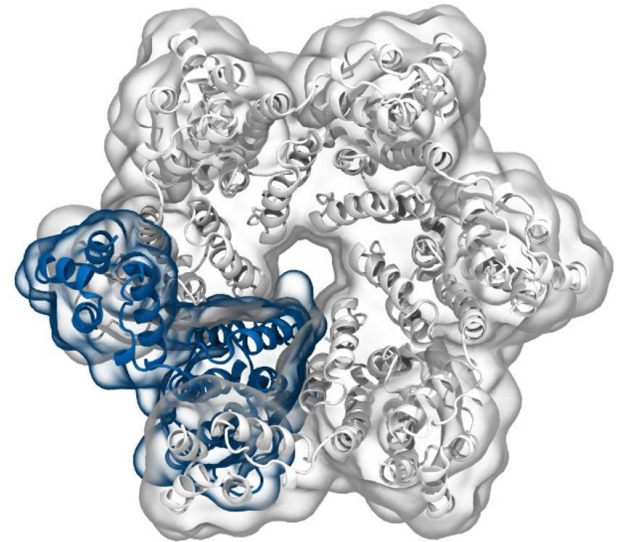
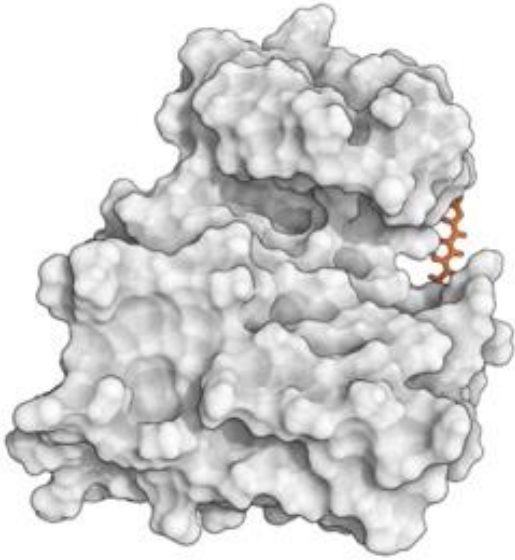


# Simulation of Biomolecules

## Markov State Modelling

### 2023 CCP5 Summer School



Dr Matteo Degiacomi  
Durham University

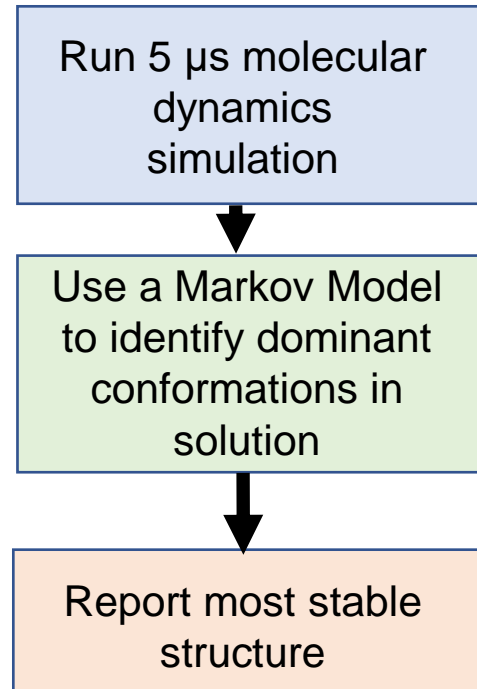
[matteo.t.degiacomini@durham.ac.uk](mailto:matteo.t.degiacomini@durham.ac.uk)

Dr Antonia Mey  
University of Edinburgh

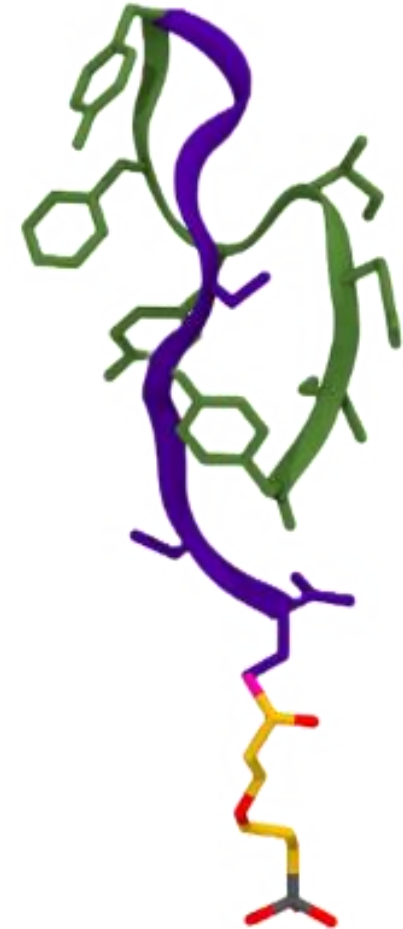
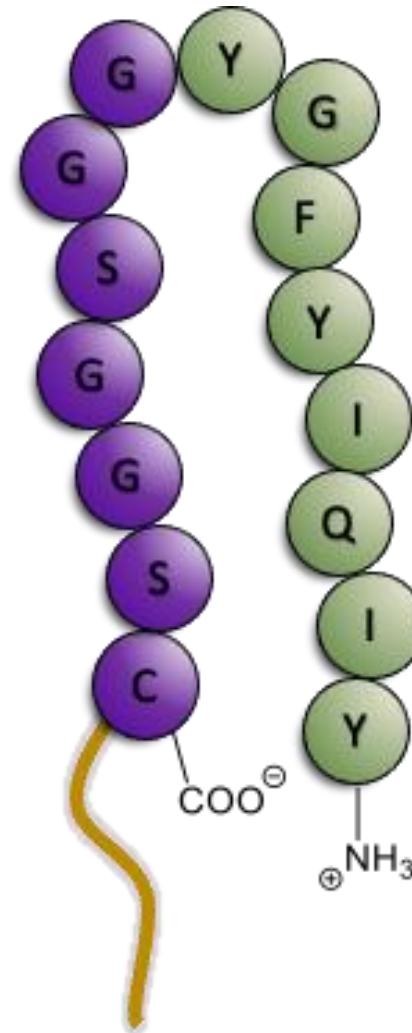
[antonia.mey@ed.ac.uk](mailto:antonia.mey@ed.ac.uk)

# Markov State Modelling (MSM)

Can we predict the equilibrium structure of a peptide?

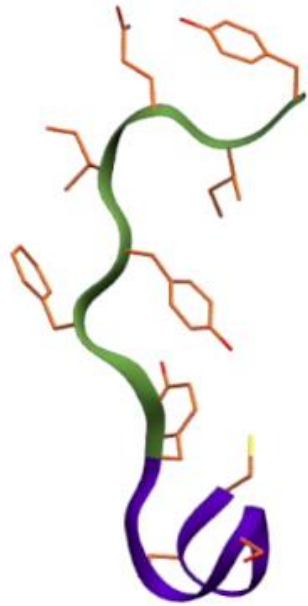


*AlphaFold suggestion*



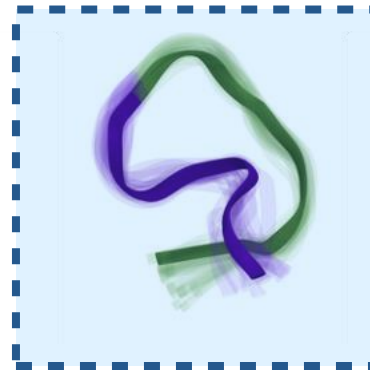
# MSMs allow for quantitative data from molecular simulations

