# Optimal Construction and Coarse-Graining of Markov State Models

**Adam Kells**

A thesis presented for the degree of
Doctor of Philosophy

October 31, 2020

Department of Chemistry
Kings College London
United Kingdom

To my parents

# Abstract

Markov state models (MSMs) are a widely used approach for creating interpretable memoryless kinetic models from time series data. In this PhD thesis, we develop upon the existing MSM theory and investigate how MSMs can be used to automate and remove ambiguity in the analysis of kinetic systems. Firstly we consider using coarse-graining to identify metastable and transition states from high-dimensional models. We propose an eigenvalue-based variational optimization method for obtaining these states based on a lagtime independent projection approach. We demonstrate that this variational protocol is linked to the theory of mean first passage times to provide an intuitive interpretation. We show that our kinetic clustering can be extended to more general geometric networks, provided they are well described by a stochastic block model. We illustrate the broad applicability of this method by deriving a parallel tempering approach for accelerating the coarse-graining on models with large numbers of nodes. Secondly, we derive an equation for estimating the relaxation time of a Markovian system in the long lagtime limit by making some simple assumptions about the functional dependence between the true high-dimensional Markovian dynamics and the discrete approximation. We show that this equation can allow for accurate relaxation times to be extraced from limited simulation data. Finally we demonstrate how MSMs can be used in conjunction with umbrella sampling simulations to quickly, conveniently and accurately calculate membrane permeabilities.

# Acknowledgements

In completing this PhD I am immeasurably in debt to the many people who have supported me throughout.

First and foremost I thank Drs. Edina Rosta and Alessia Annibale for their patient supervision and encouragement. Throughout this PhD, start to finish, they have been always willing to make time for me and to offer research and career guidance.

I am grateful to everyone who made this four year journey so much more enjoyable through their friendship. I thank my entire cohort at CANES for helping me survive that first year with special mention to Alistair and Evan for having shouldered the additional burden of living with me. I also thank my fellow PhDs in the chemistry department Magd, Dénes, Fahim and Pedro whose company made it that bit easier to drag myself in to Britannia House in the morning.

I can't finish without thanking my girlfriend Jennifer for the immense influence she has had on my finishing the PhD. Her indiminishable ability to pick me up when I feel most ready to quit has carried me through some of the most difficult times of the past three years and for that I am forever thankful.

Finally, more than anyone else I thank my family, my parents Ronnie and Eileen and my sister Alanna. My parents have supported me through every decision I've made and I am grateful for the sacrifices they made to give me the opportunities they didn't have.

# Contents

# List of Figures

# List of Publications

1. Martini, L.*, Kells, A.*, Covino, R., Hummer, G., Buchete, N.V. and Rosta, E., 2017. Variational identification of Markovian transition states. Physical Review X, 7(3), p.031060.

2. Leahy, C.T., Kells, A., Hummer, G., Buchete, N.V. and Rosta, E., 2017. Peptide dimerization-dissociation rates from replica exchange molecular dynamics. The Journal of chemical physics, 147(15), p.152725.

3. Stelzl, L.S., Kells, A., Rosta, E. and Hummer, G., 2017. Dynamic histogram analysis to determine free energies and rates from biased simulations. Journal of chemical theory and computation, 13(12), pp.6328-6342.

4. Kells, A., Annibale, A. and Rosta, E., 2018. Limiting relaxation times from Markov state models. The Journal of chemical physics, 149(7), p.072324.

5. Badaoui, M.*, Kells, A.*, Molteni, C., Dickson, C.J., Hornak, V. and Rosta, E., 2018. Calculating Kinetic Rates and Membrane Permeability from Biased Simulations. The Journal of Physical Chemistry B, 122(49), pp.11571-11578.

6. Kells, A., Mihálka, Z.É., Annibale, A. and Rosta, E., 2019. Mean first passage times in variational coarse graining using Markov state models. The Journal of chemical physics, 150(13), p.134107.

7. Pramanik, D., Smith, Z., Kells, A. and Tiwary, P., 2019. Can One Trust Kinetic and Thermodynamic Observables from Biased Metadynamics Simulations?: Detailed Quantitative Benchmarks on Millimolar Drug Fragment Dissociation. The Journal of Physical Chemistry B, 123(17), pp.3672-3678.

The following is a list of peer-reviewed publications published during the timeline of the PhD[1]. Papers 1, 7, 4 and 5 comprise chapters 2, 3, 5 and 6 respectively. Chapter 4 features unpublished work currently in preparation. The contribution to the remaining papers is not presented in this thesis.

[1] * is used to denote equal contribution.

*"Beginnings are always troublesome"*

George Eliot

*"This model will be a simplification and an idealization, and consequently a falsification. It is to be hoped that the features retained for discussion are those of greatest importance in the present state of knowledge."*

Alan Turing

# 1 Introduction

## 1.1 A Brief History of Markov models

### 1.1.1 From Markov to Zwanzig

In 1913, Andrei Markov published a paper concerned with the statistical analysis of letters appearing in the text of a Russian novel [2, 3]. He was interested in vowels and consonants, in particular how likely one was to follow the other. By analysing this text, he was able to establish a central limit theorem (CLT) for non-independent events. Prior to this the CLT had been used only to show that certain sums of independent

random variables would lead to normal distributions.

By extending the CLT to events which were conditioned on each other, Markov instigated a fundamental shift in probability theory, away from viewing systems as experiments of independent trials and towards a series of linked events where the outcome of the next trial depends upon the current.

While Markov pursued this line of research purely for a generalization of the law of large numbers, in the years that followed several researchers saw the usefulness of the memoryless property in describing kinetic systems. Note for example that Brownian motion [4], one of the most important fields of study at the beginning of the 20th century is a Gaussian Markov process as the position at the next time step $X_{t+1}$ depends on only the current position $X_t$ (equation 1.1 where $\sigma$ is the variance parameter and $B_t$ is a number drawn from a Gaussian distribution).

$$X_{t+1} = X_t + \sigma B_{t+1} \tag{1.1}$$

This change of thinking to memoryless dynamics has proven to be ubiquitously useful. In the Markov formalism, the past and future are independent and only the present state of the system is needed to make quantitative predictions of the future. As such, over the years this approach has been used to model systems with both discrete and continuous state spaces, such as the stock market [5–8], animal migration patterns [9,10] and famously internet search results with the Google PageRank algorithm [11,12].

Discrete state and time Markov systems (usually termed Markov chains) are particularly popular due to their human interpretability. These systems consist of some discrete sets of states which the system can occupy with an associated matrix of transition probabilities to move between the states as shown for example in figure 1.1 (with $T$ denoting the transition probability).

