Markov-state modeling of biomolecular systems

Toni Giorgino
National Research Council of Italy
toni.giorgino@cnr.it
www.giorginolab.it

GitHub
@giorginolab

Thesis projects available

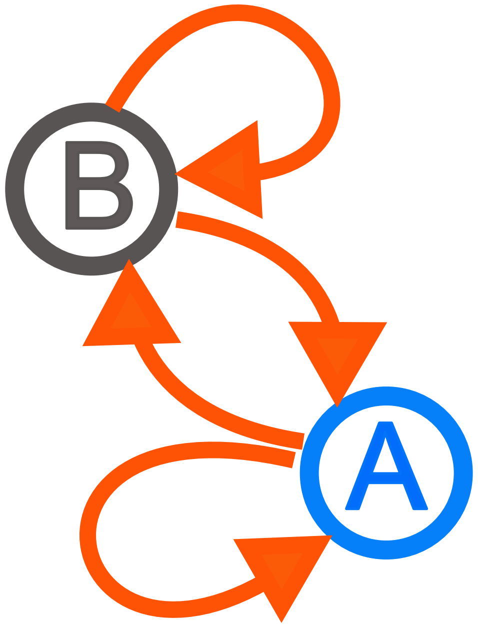
```
github.com/giorginolab/…
          …/Markov-Tutorial-BCN-2022
          …/Markov-Tutorial-Data
```

Master in Bioinformatics for Health Sciences
UPF Barcelona, 3 May 2022

Discrete-time Markov chains

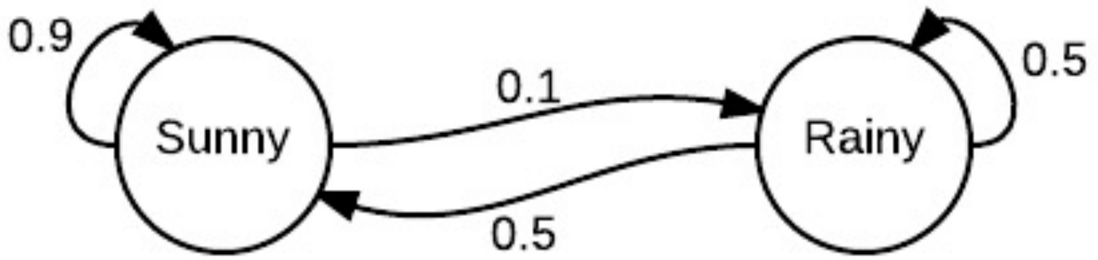
Discrete Time Markov Chains

Andrei Markov
1856-1922



- 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



$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

Assume a deterministic initial condition:

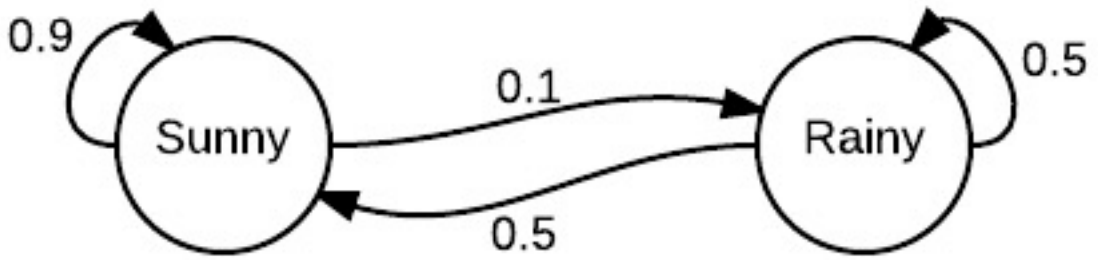
X_0 = Sunny with certainty; *i.e.*,

p(Sunny | t=0)=1 and
p(Rainy | t=0)=0 *i.e.*

…now, what is s_1?

$s_0 = [1, 0]$

First example



$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

What is s_1?

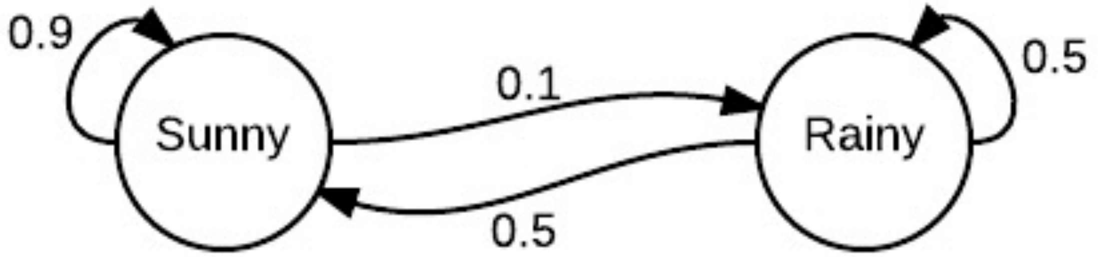
$$s_1 \left\{ \begin{array}{r} p(\text{Sunny} \mid t=1) = 0.9 \\ p(\text{Rain} \mid t=1) = 0.1 \end{array} \right.$$

In matrix form…

$$\mathbf{s}_1 = \mathbf{s}_0 \, P$$

…now, what is s_2?

