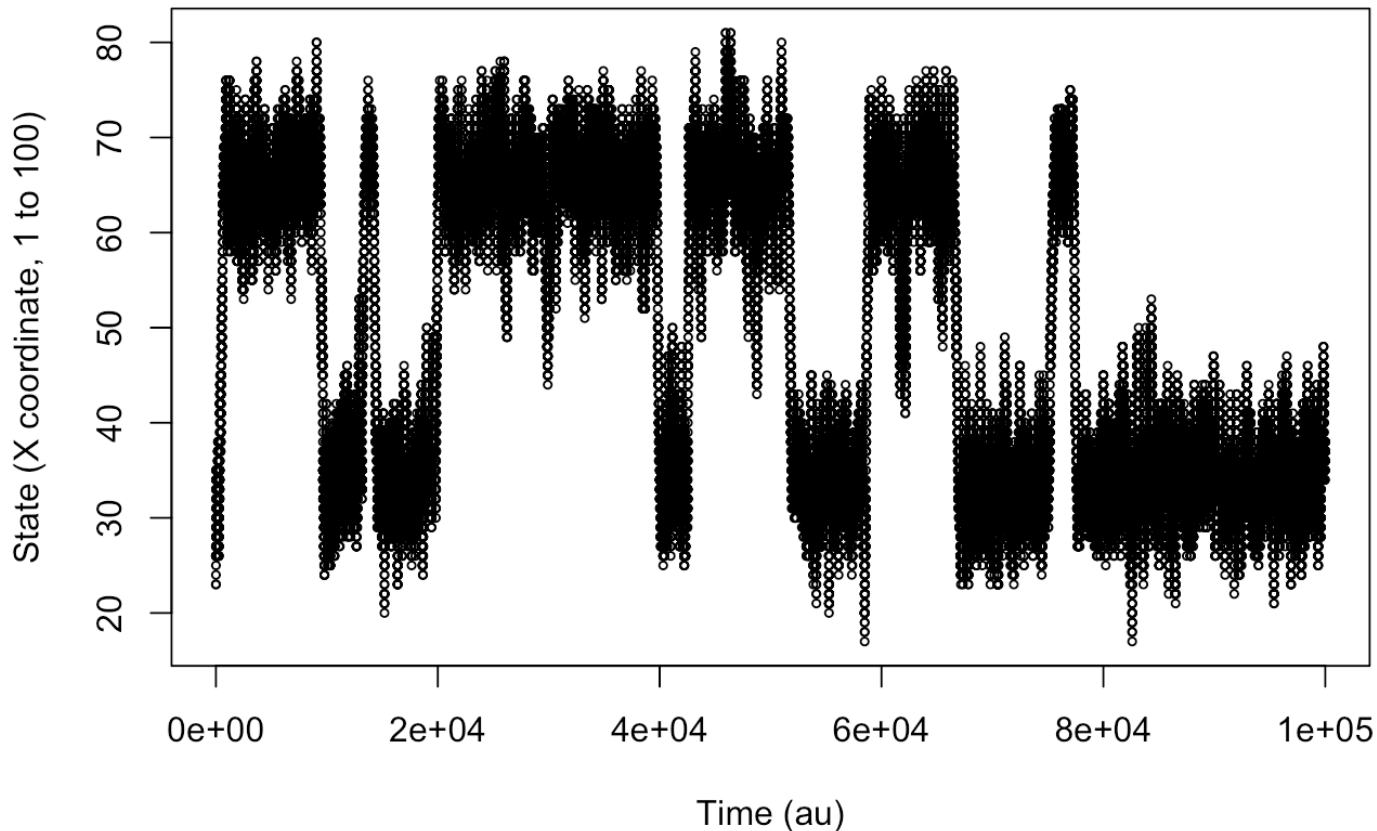
```
[a,b]=eig(P')
a(:,3)/sum(a(:,3))
= 0.385 0.410 0.205 = [5/13 16/39 8/39]
```