What was less clear following Markovs original work was whether one could take a system with continuous Markovian dynamics and describe it accurately by a system of

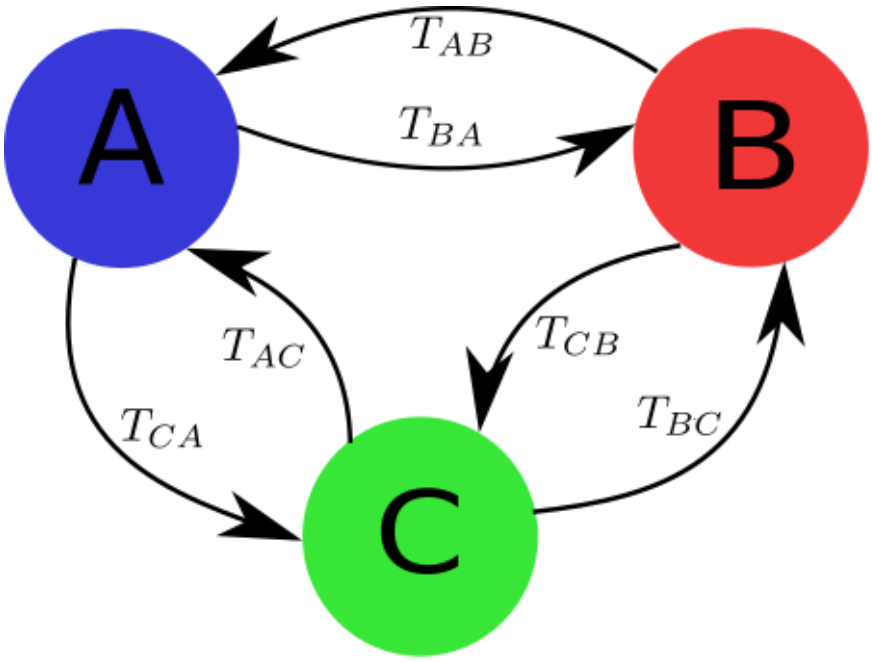


Figure 1.1: A simple example of a Markov model.

discrete states. The breakthrough came from the seminal work by Zwanzig in 1983 [13]. In this work, Zwanzig considered projecting a continuous space classical dynamics on to some discrete states as shown in figure 1.2. He commented that a theory for the movement of a dynamical system between regions of configuration space could be of value in the context of studying the metastability of supercooled fluids.

To do this, he projected his continuous Markovian dynamics on to a discrete space, introducing memory in to the system. By assuming a short memory approximation he was able to achieve a discrete Markovian description of the system dynamics. This assumption of the short memory approximation is non-trivial as it assumes that the system quickly equilibrates within its current state. Acknowledging this Zwanzig concluded that only if the discrete states to project on to were chosen 'sensibly' and the dynamics was 'sufficiently complex' would the system then not retain memory of reaching its current coarse-grained state, and one could obtain a 'remarkably simple' expression for transition rates between the clustered system states.

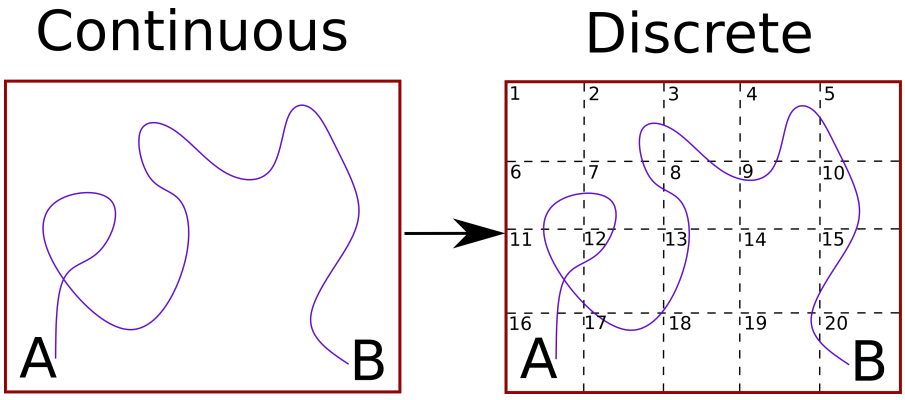


Figure 1.2: Visualisation of Zwanzigs discretization.

This result allowed the theory of Markov models to be applied to a much broader range of problems than previously as now complex continuous dynamics could be accurately modelled by a much simpler and intuitively appealing Markov model with a discrete set of states.

### 1.1.2 From Zwanzig to the Markov State Model

Building upon the work of Zwanzig [13] (and other practioners/advocators of the master equation methods for transition networks [14–18]), the late twentieth century saw a marked increase in the application of these discrete memoryless models to provide interpretative models of biophysical systems [19–22][1].

For many experimentalists, the challenges of modeling atomistic-level kinetics from experiment led to molecular dynamics (MD) simulation being employed to capture dynamics. Consider for example 'The Protein Folding Problem', the question of how a protein transforms from its unfolded to folded state. While experimentally, identifying the crystal structures of the initial and final state is possible, examining the mechanism through which folding occurs is much more difficult [23]. MD allows for the full resolution dynamics to be simulated and examined. However the difficulty of achieving

[1]These early master equation methods were not employed in the modelling of molecular simulation data but in somewhat different contexts, such as chemical reaction rate theory or for Monte Carlo simulations of polymer models of proteins.

physically relevant timescales with sufficient statistical sampling from MD simulation led to a huge surge of research in to improved hardware [24–30], software [31–34] and data modeling.

In terms of data modeling, the vital breakthrough came with the realisation that this theory first refined by Zwanzig was perfectly suited to the problem at hand. The MD systems being studied were effectively random walkers through a continuous configuration space and the end users wanted to have an easily interpretable discrete state Markov model. Even more importantly, these memoryless master equations, called Markov State Models (MSMs), allowed for the trajectories of multiple independent simulations to be combined in a statistically optimal and rigorous manner to create a single dynamic model where the conformations observed during the simulation each belonged to one of the discrete states [35][2]. In the following months and years, Markov models became the primary means of extracting kinetic insight from simulations and in particular for protein folding [36–42].

MSMs became popular within the community for three main reasons [43]:

- They made it much easier to conceptualize and understand the highly complex kinetic processes observed in MD trajectories.
- The mathematical formalism allowed for a concrete connection to experimental observables.
- They helped to accelerate MD simulations by suggesting configurations from which to reinitialise trajectories.

In the following sections of this introductory chapter, the necessary background and theory for the coming chapters is covered in depth. For further details on the state of the art in MSM construction, see recent reviews [44,45]. In section 1.2 the mathematical formulations of discrete Markov chains and master equations are introduced along with

[2]This initial application of memoryless master equations to the modeling of molecular simulation coined the term Markov state modeling.

some illustrative examples. In section 1.3, the field of molecular dynamics is presented since although the theory of MSMs is broadly applicable to a wide range of fields[3] the primary application at the forefront of our minds will be for modeling atomistic simulations. Section 1.4 covers the practical steps involved in building Markov models from data and reviews the last twenty years of research in refining those steps. Finally section 1.5 outlines the topics covered in the chapters of this thesis and aims to give some context and motivation to the questions we will aim to answer.

## 1.2 Markov State Model Theory in a Nutshell

A Markov state model consists of two pieces of information; a discrete set of states $\{x\} = 1, ..., N$ which the system can occupy and either transition rates $k_{ij}$ ($i$,$j \in \{x\}$) which describe the rate at which the system moves between the states within $\{x\}$ or transition probabilities $T_{ij}$ which describe the probability that the system moves from one state to another[4]. The element $k_{ij}$ denotes the rate of transition from state $j$ to state $i$ (similarly for $T_{ij}$)[5].

With this model in hand, a differential equation can be written down for the evolution of the probability to occupy a particular state at a given time $p_j(t)$.

$$\frac{dp_j(t)}{dt} = \sum_{i \neq j} \Big[ k_{ji} p_i(t) - k_{ij} p_j(t) \Big] \tag{1.2}$$

Equation 1.2 is called a master equation. Intuitively, the change in the probability to occupy state $j$ will depend on the flow of probability in and out of the state. The term $\sum_i k_{ji} p_i$ is the rate of transition from all states in to state $j$. Similarly, $\sum_i k_{ij} p_j$ is the rate of transition from state $j$ in to any other state.

[3] And we will in fact apply some of our later results to general geometric networks.

[4] The distinction between the transition rate and transition probability formulations is the difference between continuous and discrete time. The transition rates are given in units of inverse time and so are continuous. Meanwhile the transition probabilities are the likelihood to make a transition in a discrete interval of time.