YITIIY TAYAG SGGSC

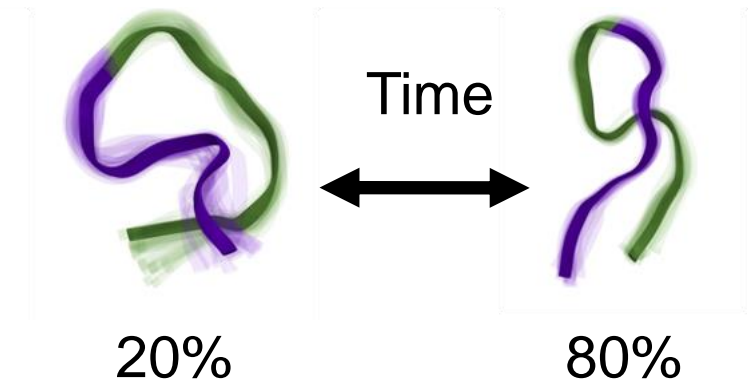


Protein dynamics

**Clustering** is the first step towards a model providing quantitative equilibrium and kinetic information from the simulation



Simulation movie



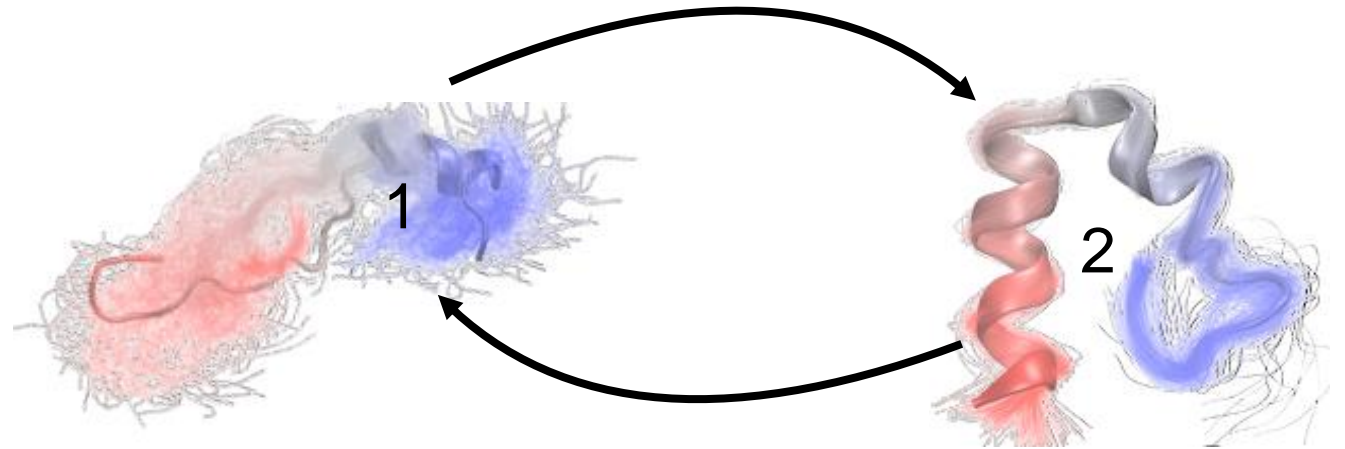
Probabilistic understanding of the trajectory

# Frogs jumping and proteins moving

Over the period of a day:

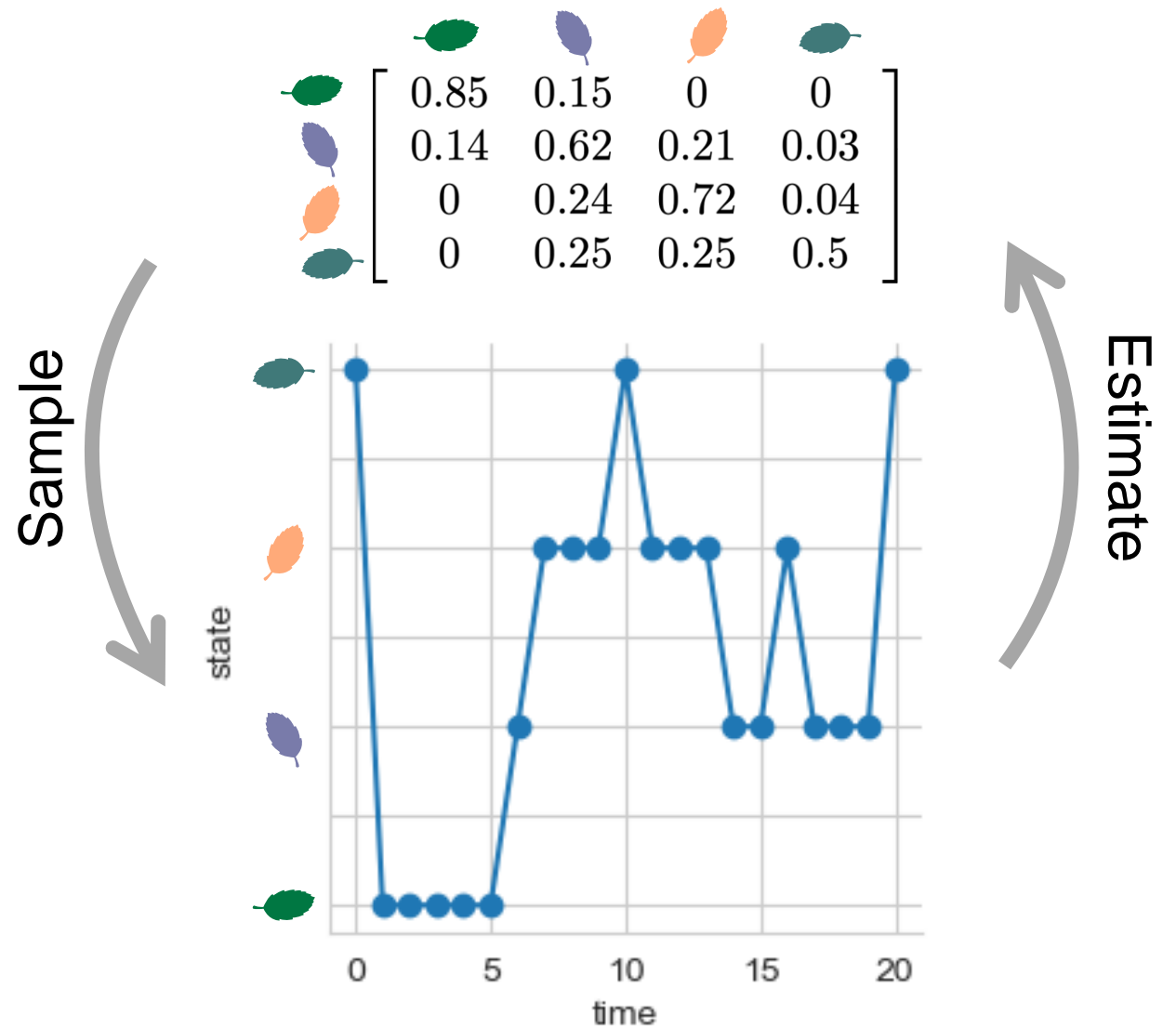
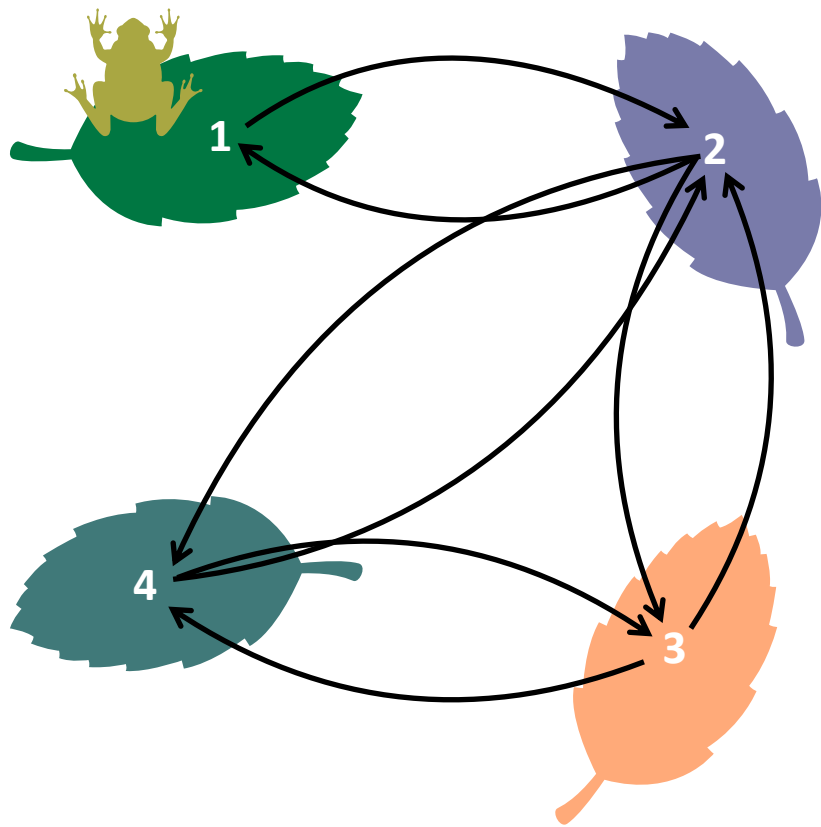


What do a protein and a frog jumping have in common?