Markov modeling a ID trajectory

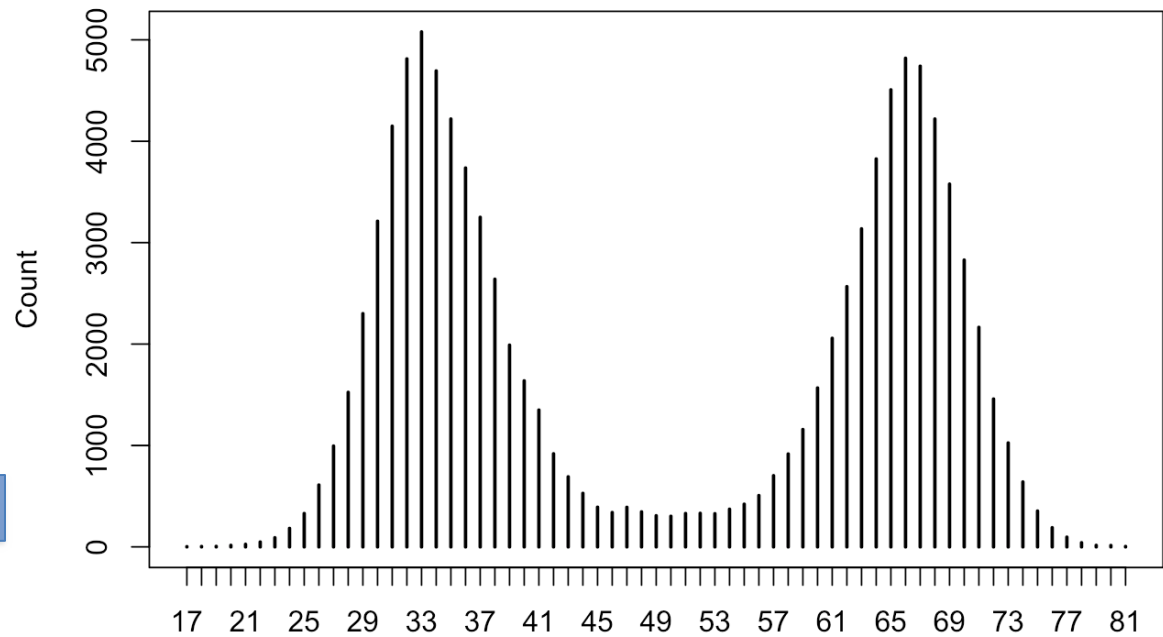
(Please find the extended version online,
“Markov state models of a ID trajectory”,
with R code at
github.com/giorginolab/Markov-Tutorial-Data)