[5] It should be noted that this notation is not universal and many MSM practioners favour the alternate notation (where $k_{ij}$ denotes the rate of transition from $i$ to $j$). This should be borne in mind when accessing any documents referenced in this document.

By defining a rate matrix $[\mathbf{K}]_{ij} = k_{ij}$ and a probability vector $\mathbf{P}$ with elements $p_i$ then the master equation can be written in a compact form as in equation 1.3. In this formulation, the diagonal elements are given by $k_{ii} = -\sum_{j \neq i} k_{ji}$.

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{K}\mathbf{P}(t) \tag{1.3}$$

This time-evolution equation has the following solution in terms of the initial probability vector.

$$\mathbf{P}(\tau) = e^{\mathbf{K}\tau}\mathbf{P}(0) \tag{1.4}$$

From equation 1.4, the quantity $e^{\mathbf{K}\tau}$ is the propagator which, given the probability distribution at some time, can be used to compute the probability at a later time. By writing out equation 1.4 in element wise form, it can be seen that the elements of the propagator must be conditional probabilities.

$$p_j(\tau) = \sum_i [e^{\mathbf{K}\tau}]_{ji} p_i(0) \tag{1.5}$$

$$p_j(\tau) = \sum_i p(j, \tau | i, 0) p_i(0) \tag{1.6}$$

1.5 is the element wise version of 1.4 while 1.6 is a basic result of probability theory. $p(j, \tau|i, 0)$ is the conditional probability to be in state $j$ at time $\tau$ given that the system was in $i$ at time 0. By connecting that equations 1.5 and 1.6 must both be true then the elements of the propagator must be the conditional probabilities. This matrix of propagators is defined to be the transition probability matrix $\mathbf{T}(\tau)$ (with elements $T_{ij}(\tau) = p(j, \tau|, i, 0)$). The quantity $\tau$ is called the lagtime of the model, it is the time-step in the discrete time transition probability formulation.

$$\mathbf{T}(\tau) = e^{\mathbf{K}\tau} \tag{1.7}$$

In the Markovian framework, it is assumed that the conditional probabilities are independent of the particular time. In other words, if one has a probability distribution at a time $t$ and wants to calculate the distribution some time $\tau$ later then the propagator will depend only on the time-step $\tau$ and not on the time $t$.

$$p(j, t+\tau|i, t) = p(j, \tau|i, 0) \qquad \forall t \tag{1.8}$$

Further to this, if the system is Markovian then the model quality should not be sensitive to the particular choice of lagtime $\tau$. If a vector $\mathbf{P}(t)$ is propagated twice with $\mathbf{T}(\tau)$ then this should be equivalent to propagating once with $\mathbf{T}(2\tau)$. This requirement is known as the Chapman-Kolmogorow (CK) condition[6].

$$\mathbf{T}(n\tau) = \mathbf{T}(\tau)^n \tag{1.9}$$

It should also be observed that the equation 1.7 does not establish a one-to-one relationship between $\mathbf{K}$ and $\mathbf{T}$ due to the matrix exponential. Each $\mathbf{K}$ has (for a particular value of $\tau$) a unique corresponding $\mathbf{T}(\tau)$. However, the inverse is not true. There may be multiple distinct $\mathbf{K}$ which can give rise to identical $\mathbf{T}(\tau)$. This is known as the embedding problem[7].

### 1.2.1 Spectral Decomposition

Once the rate matrix $\mathbf{K}$ or the transition probability matrix $\mathbf{T}(\tau)$ has been obtained the dynamical behaviour of the system can be studied by examining the eigenvalues and eigenvectors (spectral decomposition). As the link between $\mathbf{K}$ and $\mathbf{T}(\tau)$ has been established in equation 1.7 the following discussion of the spectral decomposition of $\mathbf{K}$ can be performed analogously for $\mathbf{T}(\tau)$.

[6]The Chapman-Kolmogorow condition is a necessary but not sufficient condition for Markovianity. It is typically used as a check to build evidence for the memoryless nature of the model but it does not guarantee that the system is Markovian.

[7]This problem can be equivalently stated as follows. Given a matrix $\mathbf{T}(\tau)$ satisfying the minimum requirements of a transition probability matrix (elements all non-negative and rows summing to one) then does there exist a matrix $\mathbf{T}(\tau/2)$ such that $\mathbf{T}(\tau/2)^2 = \mathbf{T}(\tau)$?

Spectral decomposition refers to the unique description of a matrix by a set of eigenvalues and linearly independent eigenvectors. The left and right eigenvectors ($\psi^L$,$\psi^R$) of the rate matrix $\mathbf{K}$ are given by the vectors satisfying the eigenvalue ($\lambda_n$) equations 1.10 and 1.11.

$$\mathbf{K}\psi_n^R = \lambda_n \psi_n^R \tag{1.10}$$

$$\psi_n^L \mathbf{K} = \lambda_n \psi_n^L \tag{1.11}$$

For a rate matrix as defined previously (negative diagonal, non-negative off diagonal and columns summing to one) there will be one zero eigenvalue with all the rest being negative. To illustrate this, consider the following example rate matrix $\mathbf{K}$ in equation 1.12.

$$\mathbf{K} = \begin{bmatrix} -a & b \\ a & -b \end{bmatrix} \tag{1.12}$$

First consider that since all the rows sum to zero that $[1, 1]$ must be a left eigenvector of the matrix with eigenvalue $\lambda_1 = 0$. Secondly the trace of the matrix is negative $(-a - b)$ so since the trace is the sum of the eigenvalues, the second eigenvalue of the matrix must negative with $\lambda_2 = -a - b$. As such the eigenvalues of the rate matrix can be arranged in descending order to provide a natural ordering.

$$0 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N \tag{1.13}$$

The right eigenvector corresponding to the zero eigenvalue is of special interest as this means the rate of change of the vector with respect to time is zero. This is the stationary probability vector and it gives the equilibrium probabilities of the states of

the system.

$$\mathbf{KP} = 0 \implies \frac{d\mathbf{P}}{dt} = 0 \implies \mathbf{P} = \mathbf{P}^{eq} \tag{1.14}$$

The left and right eigenvectors can be related via the equilibrium probability.

$$\psi_n^L \mathbf{P}^{eq} = \psi_n^R \tag{1.15}$$

The eigenvectors also satisfy some useful orthogonality relations laid out below in equations 1.16 and 1.17. These show that we have orthogonality both for a sum over elements and for a sum over the vectors.

$$\sum_{i=1}^{N} \psi_n^L(i)\psi_{n'}^R(i) = \delta_{nn'} \tag{1.16}$$

$$\sum_{n=1}^{N} \psi_n^L(i)\psi_n^R(j) = \delta_{ij} \tag{1.17}$$

If the eigenvalues/vectors have been obtained (and normalized to satisfy the orthogonality relations[8]) then each element of the rate matrix can be written as a linear combination of these quantities as in equation 1.18. This way of representing rate matrix elements will be particularly useful in our later analysis.

$$\mathbf{K}_{ji} = \sum_{n=1}^{N} \lambda_n \psi_n^R(j)\psi_n^L(i) \tag{1.18}$$

### 1.2.2 Rate Matrix Dynamics and Relaxation Timescales

The eigenvalues and vectors of the rate matrix describe the dynamics of the Markovian process. Each eigenvalue is linked to a relaxation timescale and the corresponding eigenvector gives sets of states between which the relaxation process occurs. A relax-

[8]Note that, in practice, software to spectrally decompose a matrix does not automatically normalize or make orthogonal the eigenvectors.

ation timescale describes the speed at which the systems initial probability distribution converges towards its equilibrium distribution.