First example



$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$

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…

$$\mathbf{s}_2 = \mathbf{s}_1 P$$

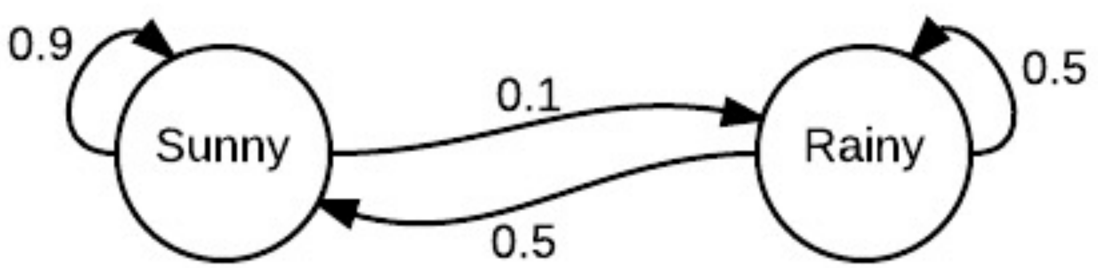
And in general…

$$\mathbf{s}_{t+1} = \mathbf{s}_t P$$

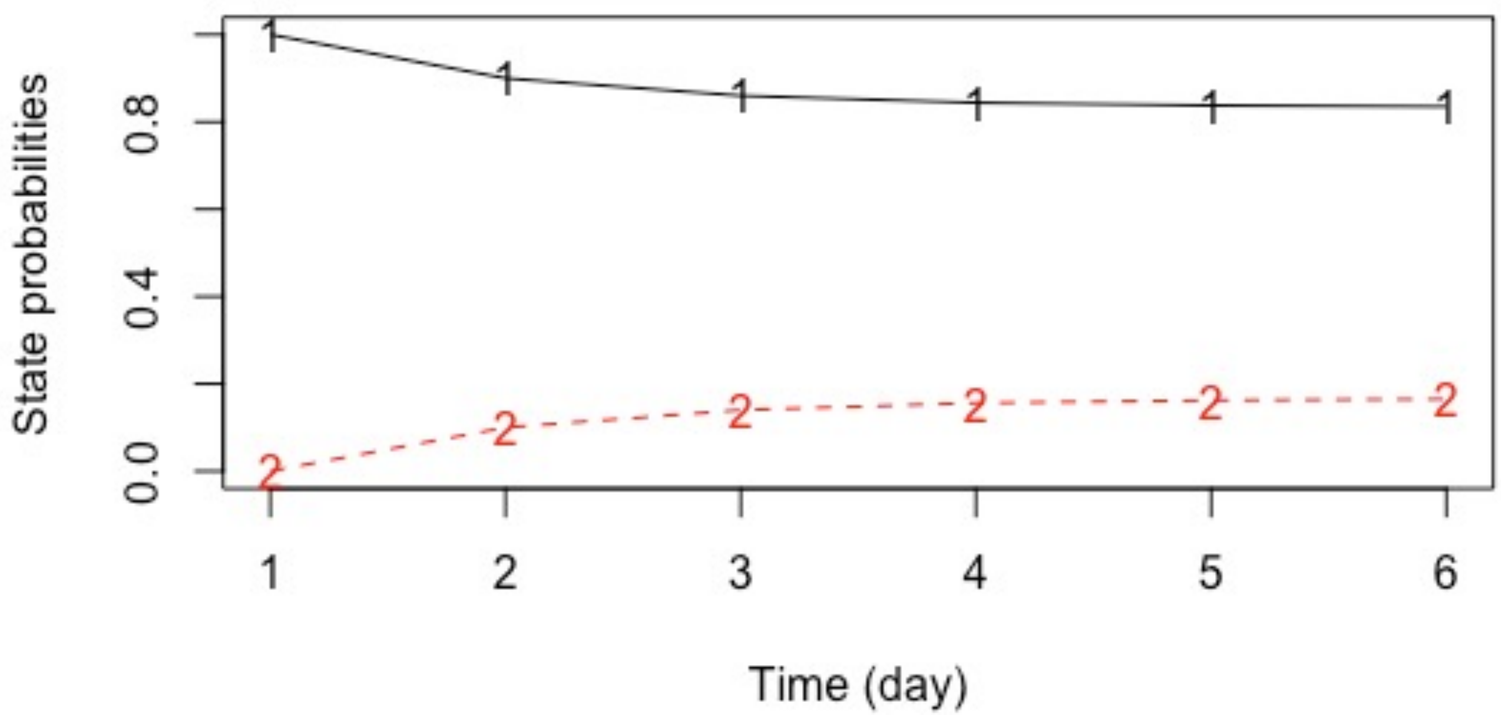
Meaning…

$$\mathbf{s}_t = \mathbf{s}_0 P^t$$

Let's do a numerical test…

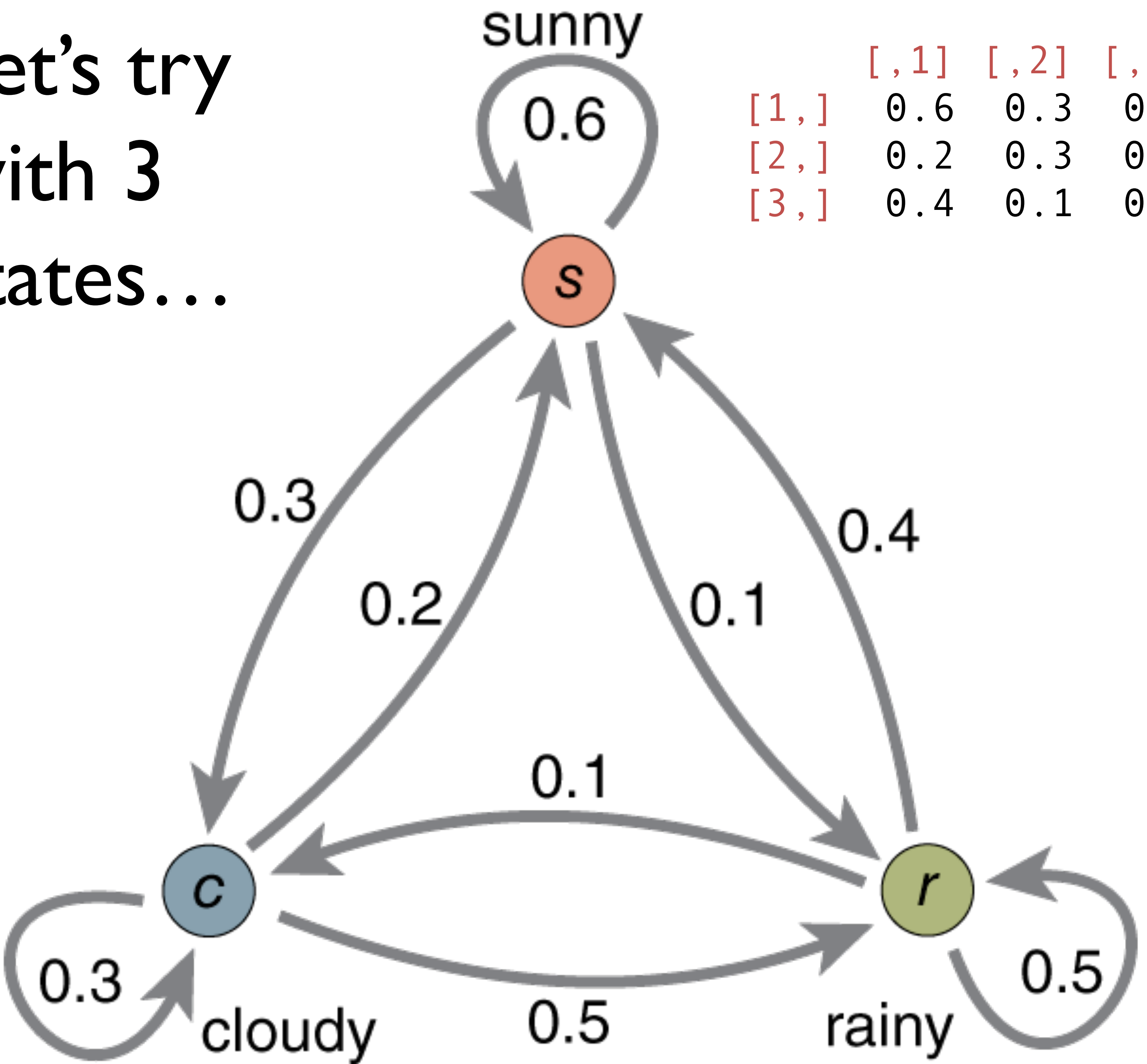


$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}$$



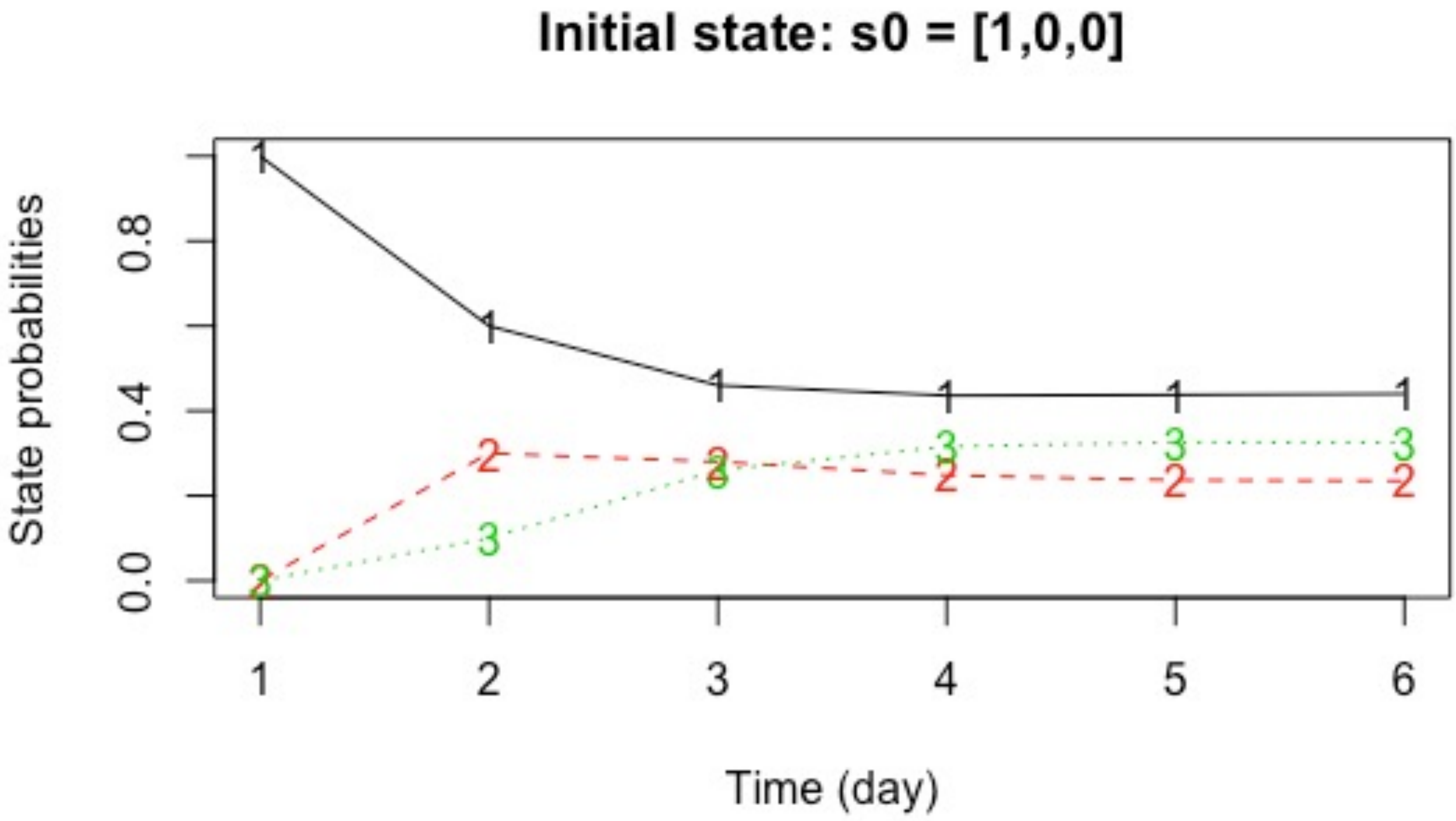
Let's try with 3 states…

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



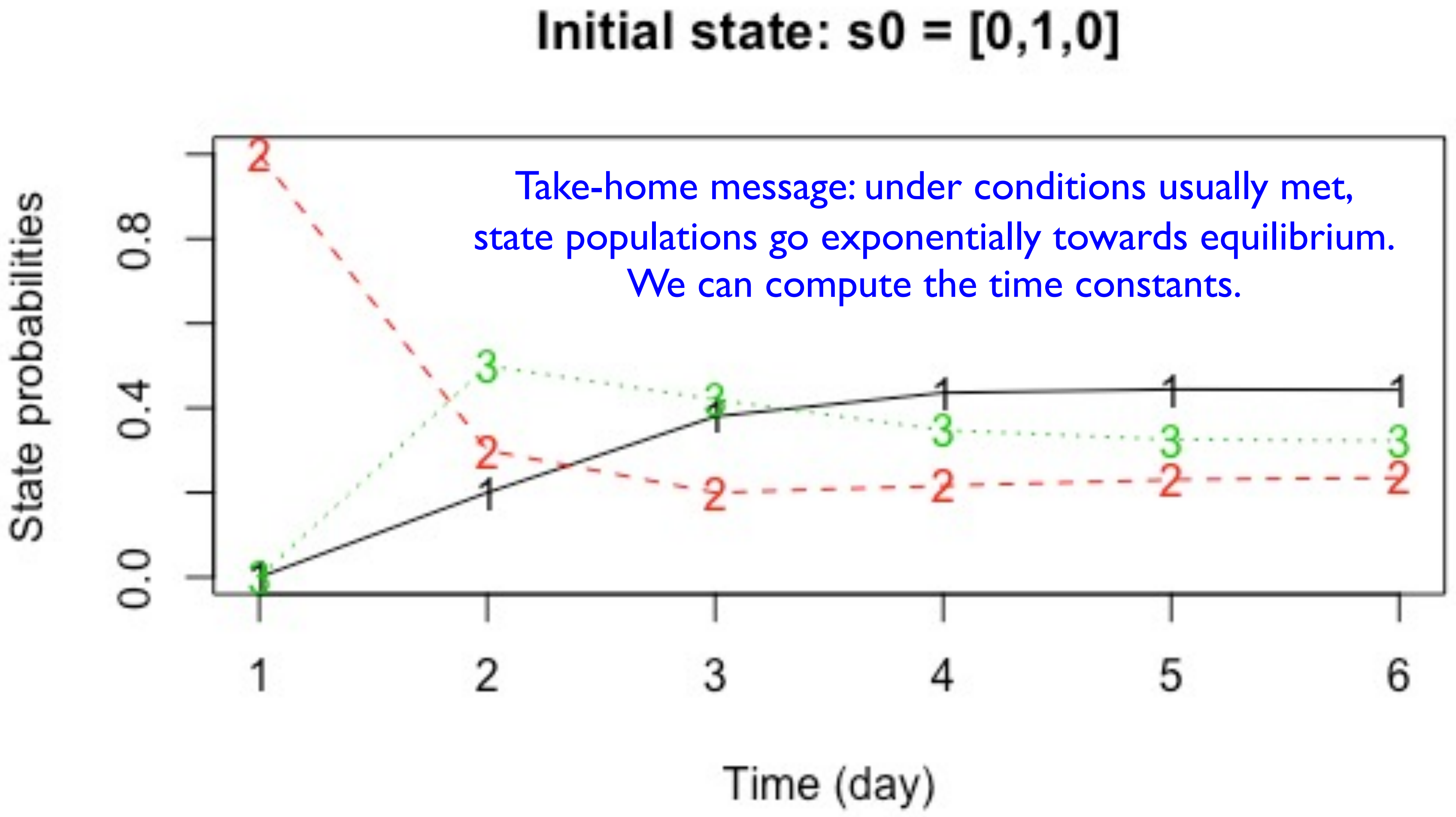
Starting from *Sunny*…

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



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
```



Take-home message: under conditions usually met, state populations go exponentially towards equilibrium. We can compute the time constants.

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

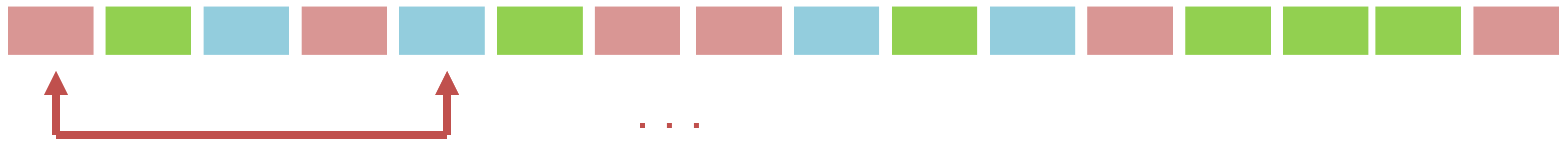
- Stationary distribution ($\rightarrow$ eq. probabilities)
- Relaxation times
- Mean-first passage times ($\rightarrow$ kinetic rates)
- And others we won't discuss:
  - Committor probabilities
  - Fluxes
  - …