# What is a Markov chain?

Over the period of a day:



# What is a Markov chain?

Transition matrices are:

- Stochastic matrices
- Containing conditional jump probabilities

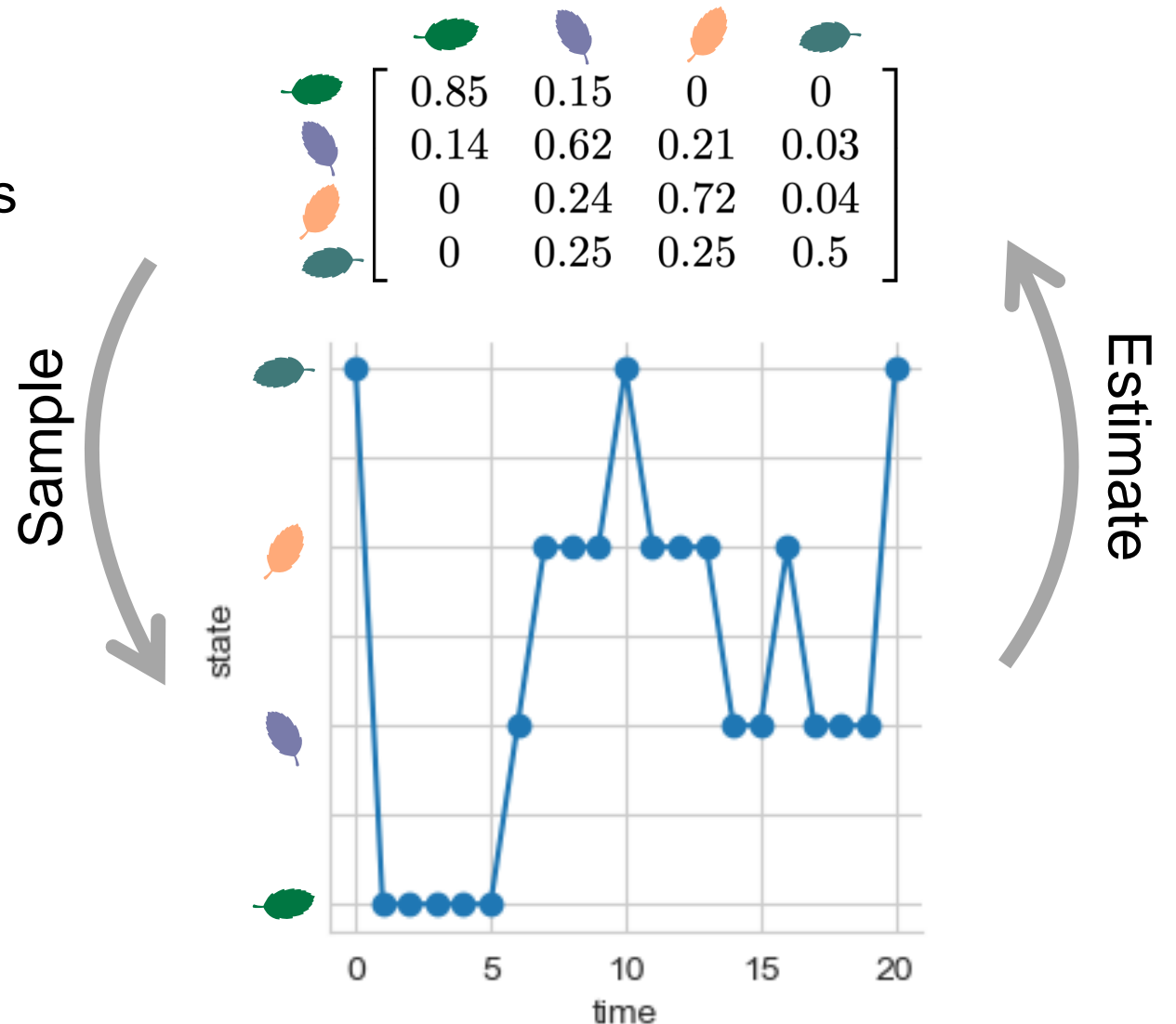
If they are **irreducible** and **ergodic**:

- They have a stationary probability
- Has a stationary probability  $\pi$ .
- The stationary probability is the eigenvector of the eigenvalue  $\lambda_1$

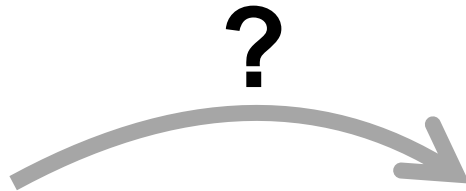
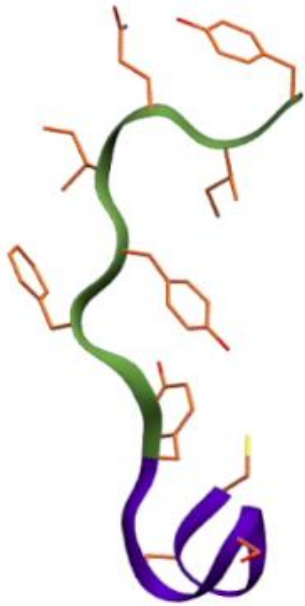
$$\lambda_1 = 1 > \lambda_2 > \lambda_3, \dots, > \lambda_n$$

If they are reversible

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



# How to model protein dynamics using a Markov approach?

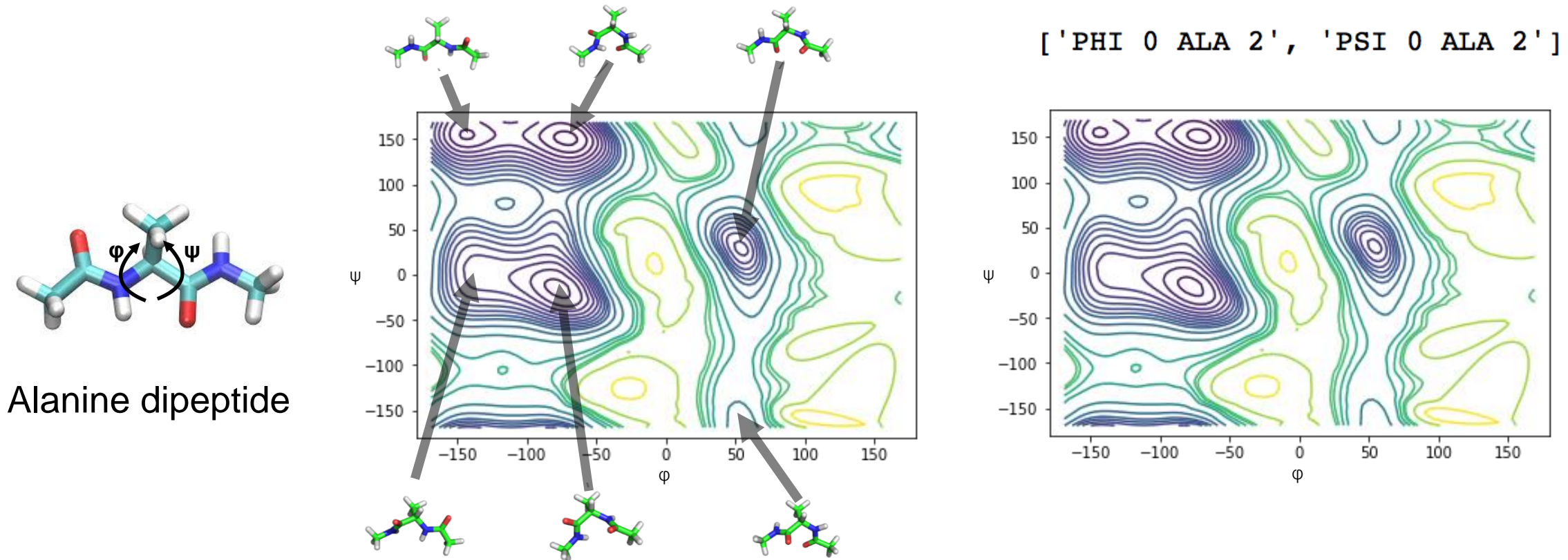


Approximate with a transition matrix from the trajectory:

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,j} & \dots & P_{1,S} \\ P_{2,1} & P_{2,2} & \dots & P_{2,j} & \dots & P_{2,S} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{i,1} & P_{i,2} & \dots & P_{i,j} & \dots & P_{i,S} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{S,1} & P_{S,2} & \dots & P_{S,j} & \dots & P_{S,S} \end{bmatrix}.$$

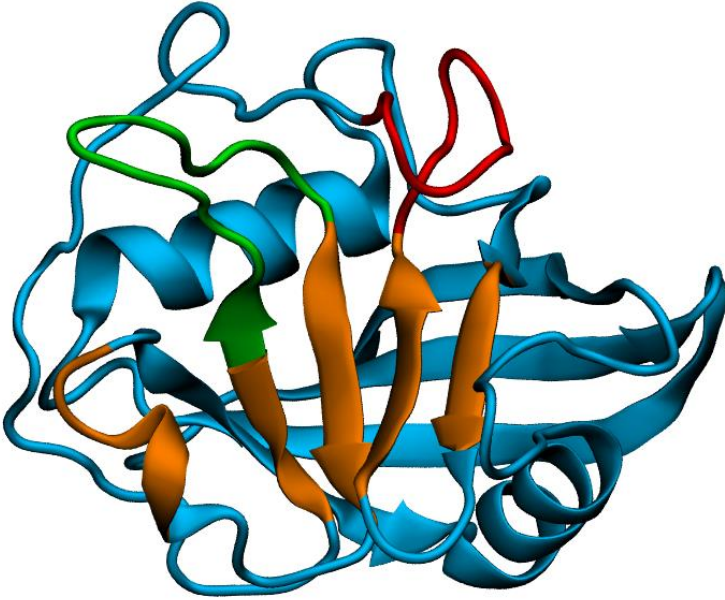


# Dimensionality reduction is (often) needed for MSM





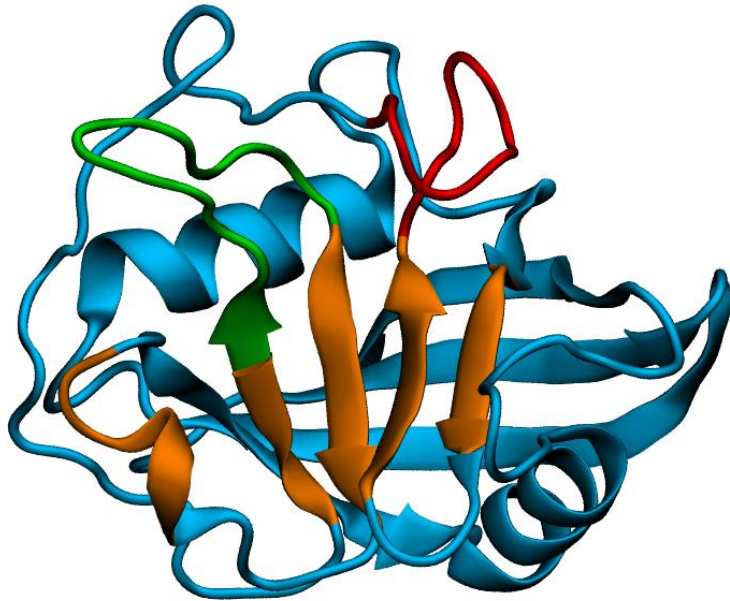
# Dimensionality reduction is (often) needed for MSM



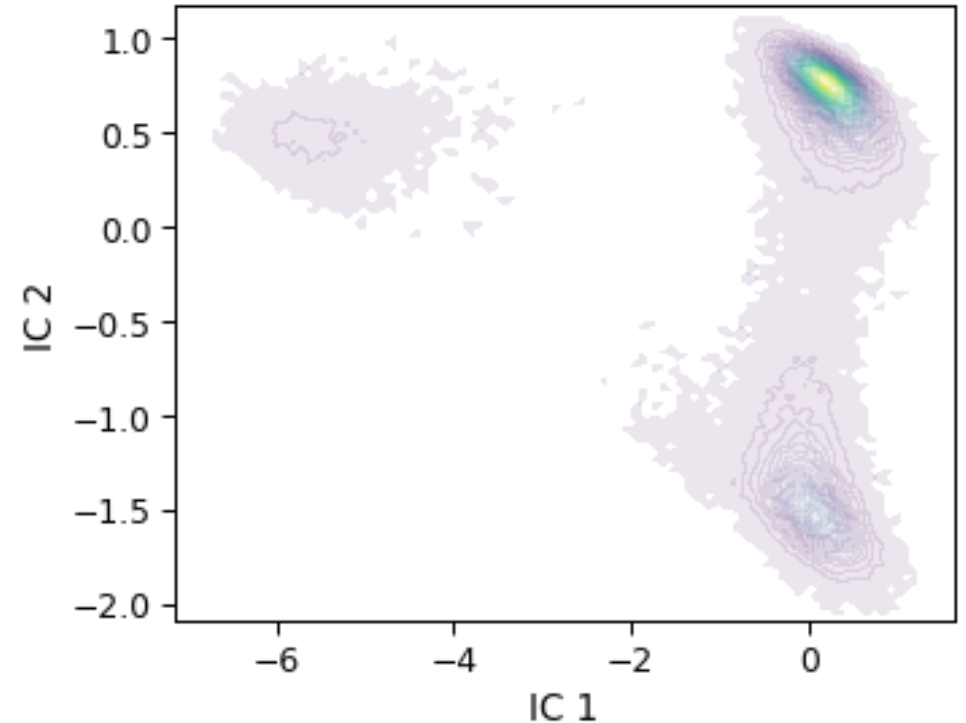
```
[ 'ATOM:ACE 1 CH3 1 x',  
  'ATOM:ACE 1 CH3 1 y',  
  'ATOM:ACE 1 CH3 1 z',  
  'ATOM:ACE 1 C 4 x',  
  'ATOM:ACE 1 C 4 y',  
  'ATOM:ACE 1 C 4 z',  
  'ATOM:ACE 1 O 5 x',  
  'ATOM:ACE 1 O 5 y',  
  'ATOM:ACE 1 O 5 z',  
  'ATOM:ALA 2 N 6 x',  
  'ATOM:ALA 2 N 6 y',  
  'ATOM:ALA 2 N 6 z',  
  'ATOM:ALA 2 CA 8 x',  
  'ATOM:ALA 2 CA 8 y',  
  'ATOM:ALA 2 CA 8 z',  
  'ATOM:ALA 2 CB 10 x',  
  'ATOM:ALA 2 CB 10 y',  
  'ATOM:ALA 2 CB 10 z',  
  'ATOM:ALA 2 C 14 x',
```

```
[ 'PHI 0 ALA 2', 'PSI 0 ALA 2']
```