The best way to see the link between eigenvalues and relaxation processes is to derive the time dependent probabilities of a two state Markov system [46]. Consider a two state system described by the rate matrix in equation 1.19.

$$\mathbf{K} = \begin{pmatrix} -k_{21} & k_{12} \\ k_{21} & -k_{12} \end{pmatrix} \tag{1.19}$$

The master equation for the probability of state 1 is as in equation 1.20

$$\frac{dp_1(t)}{dt} = -k_{21}p_1(t) + k_{12}p_2(t) \tag{1.20}$$

Using the fact that $p_1(t) + p_2(t) = 1$ (and so $p_2(t) = 1 - p_1(t)$), equation 1.20 is written as a single variable differential equation in 1.21 and this is solved for $p_1(t)$.

$$\frac{dp_1(t)}{dt} = (-k_{21} - k_{12})p_1(t) + k_{12} \tag{1.21}$$

$$p_1(t) = \mathbf{C}e^{-(k_{12}+k_{21})t} + \frac{k_{12}}{k_{12}+k_{21}} \tag{1.22}$$

$\mathbf{C}$ is a constant which can be fixed by observing that i) $\frac{k_{12}}{k_{12}+k_{21}} = p_1^{eq}$ (from the first right eigenvector of $\mathbf{K}$) and ii) $\mathbf{C}$ must be such that the right hand side of the equation is $p_1(0)$ at $t = 0$. Additionally, the eigenvalues of $\mathbf{K}$ are $\lambda_1 = 0$ and $\lambda_2 = -k_{12} - k_{21}$.

$$p_1(t) = (p_1(0) - p_1^{eq})e^{-|\lambda_2|t} + p_1^{eq} \tag{1.23}$$

From this it becomes clear that as time evolves, the distribution of probability will tend towards the equilibrium distribution with a timescale characterised by $\lambda_2$.

$$\tau_2^{\text{rel}} = \frac{1}{|\lambda_2|} \tag{1.24}$$

For systems of more states, there will be more timescales (each given by the inverse of the magnitude of the eigenvalue). The corresponding eigenvectors will have positive and negative values. Hence each timescale can be interpreted as the time with which the rate matrix moves probability density between the oppositely signed regions of the corresponding eigenvector.

As an example, the first three eigenvectors of a sample two state system are shown in figure 1.3. The first eigenvector (blue circles) shows the equilibrium probabilities with the probability focused in to two main regions of space. The second eigenvector (orange circles) shows the regions of space between which it is slowest to move the probability density. Similarly, the third eigenvector (green circles) shows the regions of space between which the timescale is characterised by the third eigenvalue.

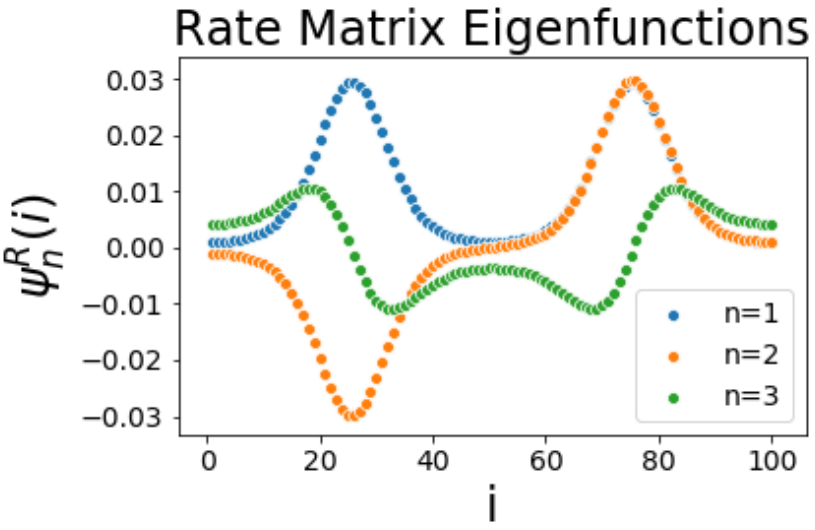


Figure 1.3: First three eigenfunctions of a sample system.

### Interlude: Link between Markov rate matrices and Quantum Mechanics

As an aside for the reader who is more familiar with quantum mechanics, one can notice the close connection between the mathematical formulation laid out so far and that of quantum mechanics. Consider for example the Schrodinger equation for the time-evolution of the wavefunction.

$$i\hbar\frac{d\Psi}{dt} = \mathbf{H}\Psi \tag{1.25}$$

This is a time-evolution of a probability density (as opposed to a probability vector) characterised by an operator $\mathbf{H}$, equivalently the quantum mechanical propagator is written as $e^{-i\mathbf{H}t/\hbar}$. The similarity in the form of the two equations and associated propagators is clear to see. The main difference between the two theories is that the operator in quantum mechanics is required to be self-adjoint whereas in using Markov matrices to describe diffusive processes we relax that condition [47].

## 1.3 Molecular Dynamics Simulations

It is clear how this new Markovian formalism proved useful as an approach for modelling complex systems. In this section, one particular field of study, molecular dynamics (MD) simulations, is introduced as it will be the application at the forefront of our minds for the chapters to come. Following this, the development and practical application of MSMs to this MD field of study will be discussed.

### 1.3.1 Origin and Basics of Molecular Dynamics

Since the development of transistor based computers in the 1950s, researchers have used these machines to simulate simplified models of systems. One of the first instances of

a molecular based simulation was performed by Fermi, Pasta, Ulam and Tsingou[9] [49]. In this work, the proposed idea was to model atoms in a crystal by simulating a 1-D chain of molecules joined by springs obeying Hookes law plus a non-linear interaction term. The simulation yielded surprising results which required the development of new theoretical results (solitons) to explain. The study provided a proof of concept for the role of computer simulations as a substitute for traditional experiment.

With the continued success of Moores law for computing [50], the power available to academic researchers has grown exponentially over the intervening time and opened up the possibility of simulating computational models of ever more complex systems. The example of this which we examine in detail here is for atomistic systems. These simulations were developed in the 1970s [51–53] and aim to understand the behaviour of macromolecules by approximating atoms as hard spheres which exert forces on each other by some classical interaction terms. These forces come in two forms, those which describe forces between bonded atoms and those which describe non-bonded interactions. The bonded potential energies come from fluctuations in the atomic bond lengths, angles and dihedrals. The non-bonded terms come from the Van Der Waals and electrostatic interactions. The accurate parameterisation of these potential terms has been one of the most significant technical challenges in obtaining sufficiently accurate simulations [54–57].

This comes with the essential caveat that a computer can never simulate reality, it can only simulate the model of reality it is provided with. As such, any simulation results will carry with them any flaws inherent to the models they are based upon. For a more thorough discussion of the philosophy of simulation see the introductory chapters of Frenkel and Smits 'Understanding Molecular Simulation' [58].