Start with a I-D trajectory

- Already discretized in 100 bins

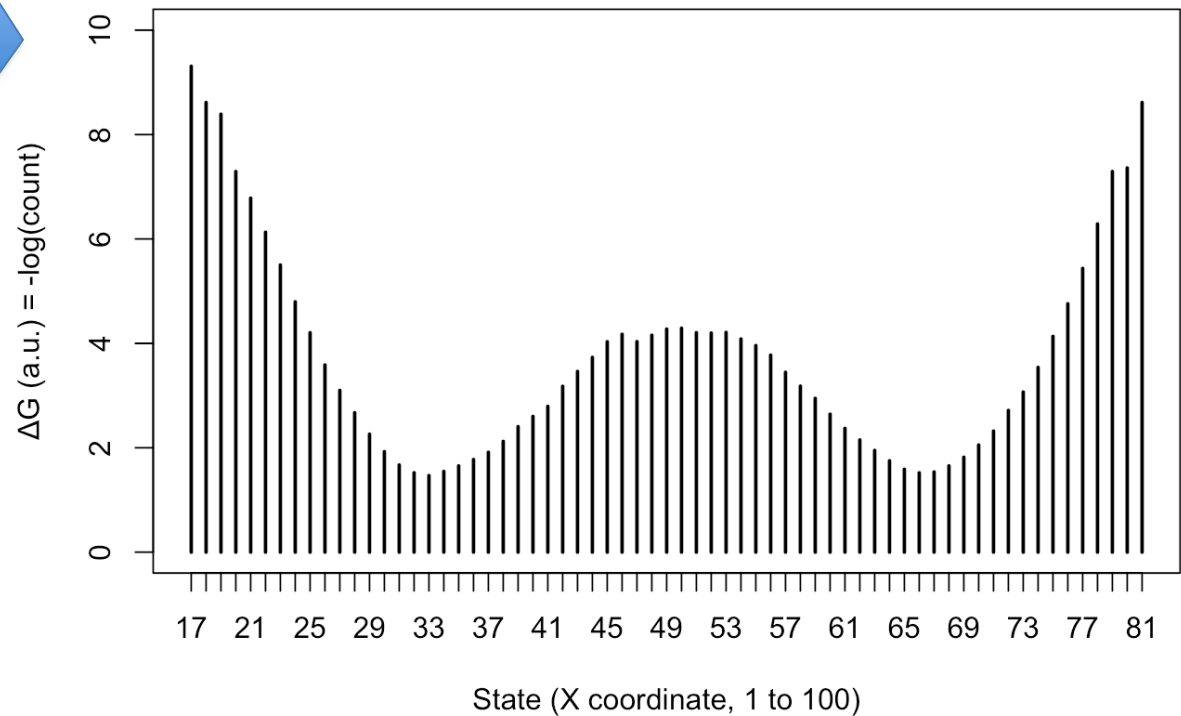


Histogram



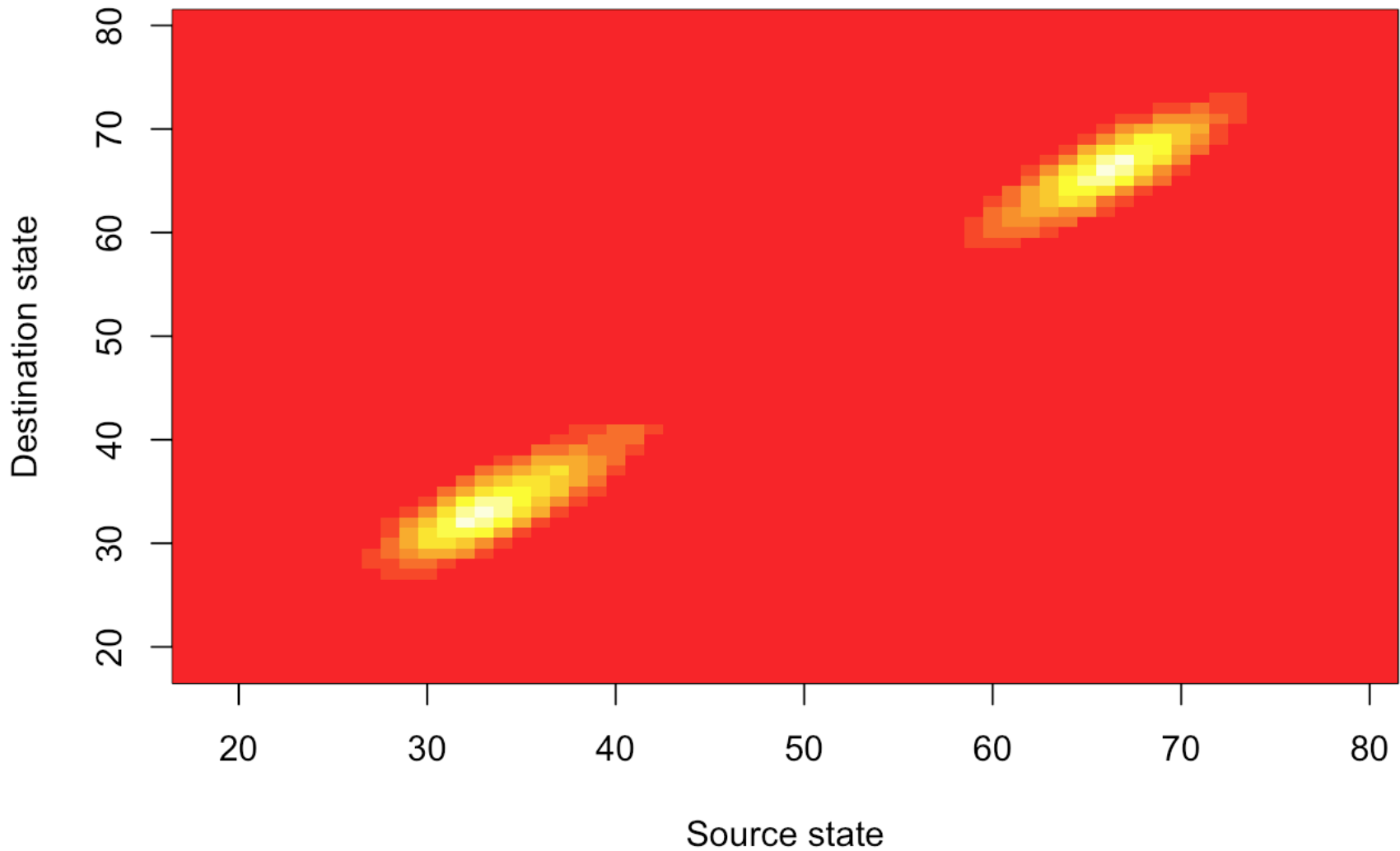
**Boltzmann
inversion**