# Dimensionality reduction — TICA, PCA, VAMP



Project features onto  
low dimensional  
subspace

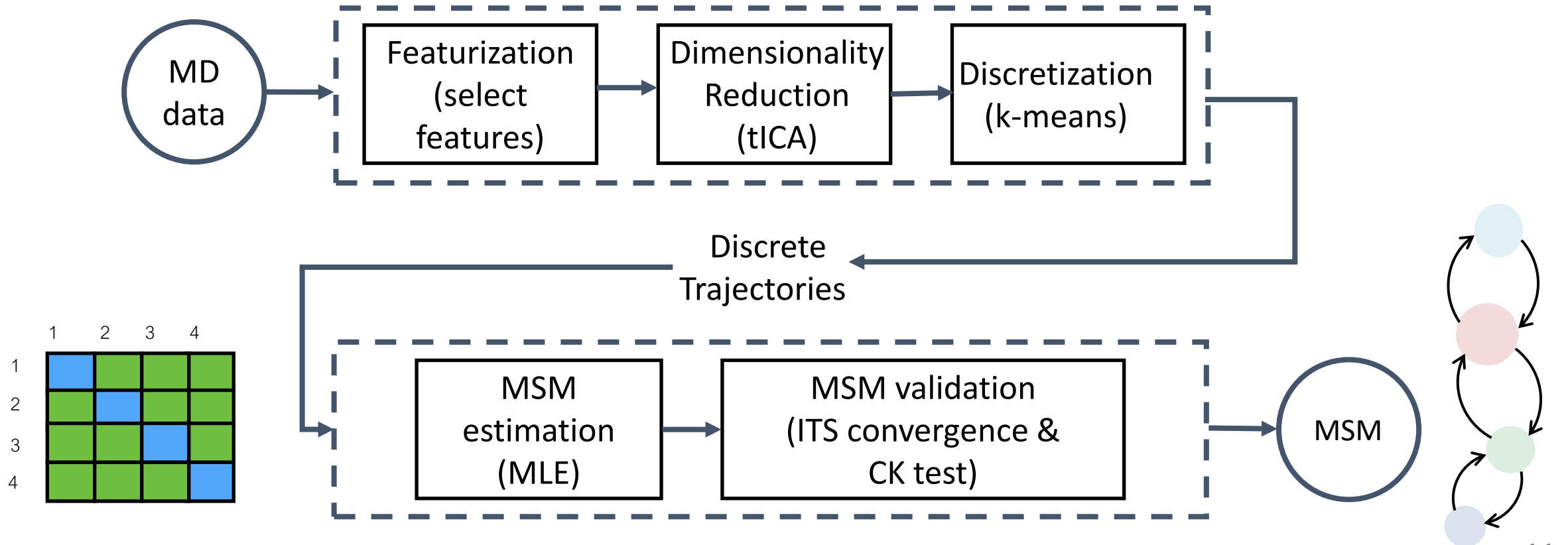
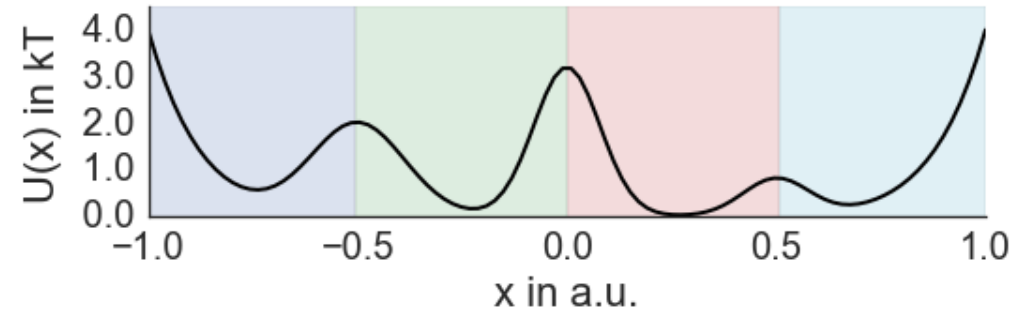
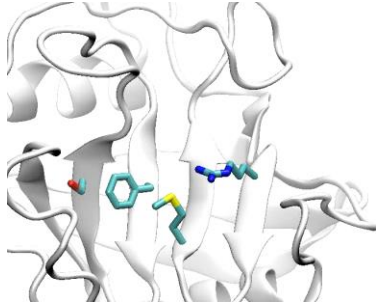


**PCA:** Linear combination of input features maximising the variance

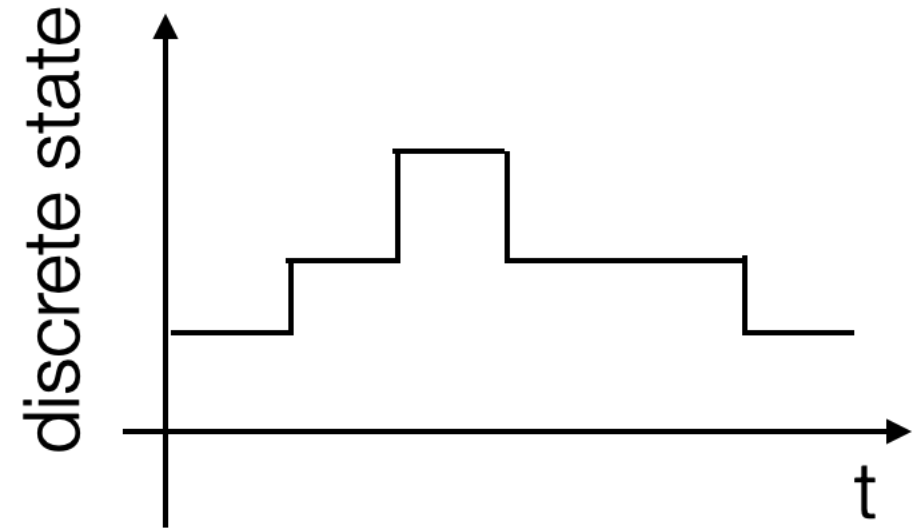
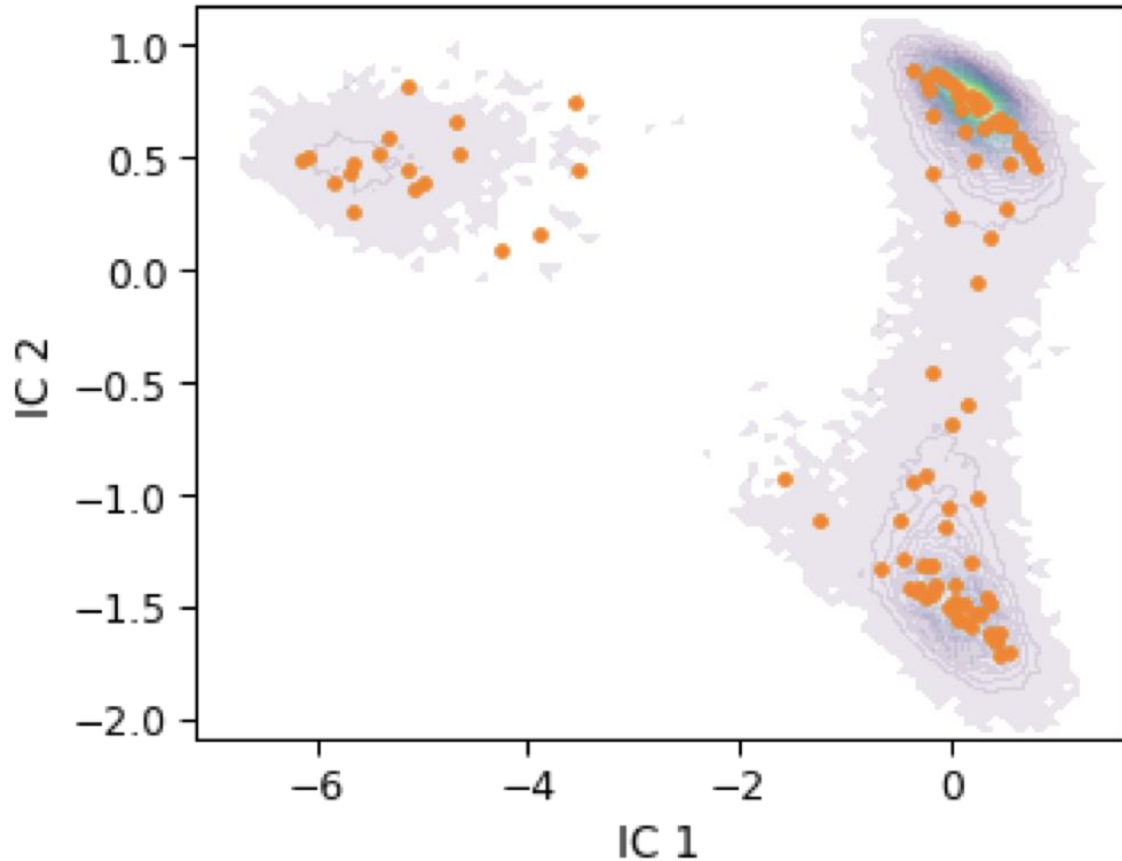
**TICA:** Linear combination of input features maximising time autocorrelation