Classical molecular simulations are based on Newtons second law (equation 1.26) which relates the acceleration of an object with mass to the force applied. In contrast

[9]Mary Tsingou was not included as an author on this paper and receives only a brief acknowledgement for the 'efficient coding of the problems and for running the computations on the Los Alamos MANIAC machine' [48].

there exist also quantum molecular simulations which use Schrodingers equation to describe the motion rather than Newtons equation. These are more accurate but also significantly more computationally expensive and become prohibitively so for systems with large numbers of atoms.

$$F = ma \tag{1.26}$$

This implies that for a system of particles (indexed by $i$) with masses $m_i$ the acceleration $a_i$ which that particle experiences is proportional to the force experienced $F_i$. As the force is the negative gradient of the potential energy $v_i$ and the acceleration is the second derivative of position $x_i$, equation 1.26 can be rewritten as equation 1.27.

$$\frac{d^2x_i}{dt^2} = \frac{-1}{m_i}\frac{dv_i}{dx} \tag{1.27}$$

So knowing the present positions of all the particles in the system as well as the potential energy function experienced at each point in space then the differentials in equation 1.27 can be approximated numerically using finite differences [59].

$$\frac{x_i(t+\delta t) - 2x_i(t) + x_i(t-\delta t)}{(\delta t)^2} = \frac{-1}{m_i}\frac{v_i(x_i(t+\delta t)) - v_i(x_i(t))}{x_i(t+\delta t) - x_i(t)} \tag{1.28}$$

$\delta t$ represents the time step introduced by discretizing the system dynamics. Equation 1.28 is obtained by a method known as central differences, several alternate methods for approximating derivatives exist but the key idea is that this approach leads to an update rule for calculating $x_i(t+\delta t)$ from the present positions $x_i(t)$, the past positions $x_i(t-\delta t)$ and the potential energy $v_i$.

Then for the accurate simulation of some physical system, one requires the initial positions of all atoms in the system and a force field which describes all the forces of interest experienced by the particles of the system. This force field parameterisation is a notoriously difficult experimental/theoretical challenge. Even the calculation of forces for something as a simple as a water molecule is the subject of significant debate [60,61].

Despite the theoretical challenges regarding complex choices of parameters, MD simulation has been hugely popular and successful as a tool to compliment traditional experiment. It is typically integrated as part of an iterative pipeline where the results of the experiment feed back in to the MD model. MD simulation has been used to better understand protein folding mechanisms [23, 62, 63], protein-ligand un/binding free energies [64–66], membrane-crossing [67–69] as well as many non-biological applications such as carbon nanotubes [70].

MD simulations produce the positions of all atoms in the system at each discrete time step. This atomistic level data then requires interpretation. There are a range of visualisation tools which allow these trajectories to be viewed as a movie, with each timestep representing a single frame. While this can be helpful for some insight (for example, viewing the route a drug molecule takes to reach the pocket), it does not immediately produce any quantitative measurement to compare to experiment.

It is here that Markov state modelling emerges as a solution, it allows all of the observed kinetic behaviour to be converted in to a statistically optimal kinetic model from which experimental observables can be computed. As an added benefit, if one decides to run extra simulations then the new data can be easily added to the existing data to create a single Markov model. This ability to combine independent trajectories in to a single humanly interpretable model allows for the construction of an iterative protocol where the Markov model itself feeds back to inform the new simulations to be run.

The question then is given this high dimensional MD simulation time series data how does one construct a MSM. We address this question next in section 1.4.

## 1.4 Practical Guide to Constructing a Markov Models

In practical examples, one is interested in describing a real world system by a Markov model given some time series data. There are typically five steps in constructing an MSM [43].

1. Identify the microstates of the model and assign each observation in time to a single microstate.
2. Choose the lagtime at which the MSM should be constructed via the CK test.
3. Calculate the transition probabilities between microstates.
4. Cluster together microstates to create coarse-grained macrostates.
5. Compute the transition rates on the coarse-grained state space to arrive at a humanly interpretable kinetic network description of the system.

### 1.4.1 Defining Microstates

Each discrete time observation in the simulation needs to be assigned a discrete state label in order to construct the model. At each instance in time, the state of the system is described by the XYZ coordinates of every atom in the system. There are two main approaches to assign states, using a reaction coordinate or a density based approach.

- Reaction coordinate: If a feature can be chosen so that the behaviour of interest is well characterised by movement along this coordinate then each simulation observation can be classified based on where it falls along this coordinate[10]. This coordinate might be a particularly interesting bond or dihedral angle.

[10]The effective identification of reaction coordinates is an entire area of research unto itself, particularly in the context of machine learning. Common methods in the Markov modelling field include PCA [71] (which finds directions of maximal variance) and TICA [72] (which identifies a maximally slow subspace of input dimensions). An open challenge in the field is to how to characterize coordinates which are biophysically relevant but have relatively fast dynamics [73]. This is also one of the most important questions in the field of machine learning research.

- Density based clustering: One can take some higher dimensional set of features (or just the raw XYZ coordinates) and identify states by grouping together frames which are closest together in this full configuration space.

Which of these two methods is preferential to use? This is not a straightforward question to answer and depends greatly on the system being studied. For example, while the reaction coordinate identification method greatly simplifies the interpretation of the system it also requires the assumption that the degrees of freedom which have been ignored equilibrate much faster than the coordinate of interest. This may be problematic for systems where the interesting process is not the slowest occuring. Similarly, while the density based approach navigates around this issue by retaining a full high-dimensional representation of the system for clustering, it also is less directly interpretable.

As an illustration of these two approaches consider the case of a ligand-protein binding simulation [74, 75]. In the reaction coordinate framework, one might identify the 1D distance between the centre of mass of the ligand and the binding pocket as an appropriate feature and classify frames by discretizing this coordinate. In contrast, in the density based framework, one might identify states by taking the protein as fixed and using the XYZ coordinates of the ligand in the frame of reference of the protein. Then states can be identified by grouping frames with similar XYZ coordinates using traditional pattern recognition clustering (k-means [76], k-nearest neighbours [77] etc.).

### 1.4.2 Estimating Transition Probabilities

If we assume that we have managed to identify each frame of our trajectory with a discrete numeric label then the next step is to estimate the Markovian transition probabilities. We want to know, given our observation what is the most likely value for the transition probabilities to have? This is then a likelihood maximization problem. We write down the likelihood of the observed trajectory given some transition probablities and then optimize to find the transition probabilities which maximize the likelihood.

In probability, the likelihood of observing multiple events is multiplicative so the likelihood of the entire trajectory is the product of the probability of each transition.

$$L = T_{x(2)x(1)}T_{x(3)x(2)}T_{x(4)x(3)}...T_{x(N)x(N-1)} \tag{1.29}$$

This is equivalent to taking the probability of each possible transition raised to the power of the number of times it was observed.

$$L = \prod_{i,j} T_{ji}^{C_{ji}} \tag{1.30}$$

Here $C$ is a count matrix where the element $C_{ji}$ contains the number of observed transitions from state $i$ to state $j$. It is usually easier to work with the log-likelihood since it changes the product to a summation.

$$\log(L) = \sum_{i,j} C_{ji} \log(T_{ji}) \tag{1.31}$$

Since the only information we possess is the transition count matrix $C$, we want to ask the question, given the observed transitions, what is the most likely underlying probabilities $T_{ji}$ which gave rise to these observations? In other words, what values of $T_{ji}$ maximise the likelihood of the trajectory $L$. We want to find the values of $T_{ji}$ which maximise this, however we do not have complete freedom of choice[11], we require that the columns of our transition matrix sum to 1 ($\sum_j T_{ji} = 1$). We add a Lagrange multiplier condition to enforce this property, this condition is enforced by $\lambda_i$ in equation 1.32.

$$\log(L) = \sum_{i,j} C_{ji} \log(T_{ji}) - \sum_i \lambda_i (\sum_j T_{ji} - 1) \tag{1.32}$$

[11] If we just differentiate immediately without placing constraints on our search, we find that the probabilities are infinite since this will obviously maximise the likelihood.