$-\log(\text{count})$



The transition count matrix*

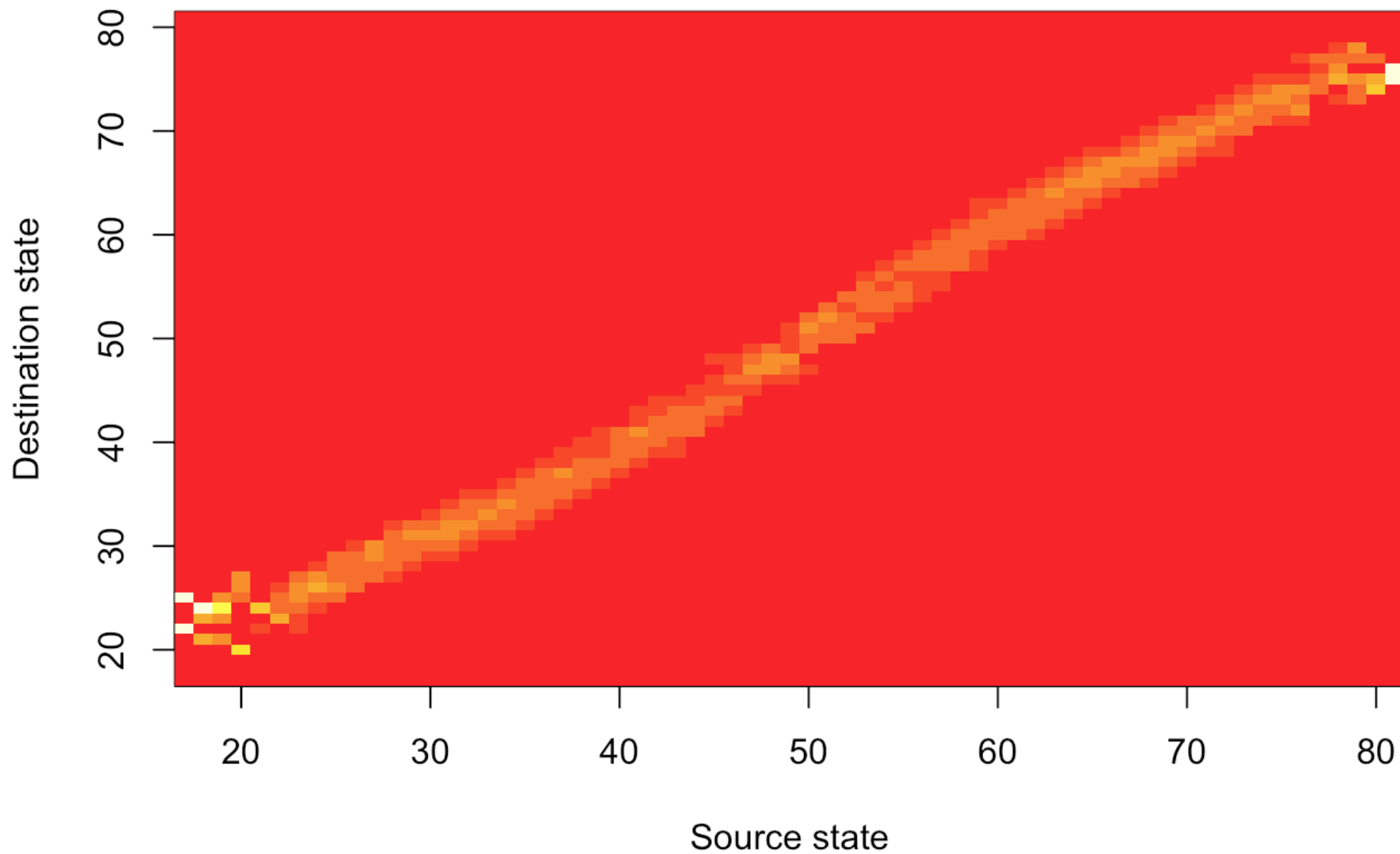
How many times we have seen state i going to j after $\tau=10$ time units