**VAMP:** Variational approach for Markov Process, true for non-equilibrium data

# Building a Markov Model requires several steps



# Clustering on reduced dimensional space



k-means

regular

etc.

spatial

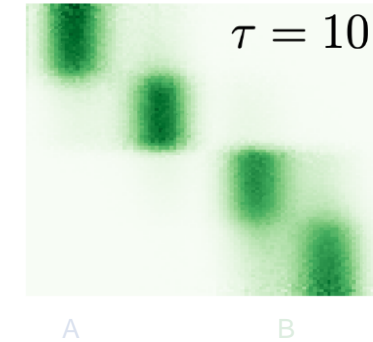
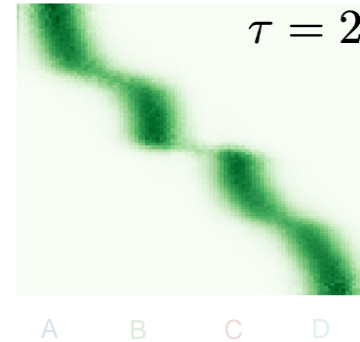


Dimensionality reduction and discretisation require a lot of parameter optimisation. It is necessary to spend a good amount of time on hyper parameter optimisation.

# The transition matrix contains all quantative properties

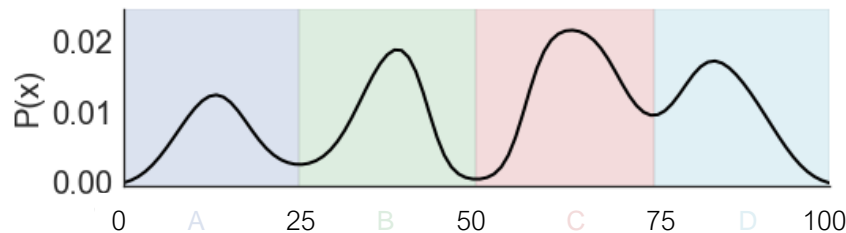
Approximate with a transition matrix from the trajectory:

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \dots & P_{1,j} & \dots & P_{1,S} \\ P_{2,1} & P_{2,2} & \dots & P_{2,j} & \dots & P_{2,S} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{i,1} & P_{i,2} & \dots & P_{i,j} & \dots & P_{i,S} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ P_{S,1} & P_{S,2} & \dots & P_{S,j} & \dots & P_{S,S} \end{bmatrix}.$$



varying the lag time

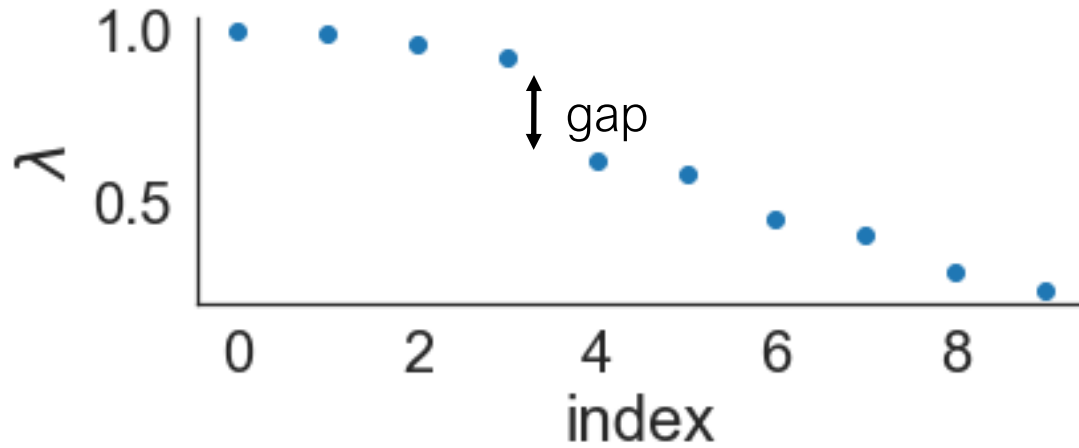
$$\lambda_1 = 1 > \lambda_2 > \lambda_3, \dots, > \lambda_n$$



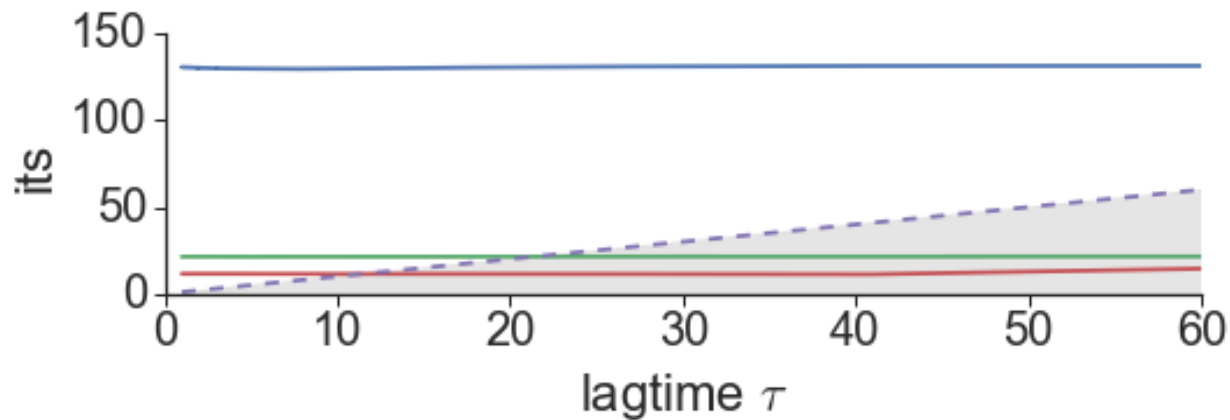
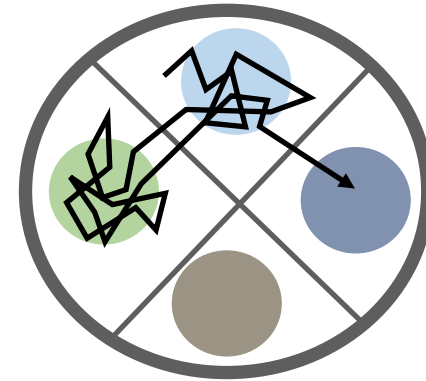
$$\begin{matrix} \text{P} & \text{v} & = & \lambda & \text{v} & \text{right eigenvector} \\ \text{P}^T & \text{v} & = & \lambda & \text{v} & \text{left eigenvector} \end{matrix}$$

We can find the most likely structure/coordinates!