Differentiating with respect to $T_{ji}$ in equation 1.32, one can obtain equation 1.33.

$$\frac{d\log(L)}{dT_{ji}} = \frac{C_{ji}}{T_{ji}} - \lambda_i = 0 \tag{1.33}$$

Summing over $j$ yields that $\lambda_i = \sum_j C_{ji}$ and so the maximum likelihood estimate for the transition probability from $i$ to $j$ is given simply by the number of observed transitions from $i$ to $j$ divided by the number of observations in $i$.

$$T_{ji} = \frac{C_{ji}}{\sum_j C_{ji}} \tag{1.34}$$

The transition count matrix $C$ is in fact lagtime $\tau$ dependent, it requires us to make a choice of how far in time we will consider two states to be separated to qualify as an observed transition as shown in figure 1.4.

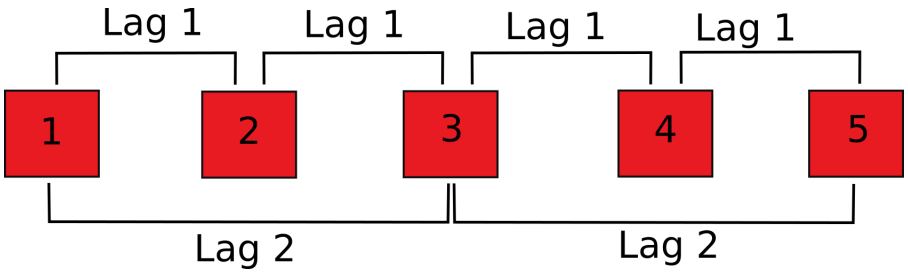


Figure 1.4: Visualisation of lagtimes.

This is where we use the CK test to decide what lagtime to use to construct the count matrix $C$. We want that the transition probabilities are insensitive to the particular choice of lagtime. The derivation of Zwanzig upon which the validity of this formalism relies makes the assumption of the short-memory approximation and so that once the system enters a state it 'quickly' forgets which state it entered from. The term 'quickly' is relative to what we have defined to be the timestep of our model. If our lagtime is very short then the system will retain some memory of its previous state. Conversely, if the lagtime is very long then the transition probability becomes not only independent

of its history but also independent of the current state.

$$\lim_{\tau \to \infty} p(j, \tau | i, 0) = p_j^{eq} \tag{1.35}$$

As mentioned earlier, the CK test is necessary but not sufficient for ensuring Markovianity. For true Markovianity, one requires that equation 1.9 is true for all $n$. In practice this amounts to computing the quantities in equation 1.9 for a range of values of $\tau$ and selecting a value such that the equation holds true (within some reasonable tolerance) for several $n$.

### 1.4.3 Clustering Microstates

Once an MSM has been constructed we have a large system of nodes upon which the true dynamics are well approximated by a Markovian description. We now want to obtain a coarse-graining by aggregating these states to allow us to paint a simple and intuitive painting of the systems kinetics. In this second round of coarsening our description of the system will become less accurate as information is removed. This ultimately means that one finishes with two Markov models which serve different purposes, i) a high-dimensional MSM which describes the true dynamics as accurately as possible and can be used for computing experimental observables and ii) a low-dimensional MSM which provides a clear understanding of the system and can be used for enhanced sampling protocols.

To achieve this, the microstates from the initial MSM are grouped[12] in to physically interpretable macrostates and the new rates between them calculated (as in figure 1.5). It is this step which will be the main subject of investigation in this thesis. As such we will present a more comprehensive review of clustering methods in chapter 2.

[12]The terms coarse-graining, clustering and dimensionality reduction can all be used to describe the grouping together of states in a MSM to produce a new MSM with fewer states.

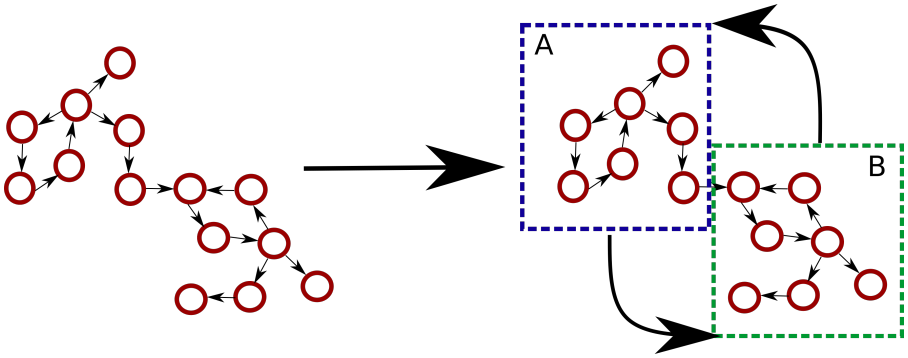


Figure 1.5: Visualisation of microstate clustering.

## 1.5 Thesis Outline

For the remainder of the thesis, we will assume that we have a Markov state model which has been constructed by some selection (or variation thereon) of the methods described previously. We will typically further assume the existence of a rate matrix which accurately describes the kinetics of the system of interest as this will uniquely define an associated transition probability matrix.

The primary thrust of this thesis from this point on will be to examine how MSM theory can automate and simplify the extraction of useful insight from time series data without comprising accuracy. The central objective will be to remove the ambiguity which exists at present in several areas of MSM usage and can be troublesome for novice users.

In Chapters 2, 3 and 4 we will present the primary work of the thesis which concerns a protocol for optimally and automatically identifying key states via coarse-graining of MSMs. In chapter 2, we describe our proposed algorithm to use the relaxation timescales as variational parameters and show how these can be used to identify both metastable and transition states in some simple test cases. In chapter 3, we delve deeper in to the theory and provide some analytic explanation and justification for the effectiveness by interpreting the relaxation times in terms of mean first passage times. The method developed, while effective, scales poorly for systems with high-

dimensionality (large numbers of microstates, clusters or reaction coordinates). Hence in chapter 4, we describe a parallel tempering based approach for extending the method to these high-dimensional systems as well as demonstrating how it might be applied to clustering of nodes in geometric graphs. In Chapter 5 we derive a new equation for effectively estimating relaxation times from MSMs in the long-lagtime limit. We further demonstrate how this equation can be used to accurately obtain relaxation times from systems with limited data. In Chapter 6 we examine a new MSM approach to calculating membrane permeabilities from MD simulations. This new analysis method achieves equivalently accurate results to existing methods while being significantly simpler to implement. Finally in chapter 7, we reexamine the broader state of the field and the main outstanding questions to be addressed in the future.

*"The most important possible thing you can do is do a lot of work."*

Ira Glass

# 2

# Variational Coarse-Graining Of Markovian Dynamics

## 2.1 Introduction

With the studying of complex systems using Markov models comes the need to extract meaningful and actionable insight in to the system. One barrier to understanding is the typically high dimensionality of these models. For this reason, there is a need for robust and interpretable lower dimensional representations of these systems by ag-

gregation of the microstates in to clusters (macrostates). Typically, in computational applications, one constructs a Markov model with a large number of states and performs a dimensionality reduction to construct a new model with a much smaller set of clustered states[1] [63, 78].

This dimensionality reduction will inevitably come with the loss of information. This will depend both on the composition and the definition of the kinetics of the clusters. There are two types of clusters which we are interested in identifying.

- Metastable Clusters: Groups of microstates that move very quickly amongst themselves but are slow to move to other macrostates.
- Transition States: Groups of microstates that are occupied only very briefly but are important in linking together the major metastable clusters[2].