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

		To		
		🟥	🟩	🟦
From	🟥	0	0	0
	🟩	0	0	0
	🟦	0	0	0

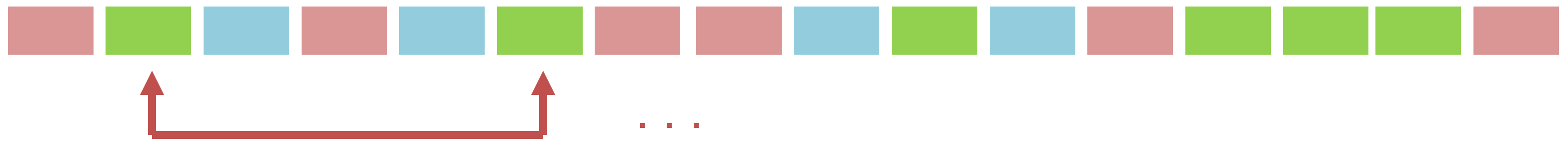
⇒

		To		
		🟥	🟩	🟦
From	🟥	0	0	1
	🟩	0	0	0
	🟦	0	0	0

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)

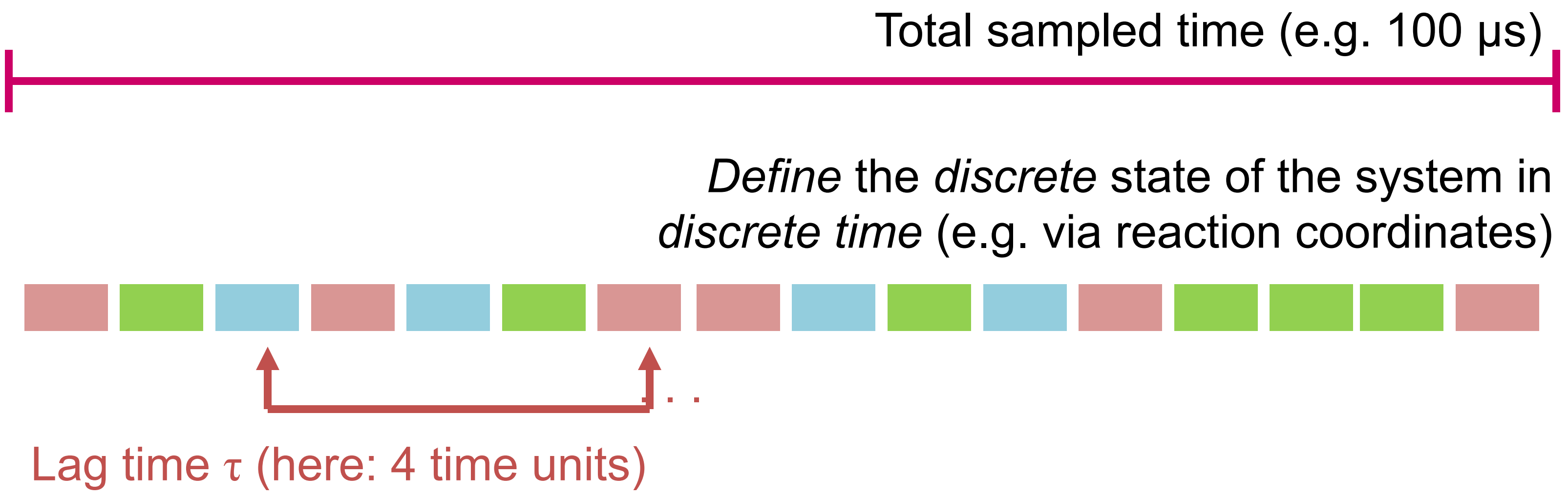
Compute the transition probability matrix sliding a window of lag time τ

		To		
		🟥	🟩	🟦
From	🟥	0	0	1
	🟩	0	0	0
	🟦	0	0	0

$\Rightarrow$

		To		
		🟥	🟩	🟦
From	🟥	0	0	1
	🟩	0	1	0
	🟦	0	0	0

Markov models


Total sampled time (e.g. 100 μs)
. . .
Lag time τ (here: 4 time units)

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

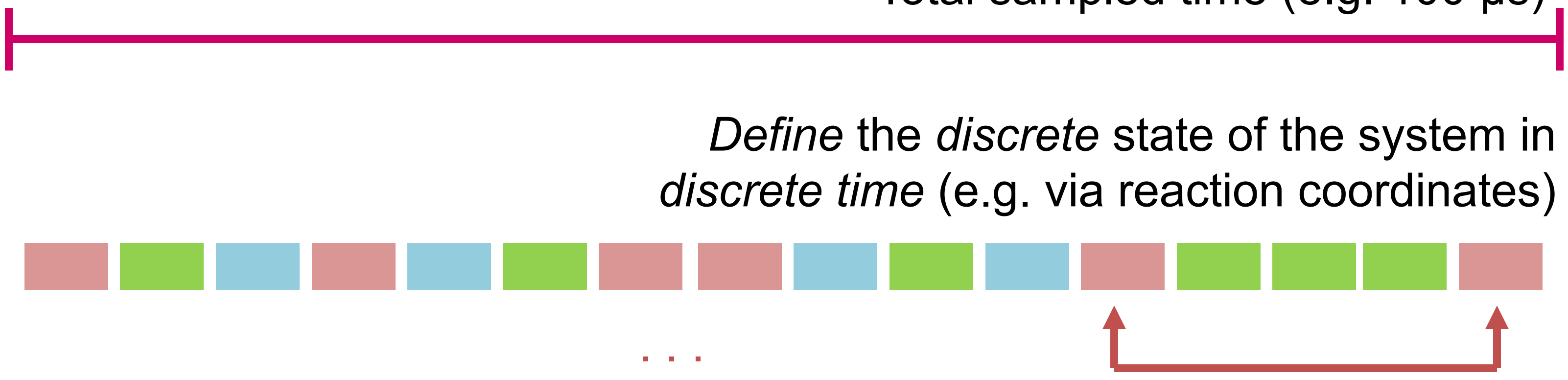
From \ To	🟥	🟩	🟦
🟥	0	0	1
🟩	0	1	0
🟦	0	0	0

$\Rightarrow$

From \ To	🟥	🟩	🟦
🟥	0	0	1
🟩	0	1	0
🟦	1	0	0

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	
		🟥	🟩	🟦
	🟥	3	0	2
From	🟩	1	3	0
	🟦	1	2	1

Transition counts

		To		
		🟥	🟩	🟦
From	🟥	3	0	2
	🟩	1	3	0
	🟦	1	2	1

$\Rightarrow$

Normalize by rows

Transition probabilities

		To			Σ_j
		🟥	🟩	🟦	
From	🟥	3/5	0	2/5	1
	🟩	¼	¾	0	1
	🟦	¼	2/4	¼	1

P_{ij}

Probability vector

Evolved (after τ) state

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \end{pmatrix}}_{s_i} \times \underbrace{\begin{pmatrix} 3/5 & 0 & 2/5 \\ ¼ & ¾ & 0 \\ ¼ & 2/4 & ¼ \end{pmatrix}}_{P_{ij}} = \underbrace{\begin{pmatrix} 3/5 & 0 & 2/5 \end{pmatrix}}_{s'_j}$$