# Eigenvalues tell us about timescales

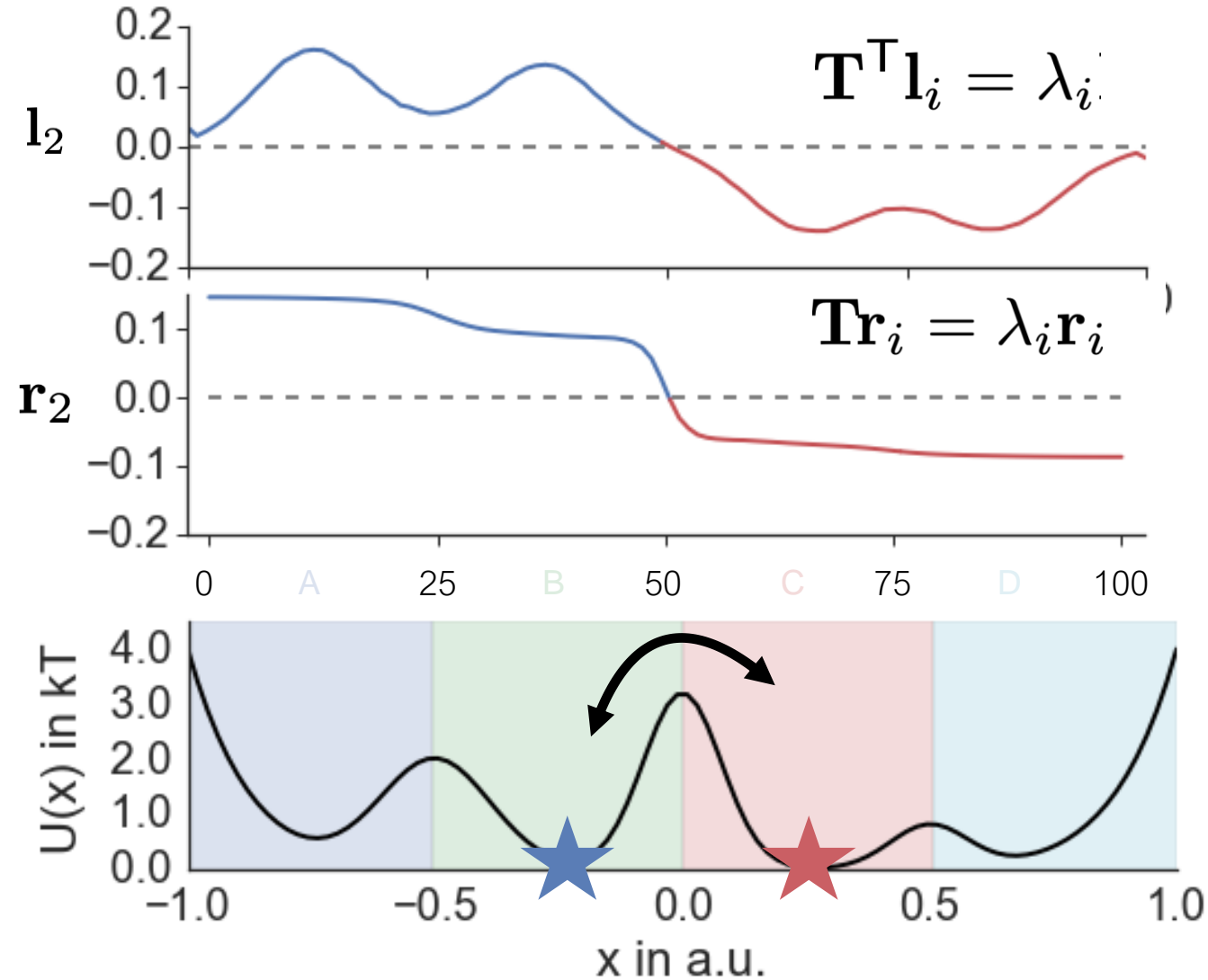


$$\lambda_1 = 1 > \lambda_2 > \lambda_3, \dots, > \lambda_n$$



$$t_i = -\frac{\tau}{\ln|\lambda_i|}$$

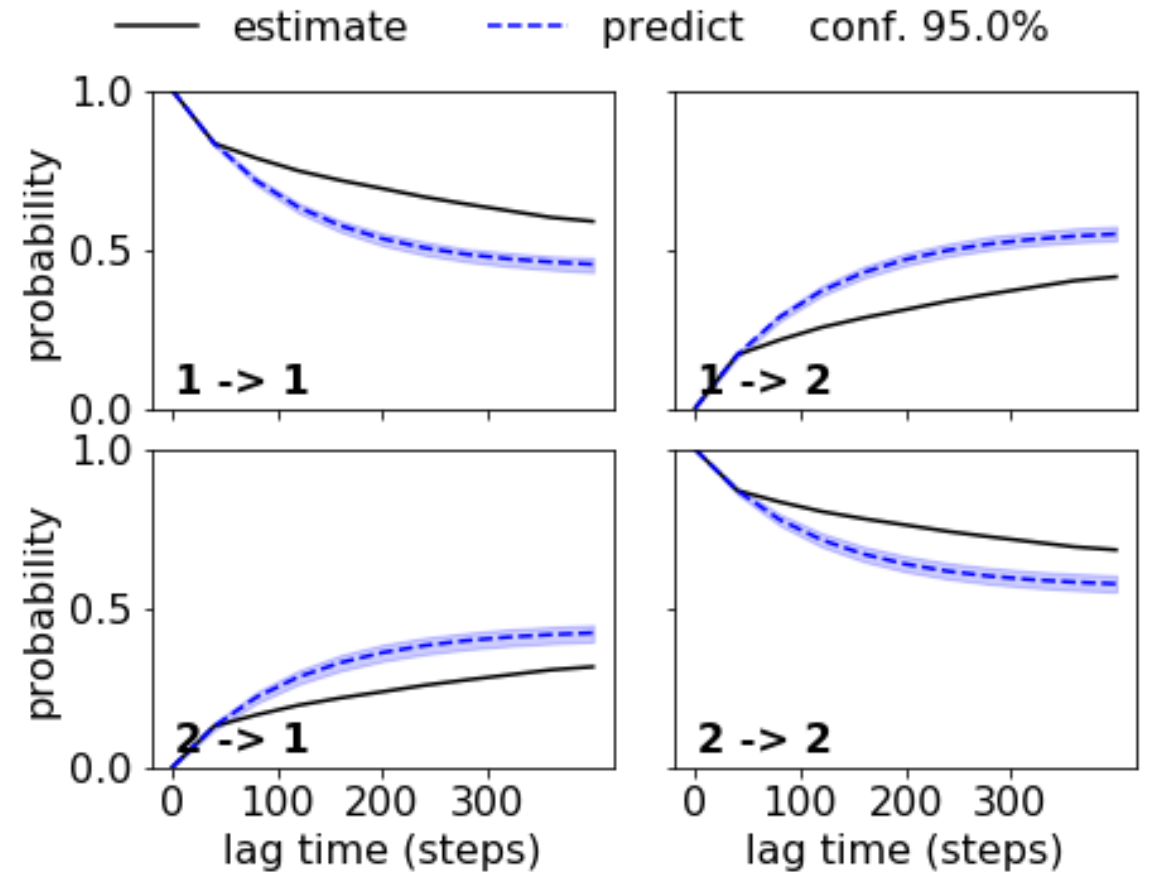
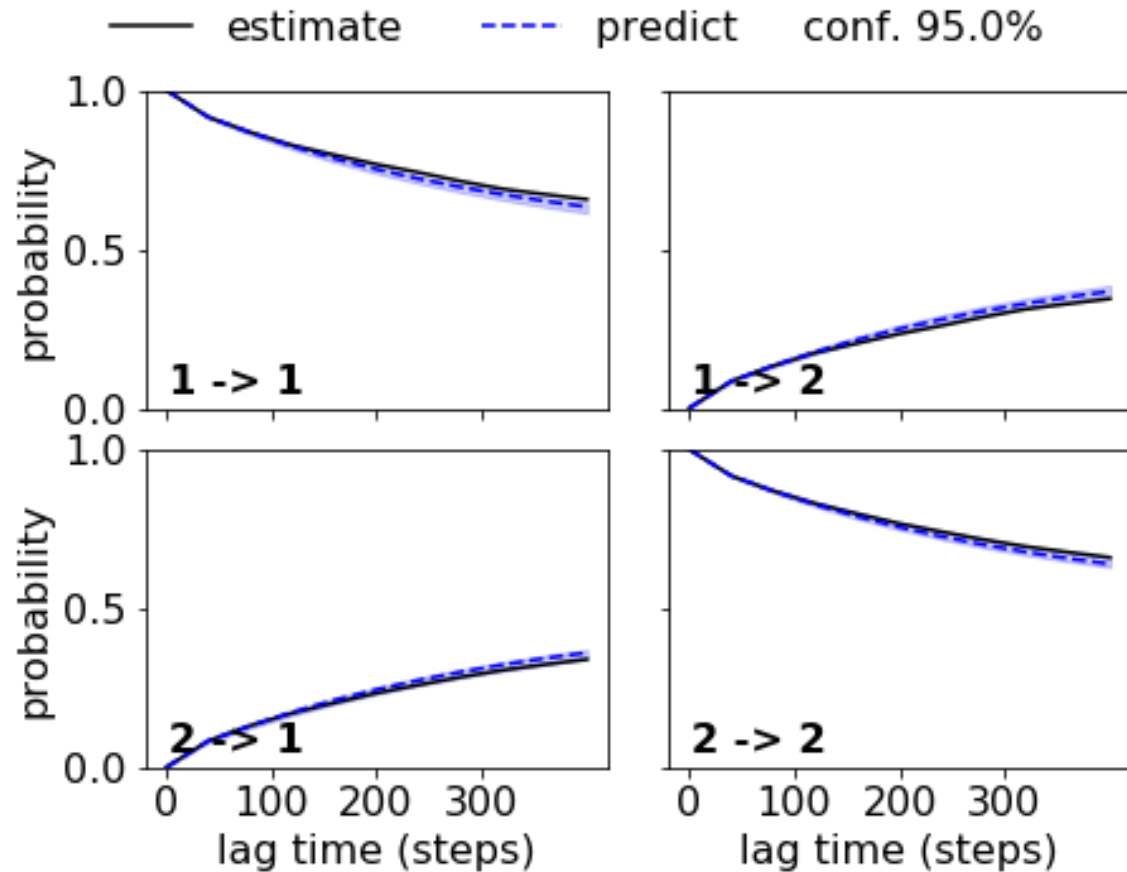
# The eigenvectors tell us about the slowest processes