* shown as an image for compactness

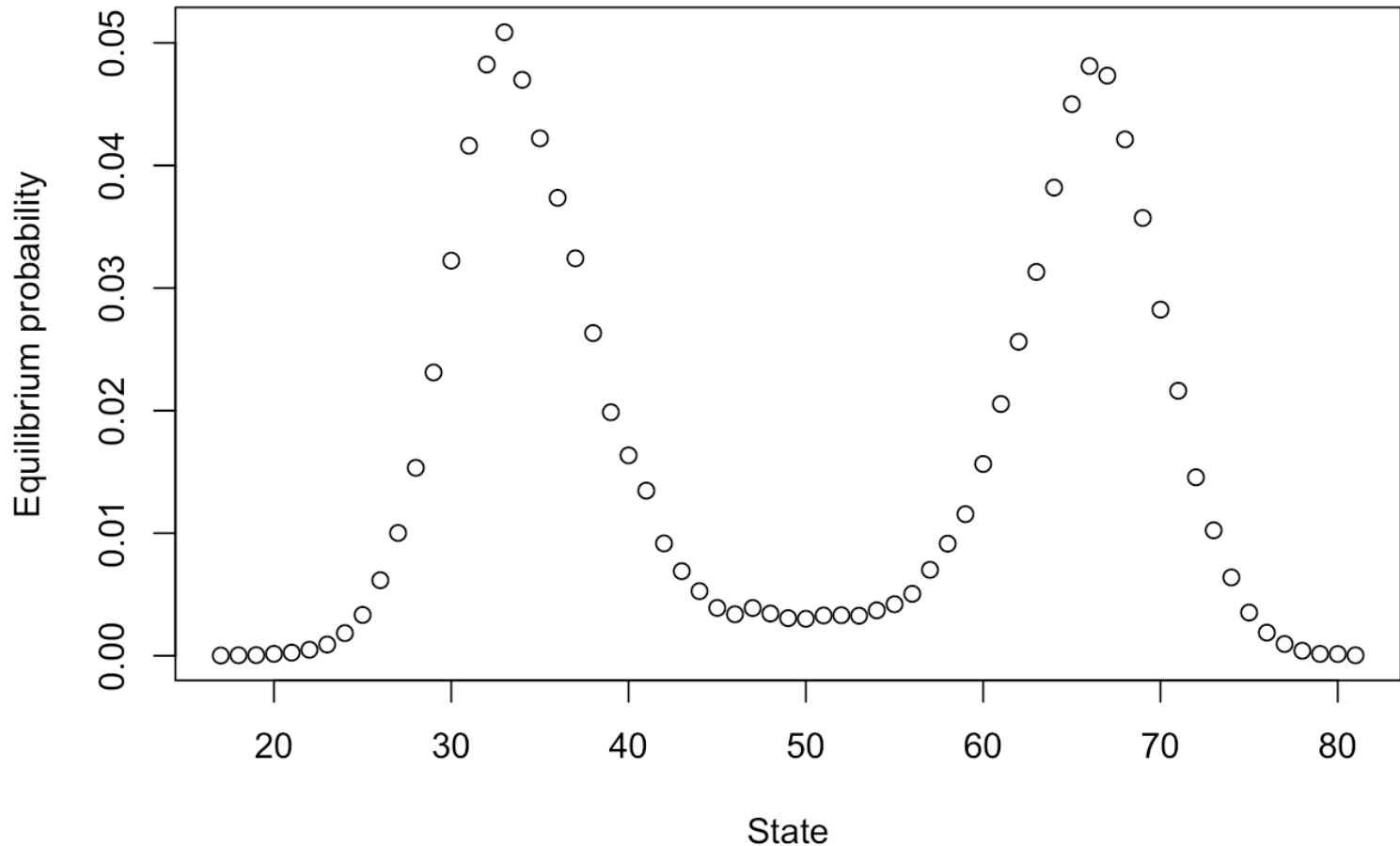
The transition *probability* matrix

Rows normalized to sum to 1



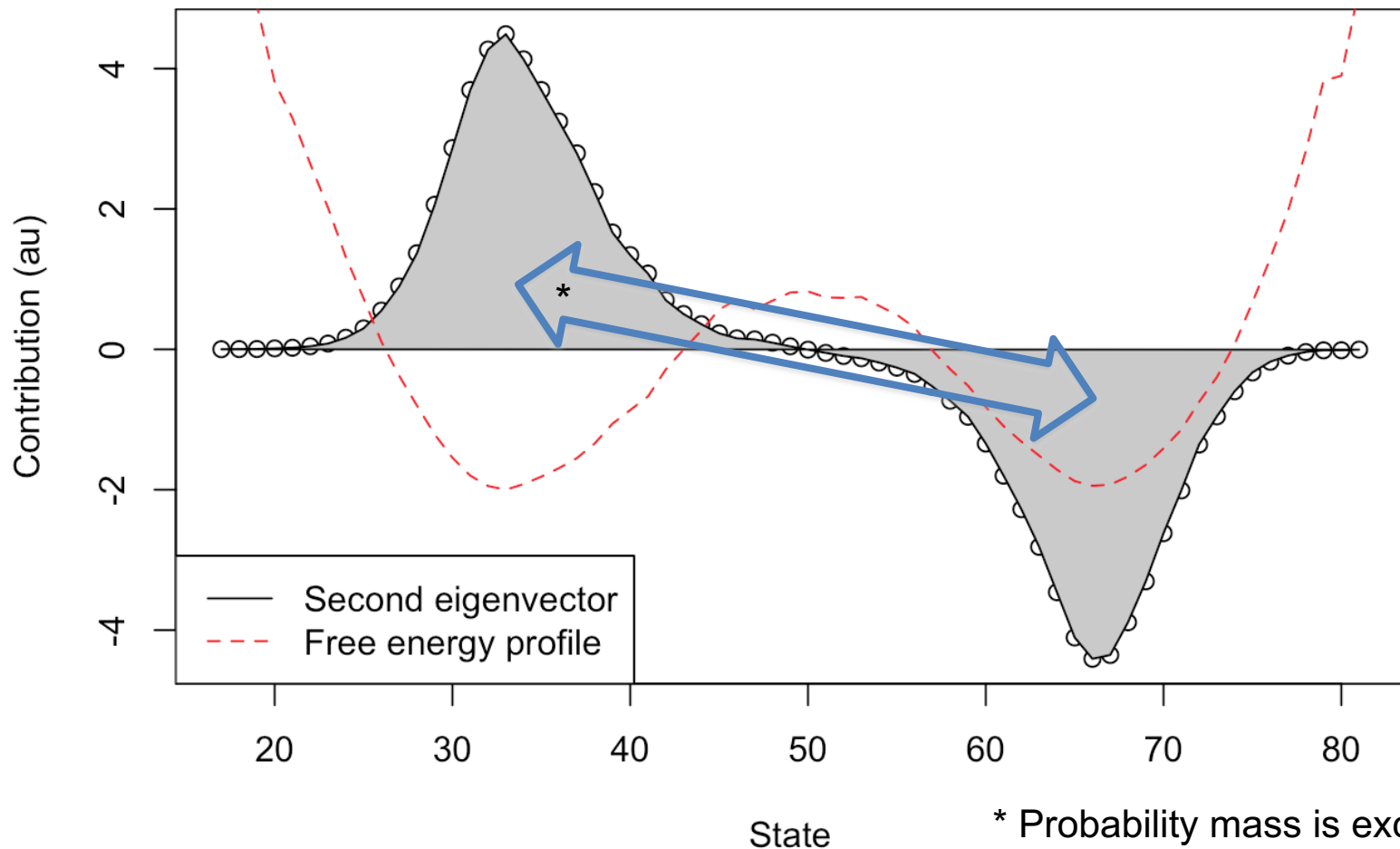
First eigenvector ($\mu_1=1$)