Our goal is to be able to identify these types of states in an optimized and automatic manner. The automatic aspect of this is neccessary as at present most dimensionality reduction procedures require a significant degree of human input. In this chapter we propose an algorithmic protocol to perform clustering which will require human input only in the definition and construction of the initial high-dimensional model. This algorithm may have a wide range of major benefits for the field. It can be used to automate the production of intuitive and humanly understandable descriptions of kinetic pathways and mechanisms.

Additionally, the automatic ability to identify simulation frames as belonging to transition states could help to accelerate MD simulations. By reinitialising trajectories from these transition state frames, one can adopt a 'shooting from the top' approach [79, 80] to improve the sampling of configuration space.

[1] The definition of what constitutes 'large' will be very system dependent. In most practical applications, a system with over 1000 microstates is considered large.

[2] In transition state theory, the transition state is described by a dividing surface with no volume. In practice, we want to be able to assign some observations from our simulation as being transition state configurations. This necessitates having a transition state of finite volume.

### 2.1.1 Definitions of Clusters

Previously we provided a qualitative definition of the difference between metastable and transition state clusters. Here we now establish the quantitative definition which we will use to inform the rest of the work in this thesis. We define a transition state $X_n$ along the pathway described by the relaxation time $\tau_n$ as a state such that for a MSM constructed at lagtime $\tau_n$, there exist two (or more) states that the state has a higher probability of transitioning to than remaining in its current state[3].

Phrased more formally, it is a state $X_n$ such that there exists distinct $i$ and $j$ where $M_{jX_n}(\tau_n) > M_{X_nX_n}(\tau_n)$ and $M_{iX_n}(\tau_n) > M_{X_nX_n}(\tau_n)$. So on the timescale of the pathway being described the state is unlikely to stay in its current state. With this working definition of a transition state, we will over the course of the next three chapters outline the results of our investigation to develop an automatic approach for identifying these states.

## 2.2 Existing Coarse-Graining Methods

Before we attempt to develop our own method, we first give an overview of the existing coarse-graining methods within the MD simulation community.

### 2.2.1 Perron Cluster Analysis

One of the earliest developed methods for performing this reduction came from a number of researchers at the Zuse Institute Berlin. In a pair of papers at the turn of the millennium, Peter Deuflhard and colleagues proposed a method for identifying 'invariant aggregates using the sign structure of the eigenvectors corresponding to the Perron cluster of eigenvalues' [81, 82], termed Perron Cluster Analysis (or PCCA[4] for short).

[3] This is likely an overly generous definition of a transition state which could be refined further. The logic is that a metastable state will not satisfy this condition. However we would expect a transition state to satisfy this condition for a lagtime much less than the relaxation time. In practice, in this thesis all the transition states identified satisfy this condition very comfortably.

[4] The additional C in the initialism exists to avoid confusion with the Principal Components Analysis (PCA) algorithm.

The 'Perron cluster' consists of the set of eigenvalues which are close to 1 in value (those corresponding to the slow processes between metastable clusters).

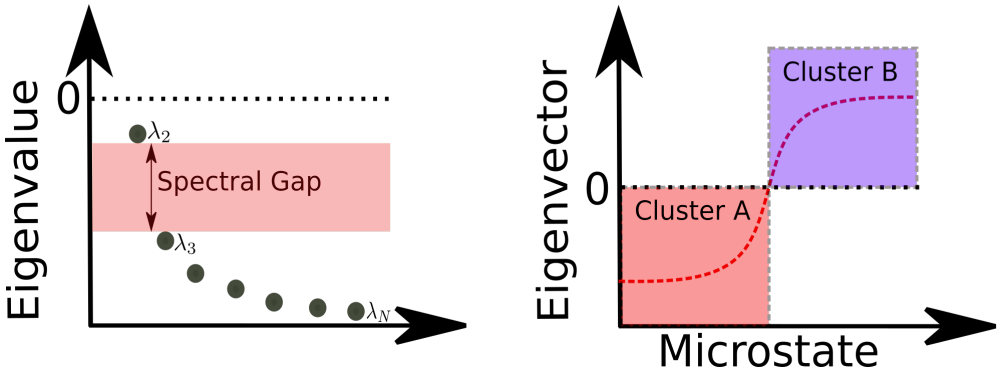


Figure 2.1: Illustration of PCCA clustering in to two metastable states.

By finding a spectral gap, one could identify the number of metastable states within the system. If for example, the gap was between $\lambda_2$ and $\lambda_3$ then one could say that there were two metastable states since there was one slow process (described by $\lambda_2$) as illustrated in figure 2.1.

Moreover, from this spectral gap one can then inspect the eigenvector corresponding to the slow processes and identify the appropriate clustering based on the signs of the elements. All microstates corresponding to negative elements are placed in to one cluster and all the positive elements in to the other as in figure 2.1. For more than two clusters, one essentially takes the relevant eigenvectors (corresponding to Perron cluster eigenvalues) and performs a density/sign based clustering of states in this eigenvector space. This approach has the clear drawback that it throws out all information related to the magnitude of the eigenvector components and as such elements near the zero values are liable to be misclassified [83].

A series of papers from this research group by Marcus Weber, Susanna Kube and others further refined PCCA to develop PCCA+ and demonstrated its success at identifying stable conformations from simulation data [84–91]. PCCA+ is a more sophisticated approach which accounts for eigenvector elements magnitudes and provides a

fuzzy clustering (as opposed to the crisp clustering of PCCA). In PCCA+, each microstate has a probability of membership for each macrostate as shown in figure 2.2 where the blue line represents the probability of a microstate belonging to cluster A and similarly for the red line and cluster B.

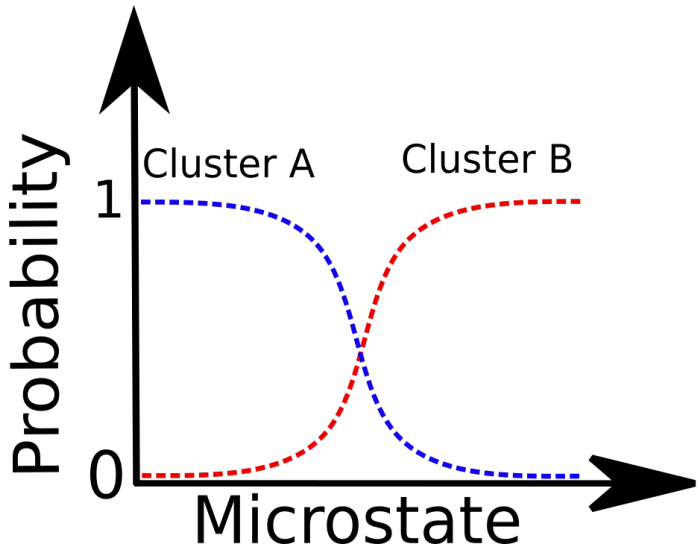


Figure 2.2: Illustration of PCCA+ clustering in to two metastable states.

### 2.2.2 Other Methods

More recent methods include the hierarchical Nystrom method which applies a scheme of hierarchical clustering to try and deal with the effect of poorly sampled states [92]. Poorly sampled states can appear to be kinetically distinct when building MSMs as due to the rarity of transition the transition rate connecting them is very slow. The Nystrom method uses a hierarchical approach by assigning poorly sampled states to the cluster to which they have the highest probability of transition.

Other early approaches used the concept of likelihood maximization for model dimensionality reduction [93,94] and similarly to the PCCA and Nystrom methods, these approaches have proven effective at identifying metastable states. The Bayesian ag-

glomerative clustering engine (BACE) approach uses an information theory approach to minimize the information lost by the clustering and to explicitly quantify the error and uncertainty introduced by the clustering [95]. Others have tried to leverage conventional machine learning approaches such as Ward clustering [96].