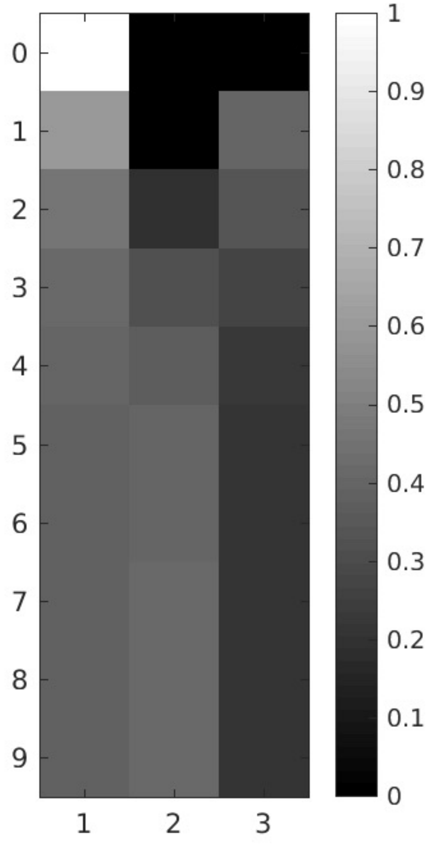
Probability vector | **Evolved state**

$$\underset{s_i}{\begin{bmatrix}1 & 0 & 0\end{bmatrix}} \times \underset{P_{ij}}{\begin{bmatrix} 3/5 & 0 & 2/5 \\ 1/4 & 3/4 & 0 \\ 1/4 & 2/4 & 1/4 \end{bmatrix}} = \underset{s'_j}{\begin{bmatrix}3/5 & 0 & 2/5\end{bmatrix}}$$

$$s' = sP$$
$$s'' = (sP)P = sP^2$$
$$s^{(n)} = sP^n$$

1	0	0
.60	0	.40
.46	.20	.34
.41	.32	.27
.39	.37	.23
.39	.40	.22
...	...	...

time $\downarrow$



Probability vector

1	0	0

s_i

x

3/5	0	2/5
¼	¾	0
¼	2/4	¼

P_{ij}

=

Evolved state (after tau)

3/5	0	2/5

s'_j

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

$$s' = sP$$
$$s'' = (sP)P = sP^2$$
$$s^{(n)} = sP^n$$
$$\dots$$
$$s^\infty = \quad ?$$

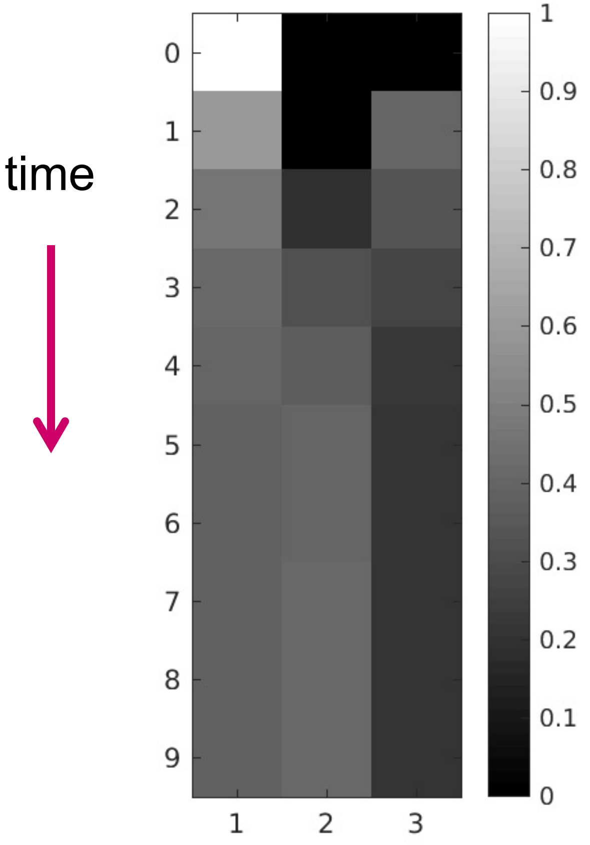
Left eigenvector of P (eigenvalue 1)

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]
```

Relaxation towards equilibrium

time



Equilibrium is reached within typical **relaxation times** T_k.

$$\mu_k = e^{-\tau/T_k}$$

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

- called *implied timescales*
- computed from the eigenvalues $\mu_k < 1$
- depend on τ