# Validation with the Chapman-Kolmogorov (CK) test

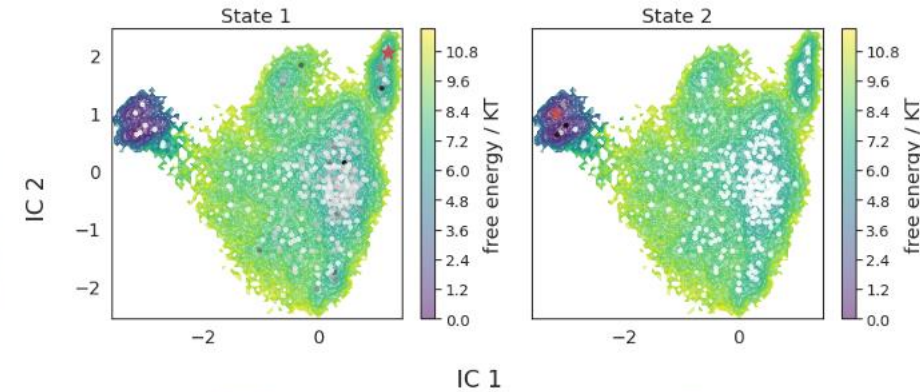
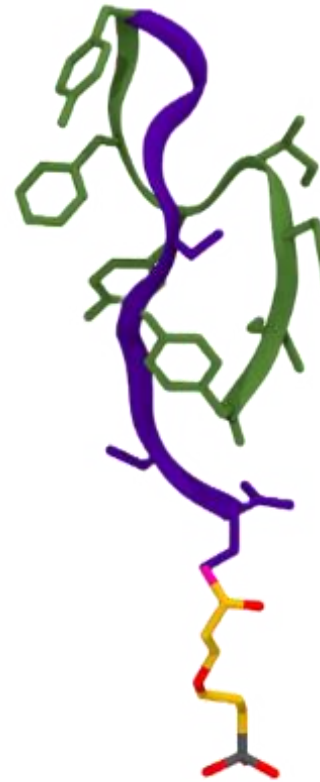
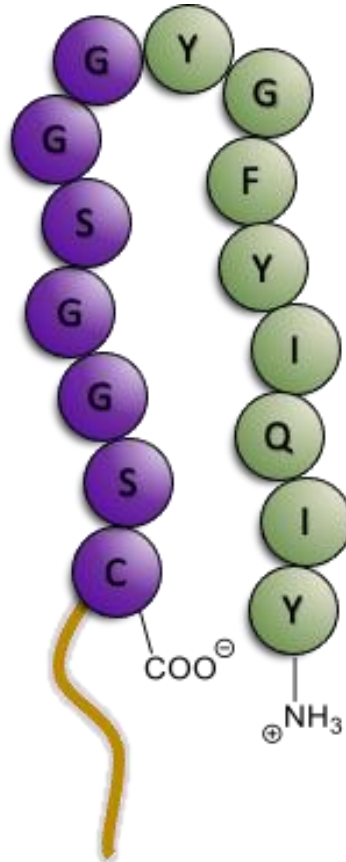
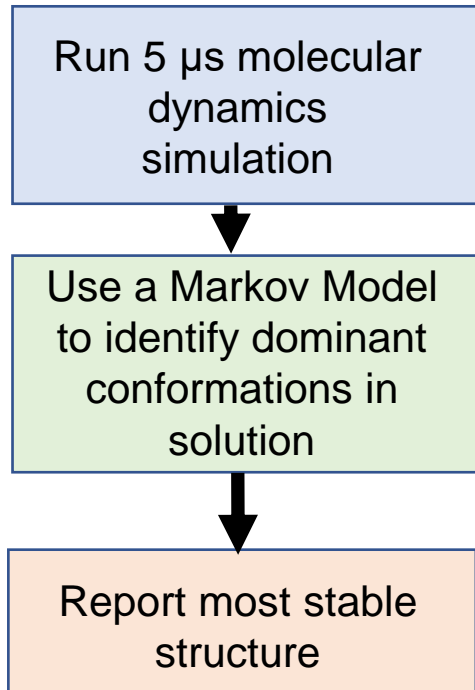
$$T(k\tau) = T(\tau)^k$$



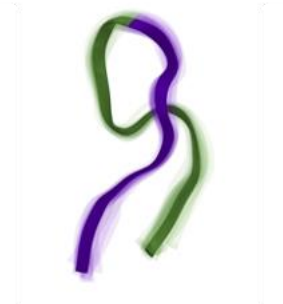
# Conformational dynamics of a peptide

Can we predict the equilibrium structure of a peptide?

*AlphaFold  
suggestion*



20%



80%

# Python tools are available for building MSMs

<https://www.livecomsjournal.org>

A LiveCoMS Tutorial

## Introduction to Markov state modeling with the PyEMMA software [Article v1.0]

Christoph Wehmeyer<sup>1†\*</sup>, Martin K. Scherer<sup>1†</sup>, Tim Hempel<sup>1†</sup>, Brooke E. Husic<sup>1,2</sup>, Simon Olsson<sup>1</sup>, Frank Noe<sup>1,3\*</sup>



[https://github.com/markovmodel/pyemma\\_tutorials](https://github.com/markovmodel/pyemma_tutorials)



markovmodel / pyemma\_tutorials

<> Code

Issues 1

Pull requests 0

Projects 0

Wiki

Insights

How to analyze molecular dynamics data with PyEMMA

<https://github.com/deeptime-ml/deeptime>

deeptime-ml / deeptime Public

<> Code

Issues 9

Pull requests

Discussions

Actions

Projects

Security

Insights