This is the stationary state (normalize so it sums to 1)



Second eigenvector ($\mu_2=0.997$)

This is the slowest relaxation mode: ITS $\tau_2 = 3610$ time units



Take-home message

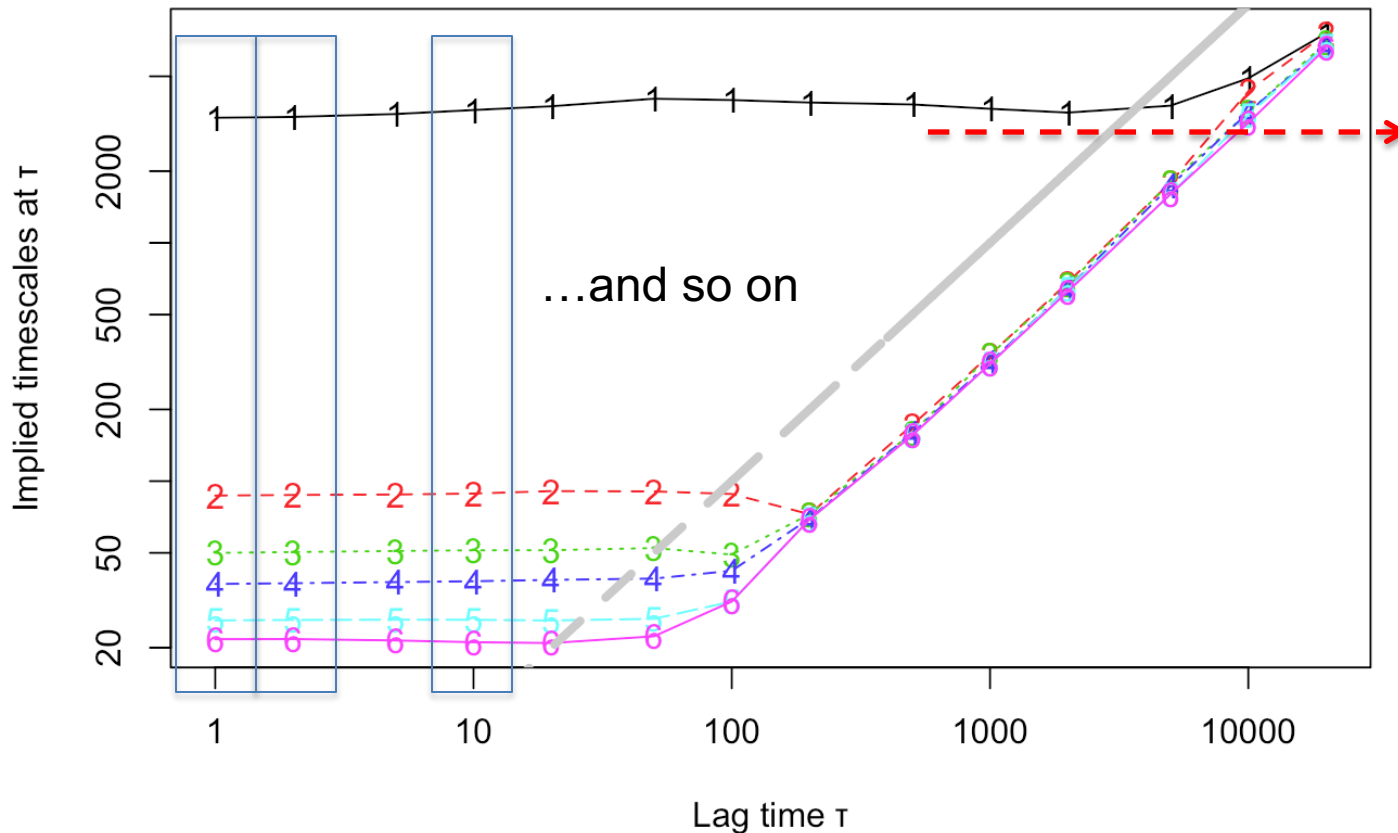
- Define states.
- Use trajectories to count transitions $\rightarrow P_{ij}$
- Eigenvalues: $1.0, \mu_1, \mu_2, \dots$
 - These are the time-scales (after $-\log$)
- Eigenvectors: $s_\infty, s_1, s_2, \dots$
 - These are the equilibrium configuration, i.e. ΔG ; and faster “oscillations” (kinetics)