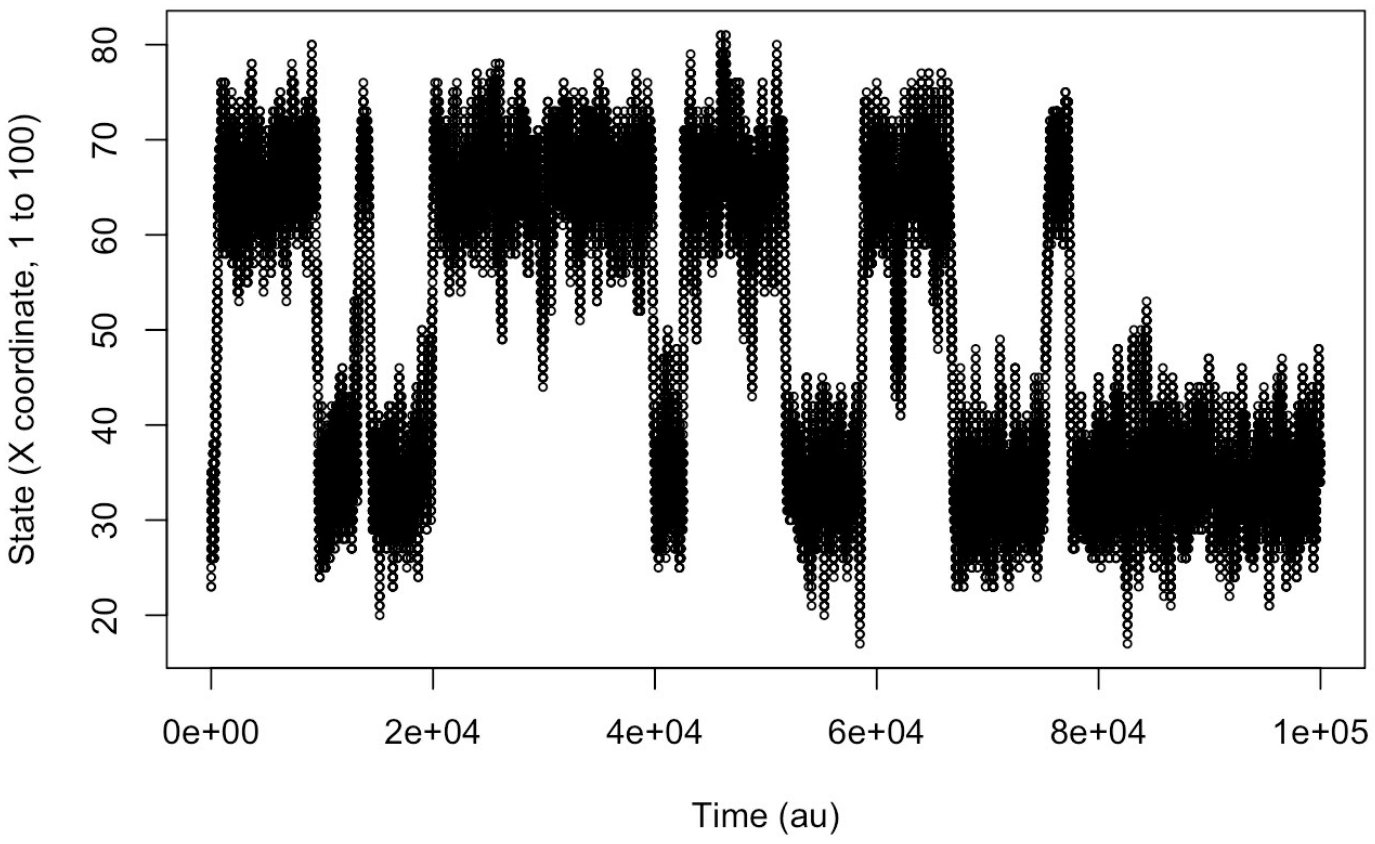
(Stationary state: $T_1 = \infty \rightarrow \mu_1 = 1.0$)

Markov modeling a 1D trajectory

(Please find the extended version online,
“Markov state models of a 1D trajectory”,
with R code at
github.com/giorginolab/Markov-Tutorial-Data)

Start with a 1-D trajectory

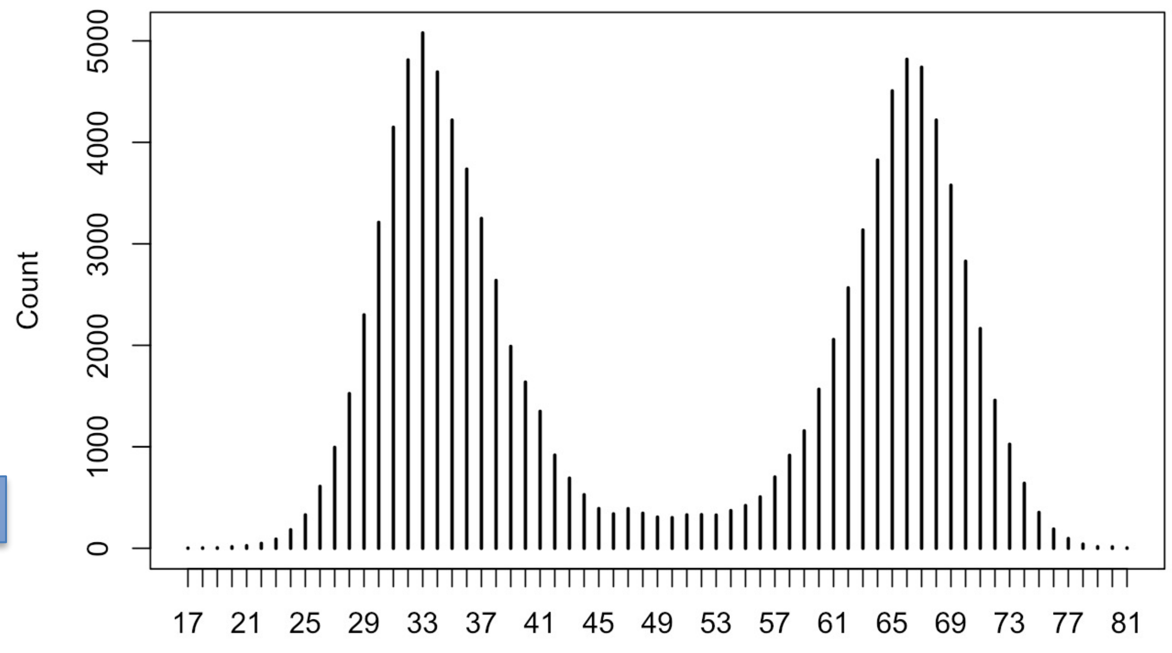
- Already discretized in 100 bins

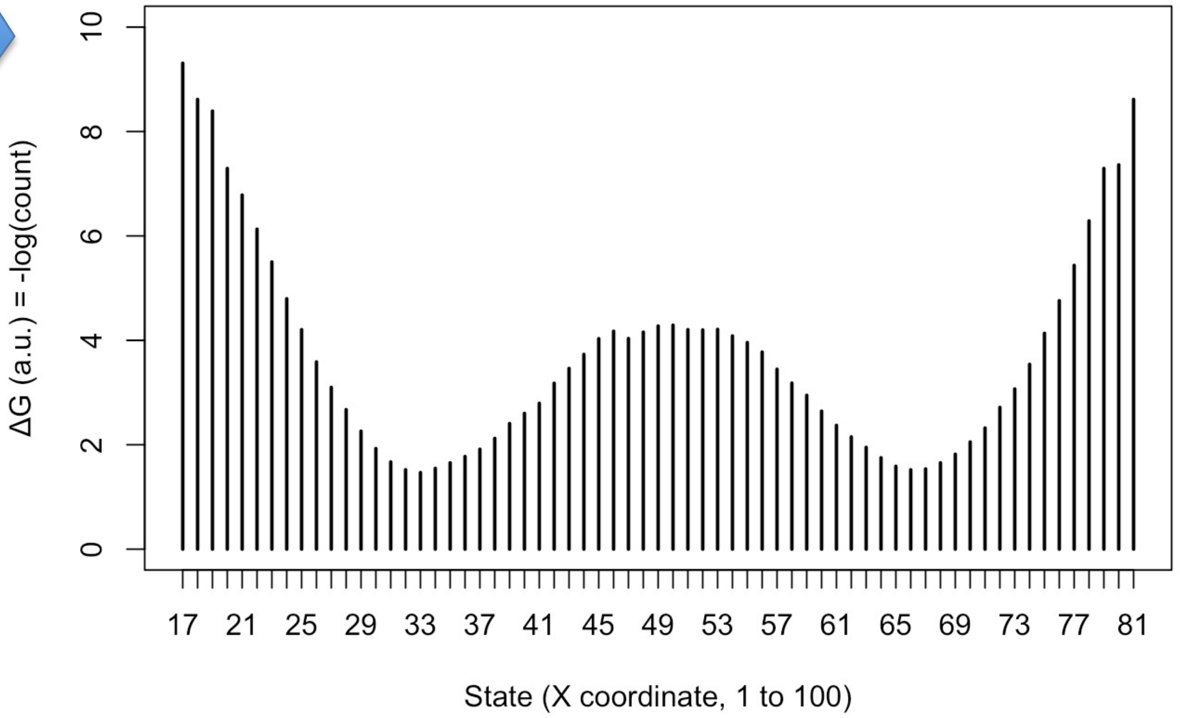


Histogram

Boltzmann inversion

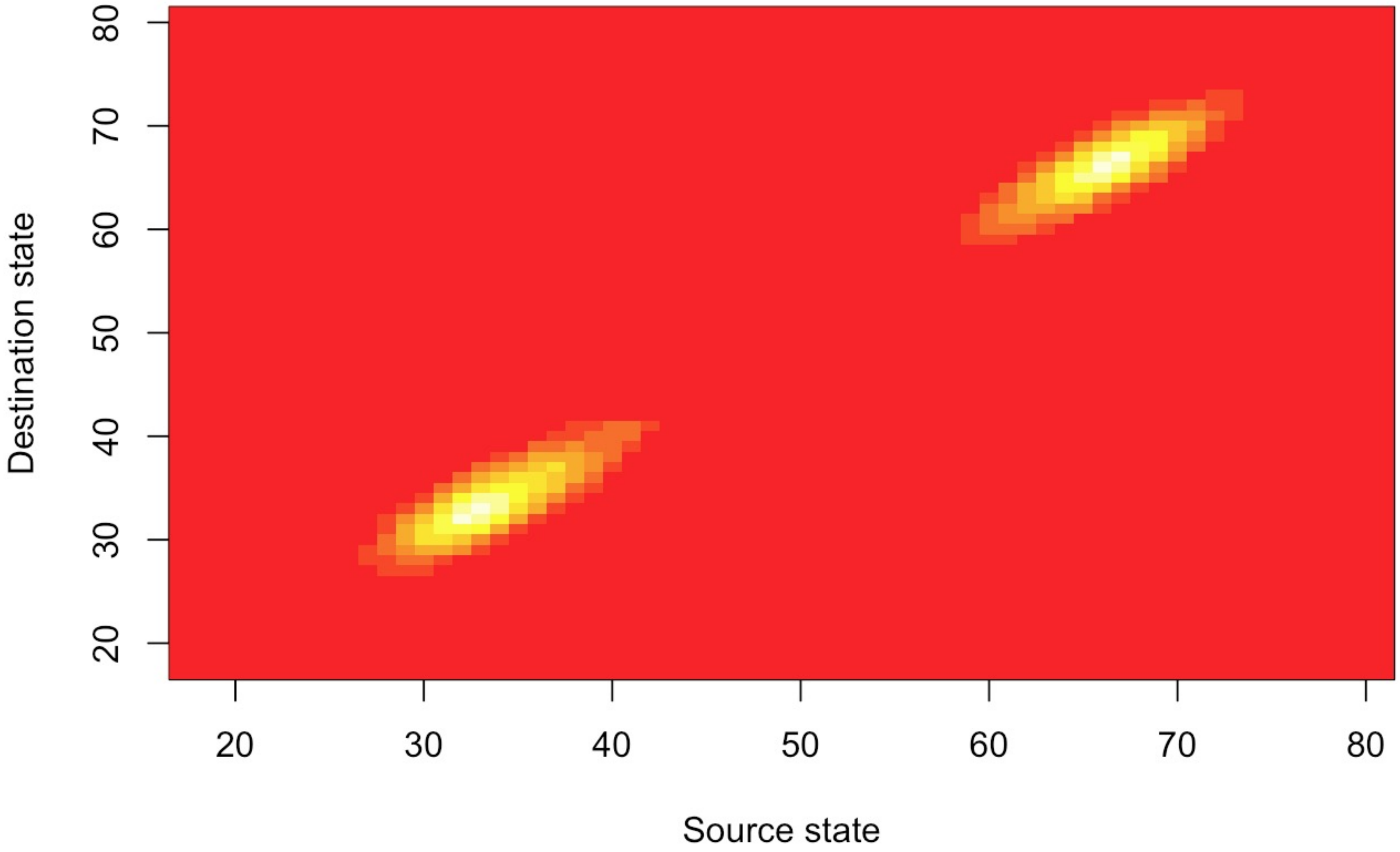
-log(count)





The transition *count* matrix*

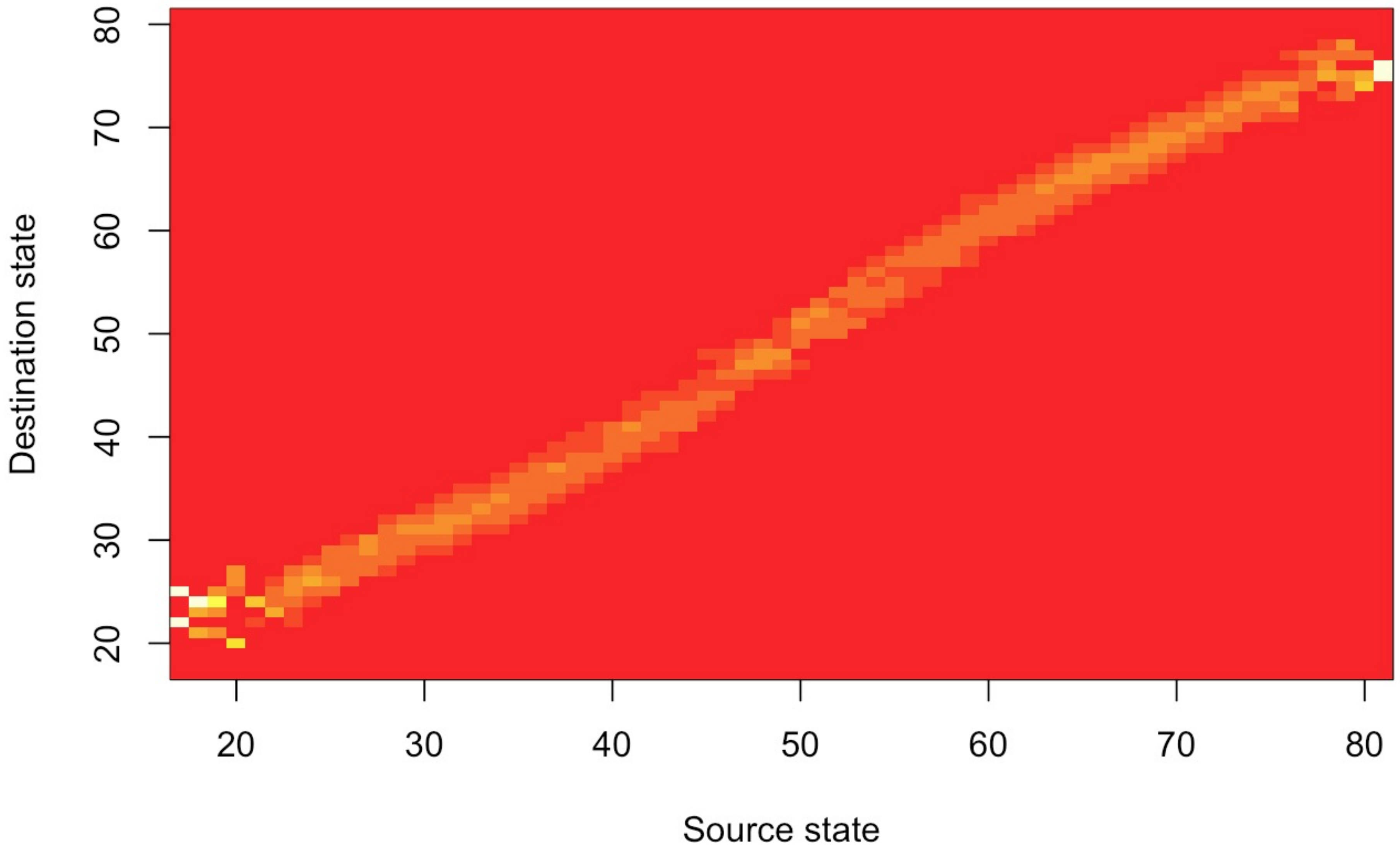
How many times we have seen state i going to j after τ=10 time units