Markovianity

- The state transition probabilities only depend on the current state.
- Examples
 - Today's weather, not yesterday's
 - Where the ligand is, not how did it got there
- The property may be false at short timescales but true at longer ones (system's memory)
- It does depend on the chosen states

Implied timescales plot

Repeat the eigenvalues determination for several lags. Check convergence.



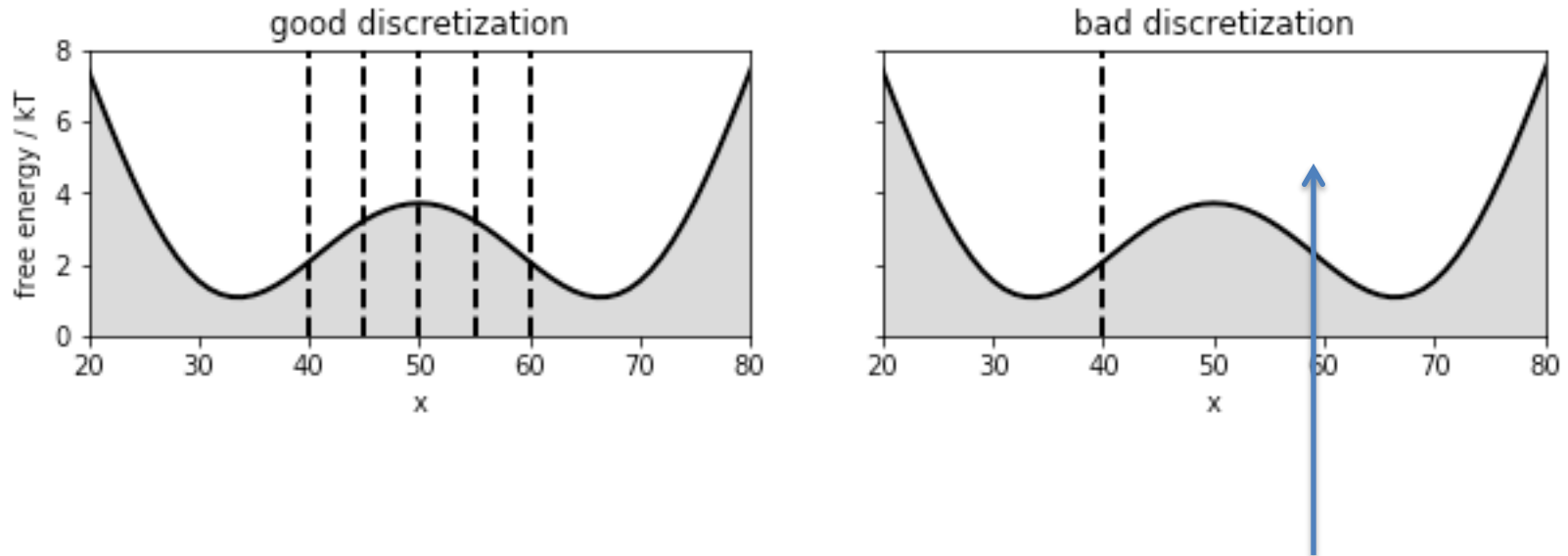
Here, convergence is achieved very early.

Reasons:

- (a) true two-state dynamics;
- (b) absence of orthogonal degrees of freedom;
- (c) fine space discretization

Macrostates

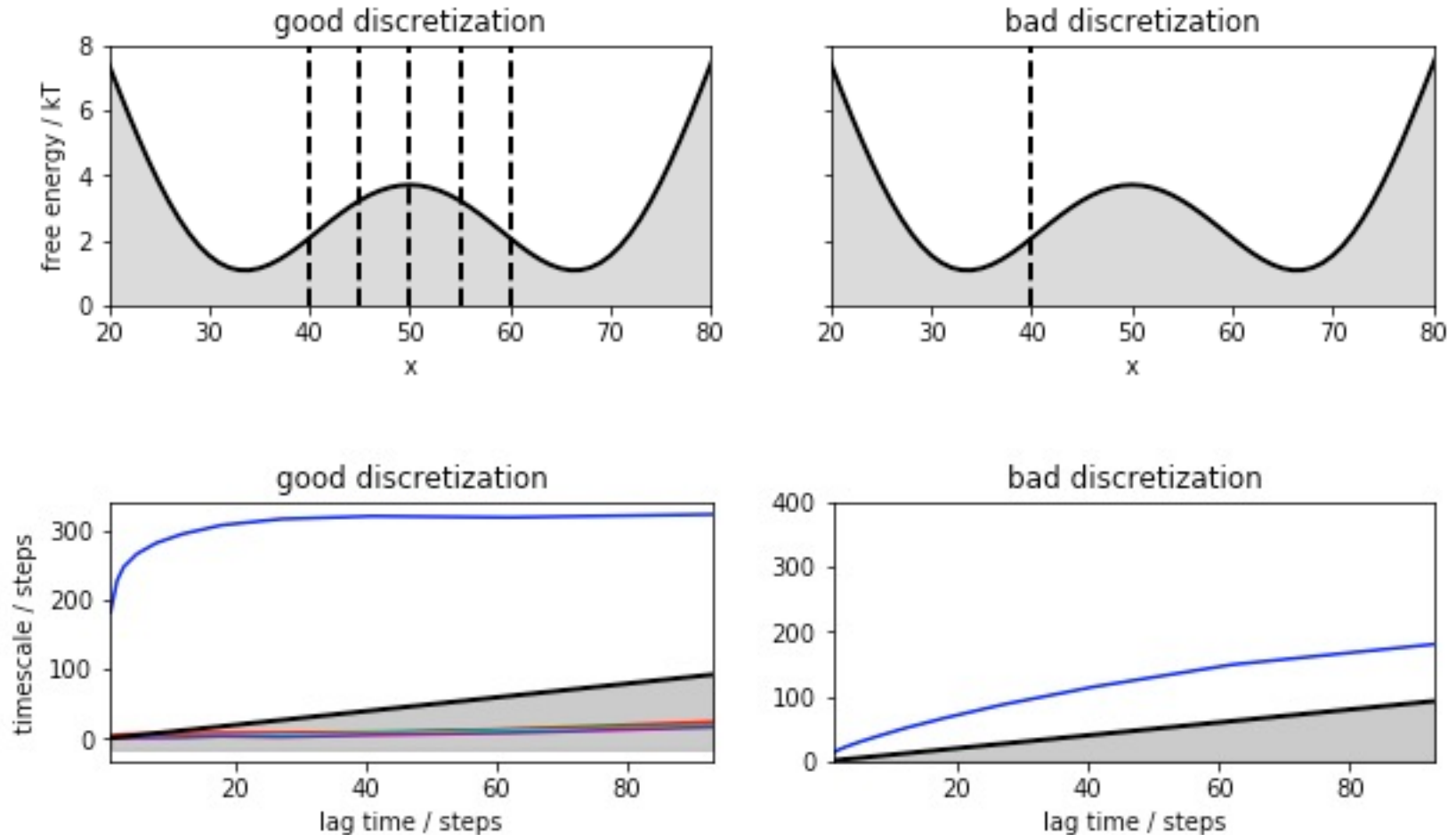
A bad choice of the discretization breaks the Markovianity assumption



In the “bad discretization” case, the barrier is embedded in one of the states. This generates a “long term memory” effect: the rightmost state could actually be short-lived (if we are on the left of the barrier) or long-lived (if we are on its right). These two cases are convoluted into the same, so that the present state information itself is not sufficient to predict the “future” of the system any more.

Macrostates

A bad choice of the discretization breaks the Markovianity assumption



End