* shown as an image for compactness

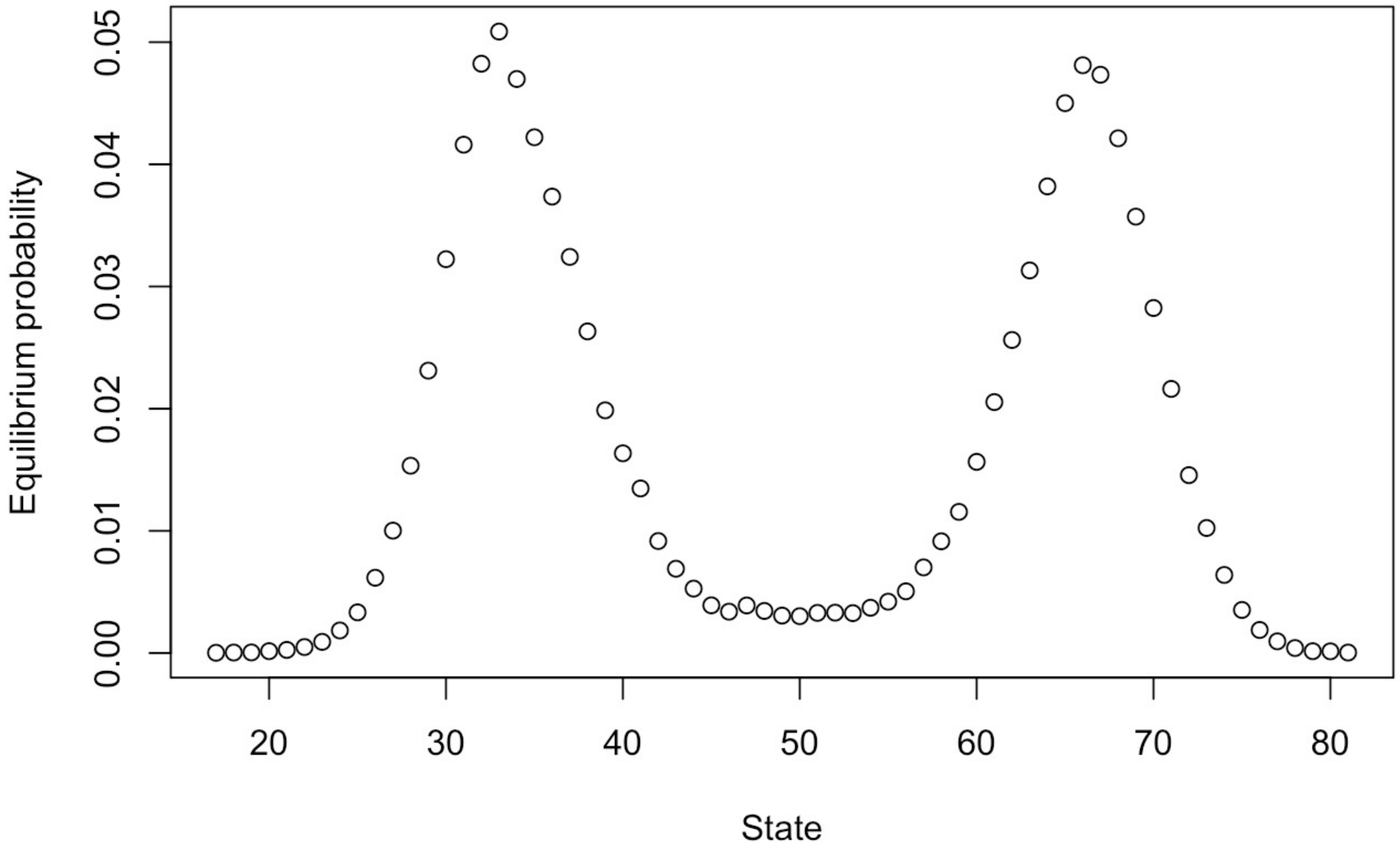
The transition *probability* matrix

Rows normalized to sum to 1



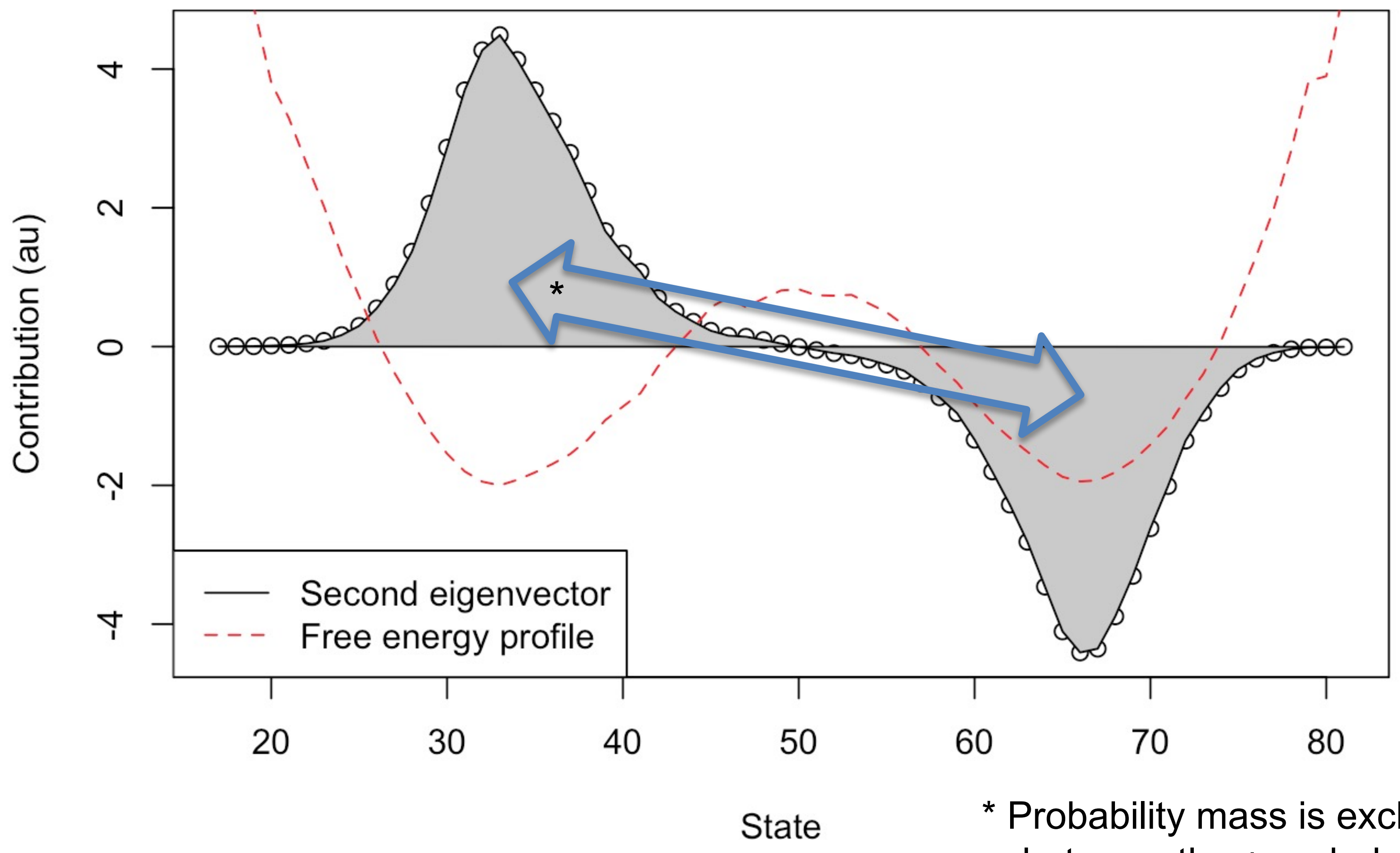
First eigenvector ($\mu_1=1$)

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



Second eigenvector (μ_2=0.997)

This is the slowest relaxation mode: ITS τ_2 = 3610 time units



* Probability mass is exchanged between the + and - basins

Take-home message

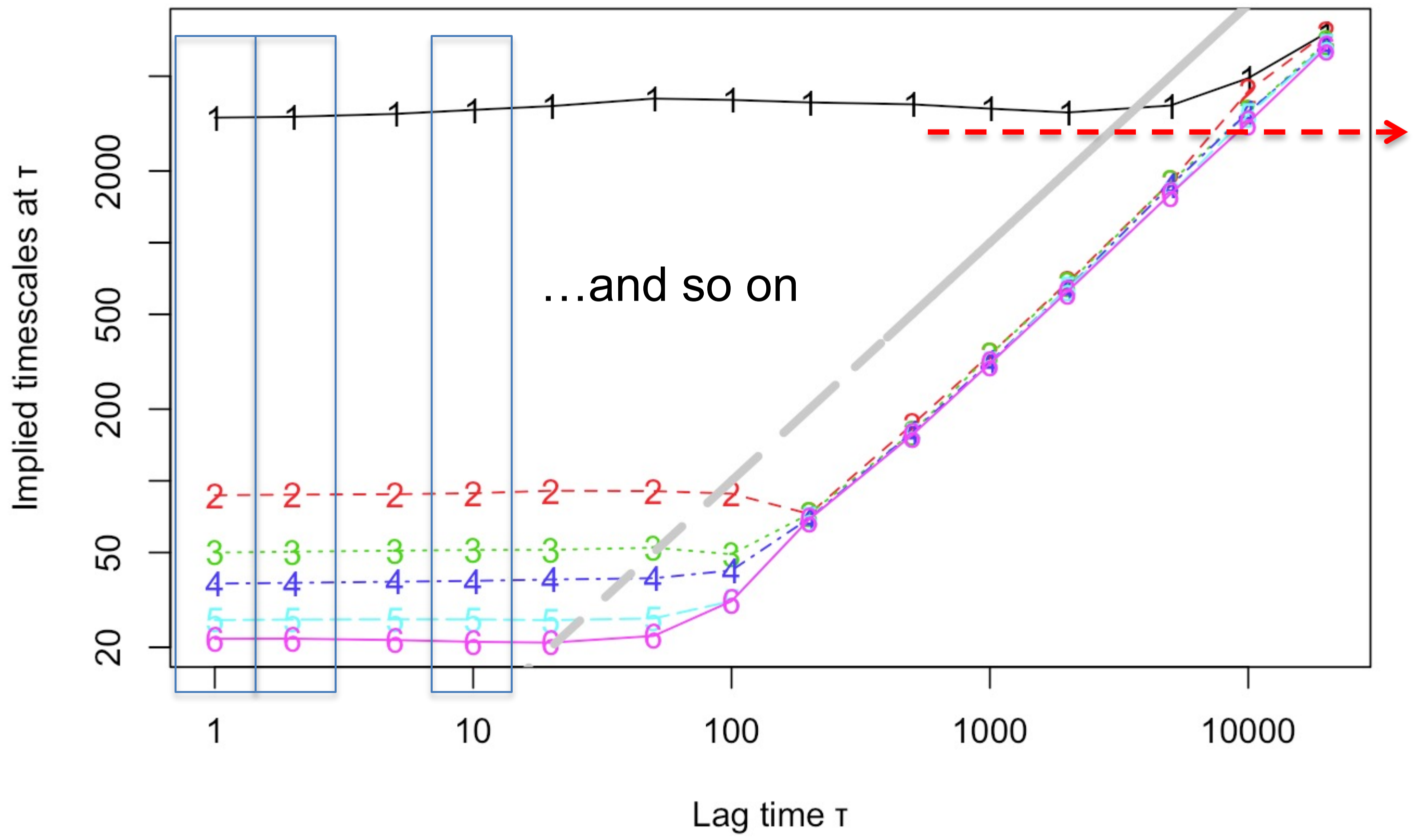
- Define states.
- Use trajectories to count transitions ➔ P_{ij}
- Eigenvalues: 1.0, μ_1, μ_2, …
  - These are the time-scales (after –log)
- Eigenvectors: s_∞, s_1, s_2, …
  - These are the equilibrium configuration, i.e. ΔG; and faster "oscillations" (kinetics)
- All are a function of τ: convergence

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:

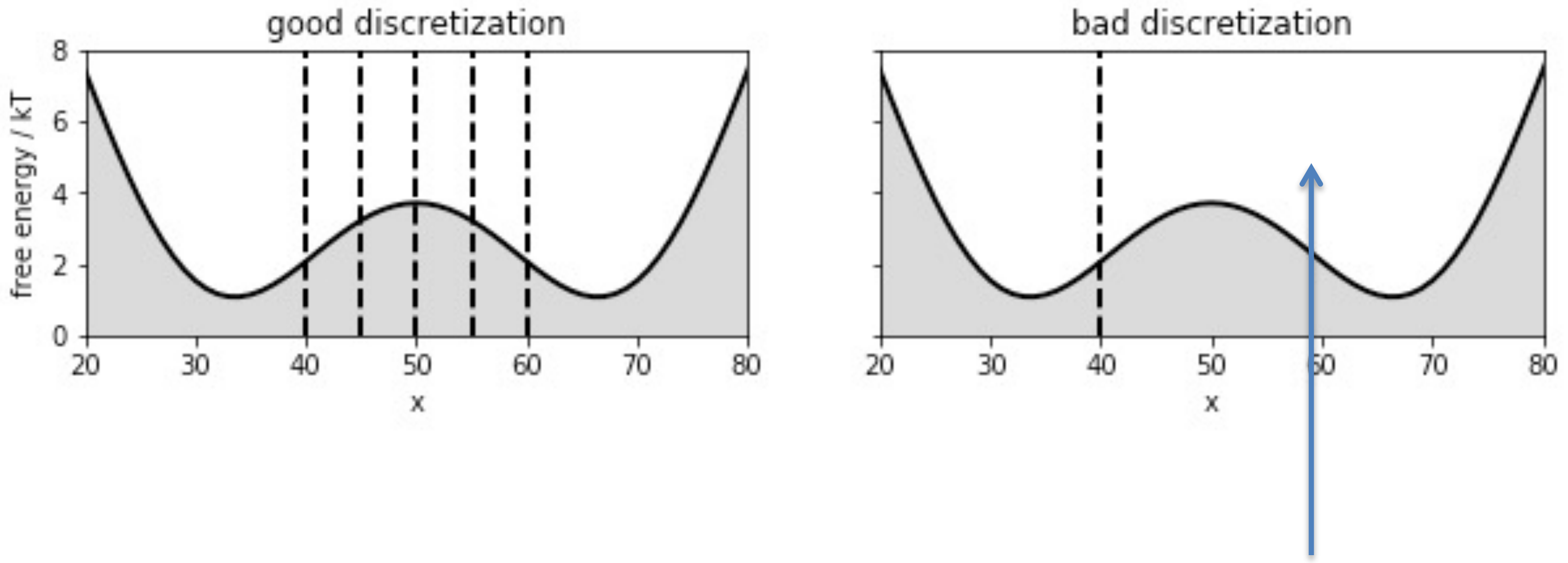
(a) true two-state dynamics;

(b) absence of orthogonal degrees of freedom;

(c) fine space discretization

Macrostates

A bad choice of the discretization breaks the Markovianity assumption

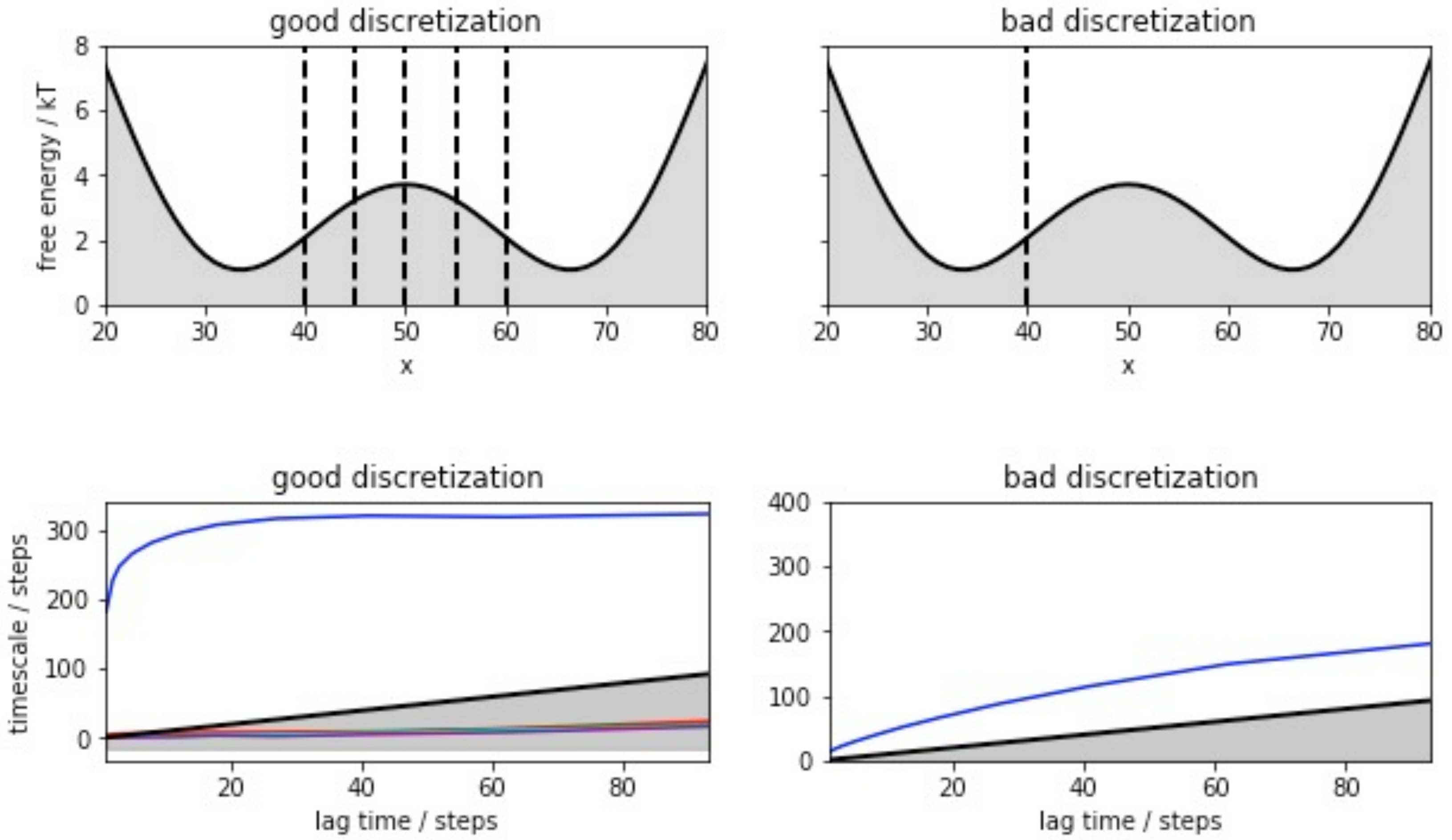


In the "bad discretization" case, the barrier is embedded in one of the states. This generates a "long term memory" effect: the rightmost state could actually be short-lived (if we are on the left of the barrier) or long-lived (if we are on its right). These two cases are convoluted into the same, so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

Based on http://www.emma-project.org/v2.0.4/generated/model_selection_validation.html

Macrostates

A bad choice of the discretization breaks the Markovianity assumption



Based on http://www.emma-project.org/v2.0.4/generated/model_selection_validation.html

Now let's head to files
DeepTime_Markov.ipynb
R_Markov.ipynb

The Analysis for HTMD page contains tutorials using actual large-scale simulation data and the HTMD library (note that it takes 30'-1h to install, download data, and run the analysis).