When surveying the existing methods to identify clusters of states two common themes emerge.

1. These methods focus on using eigenvectors to identify metastable regions. Eigenvalues are used primarily to inform the number of metastable regions in the MSM via the spectral gap. This can be problematic for systems which do not display a clear separation of timescales.

2. Some methods are developed to deal with transition states via fuzzy clustering but not to explicitly identify them as distinct regions of space in the low-dimensional representation.

## 2.3 A New Approach to Coarse-Graining

In this chapter, a new approach is described which is inspired by these two observations and uses the eigenvalues explicitly to automatically identify both metastable and transition states. The proposed method searches through potential clusterings and for each calculates the MSM on the clustered space. The dominant timescales in the reduced MSM are then used as a variational parameter to quantitatively score the quality of the clustering. The process of describing and justifying this method requires three important questions to be answered.

1. Given a Markovian system and a particular clustering of states, how does one construct a reduced MSM so as to optimally preserve the original dynamics?

2. Given such a method to compute the reduced MSM, do the MSM timescales behave variationally with respect to the clustering?

3. Does such a protocol manage to identify both metastable and transition states?

In this chapter we will address these three questions and demonstrate the results obtained from applying this protocol to some simple test systems. Before we do this we need to introduce some additional theory, namely that of correlation functions.

## 2.4 Correlation Functions

A correlation function describes how variables co-vary over time. So if we observe a change in a particular variable at time $t$, what change do we expect in some other variable at a later time $t + \tau$?

Formally, the correlation function between two observables $f$ and $g$ at lagtime $\tau$ is obtained by examining a dynamical average.

$$c(\tau) = \langle f(x(\tau))g(x(0))\rangle \tag{2.1}$$

$x(\tau)$ is the position of the system at time $\tau$. This expectation value is calculated not just with times 0 and $\tau$ but with a dynamical average over all times.

$$\langle g(x(\tau))f(x(0))\rangle = \lim_{T\to\infty} \frac{1}{T}\int_0^T g(x(t+\tau))f(x(t))dt \tag{2.2}$$

Then from the ergodic hypothesis [97], a long time average is equivalent to an ensemble average so that we can write the correlation function as a sum over all microstates.

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T g(x(t+\tau))f(x(t))dt = \sum_{i,j} g(j)f(i)p(j,\tau|i,0)p_i^{eq} \tag{2.3}$$

This gives the correlation between two observables measured at a time separation $\tau$, and it satisfies, in equilibrium, the fluctuation-dissipation relation [98, 99].

In many studies of dynamical systems, correlation functions of the occupancy number observable are useful for gaining kinetic insight in to the system [46, 100–103]. The

occupancy-number observable $\theta_i(t)$ is defined to be 1 if the system is in state $i$ at time $t$ and 0 otherwise. As such it signals the state $i \in [1, 2 \ldots, n]$ in which the system is found at any given time, where $x(t)$ denotes a discrete reaction coordinate[5] that describes the system.

$$\theta_i(t) = \begin{cases} 1 & x(t) = i \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

This choice of variable acts like a delta function to pick out certain terms from the sum in the ensemble average. It also has the property that its average value is the equilibrium probability of the state.

$$\langle \theta_i(t) \rangle = p_i^{eq} \tag{2.5}$$

As a result the correlation function is now comprised of probabilities and propagators. Additionally, it is common to extract the long lagtime limit from the correlation function to obtain the connected correlator $\bar{C}$.

$$\bar{C}_{ij}(\tau) = p(i, \tau | j, 0) p_j^{eq} - p_i^{eq} p_j^{eq}. \tag{2.6}$$

We can write equation 2.6 in matrix notation as equation 2.7.

$$\bar{\mathbf{C}}(\tau) = [e^{\mathbf{K}\tau}]\mathbf{D}_n - \mathbf{D}_n \mathbf{D}_n^T \tag{2.7}$$

Where $\mathbf{D}_n$ is the diagonal matrix with $\mathbf{p}^{\text{eq}}$ along its diagonal, i.e. with entries $(\mathbf{D}_n)_{ij} = p_i^{\text{eq}} \delta_{ij}$. It can be more convenient to work with this quantity after Laplace transformation, when it is written as $\hat{\bar{\mathbf{C}}}$ as in equation 2.8.

$$\hat{\bar{\mathbf{C}}}(s) = (s\mathbf{I}_n - \mathbf{K})^{-1}\mathbf{D}_n - \frac{1}{s}\mathbf{D}_n \mathbf{D}_n^T \tag{2.8}$$

[5] Or a continuous coordinate which has been discretised via a binning procedure.

$(s\mathbf{I}_n - \mathbf{K})^{-1}$ is the propagator in Laplace space.

### Interlude: Detailed Balance

In the coming derivation, we will make the requirement that our projected system satisfies detailed balance [104]. Detailed balance is effectively the condition of thermodynamic equilibrium. This is equivalent to the statement that the probability of observing a transition from $i$ to $j$ is equal to the probability for the reverse transition from $j$ to $i$. This can be phrased formally in equation 2.9.

$$p(j, \tau|i, 0)p_i^{eq} = p(i, \tau|j, 0)p_j^{eq} \tag{2.9}$$

## 2.5 Low-dimensional Dynamics

Armed with the knowledge of correlation functions and their importance in statistical physics, the question can now be asked as to how to optimally project high dimensional dynamics on to a lower dimensional space. In this section, we show how projecting high-dimensional dynamics on to a lower dimensional space and enforcing detailed balance reproduces a physically interpretable criterion in terms of correlation functions.

Suppose that a projection operator $\mathcal{P}$ is used to project microstates down on to some sub-space. The projected probability vector is denoted $\mathbf{u} = \mathcal{P}\mathbf{p}$ and $\mathbf{v} = \mathbf{p} - \mathbf{u}$ its orthogonal projection $\mathbf{v} = \mathcal{Q}\mathbf{p}$, with $\mathcal{Q} = \mathbf{I}_n - \mathcal{P}$ and $\mathbf{I}_n$ the $n$-dimensional identity matrix. applying these projection the original definition of the master equation (equation 1.4) a pair of coupled differential equations for the projections of $\mathbf{p}$ can be obtained:

$$\frac{d\mathbf{u}}{dt} = \mathcal{P}\mathbf{K}\mathbf{u} + \mathcal{P}\mathbf{K}\mathbf{v} \tag{2.10}$$

$$\frac{d\mathbf{v}}{dt} = \mathcal{Q}\mathbf{K}\mathbf{u} + \mathcal{Q}\mathbf{K}\mathbf{v} \tag{2.11}$$

Solving the equation for $\mathbf{v}$, with initial condition $\mathbf{v}(0) = 0$, and substituting into the equation for $\mathbf{u}$ leads to a dynamical description involving only $\mathbf{u}$

$$\frac{d\mathbf{u}}{dt} = \int_0^t \mathbf{M}(t-\tau)\mathbf{u}(\tau)d\tau, \tag{2.12}$$

which is no longer Markovian, where

$$\mathbf{M}(t-\tau) = \mathcal{P}\mathbf{K}\delta(t-\tau) + \mathcal{P}\mathbf{K}e^{\mathcal{Q}\mathbf{K}(t-\tau)}\mathcal{Q}\mathbf{K} \tag{2.13}$$

is a memory kernel encoding the effective interaction between $\mathbf{u}$ and its past values, arising from interactions with the degrees of freedom that have been integrated out.

Suppose the objective is to cluster the microstates $i \in \{1, \ldots, n\}$ into $N < n$ macrostates, that are labeled with capital indices $I, J \in \{1, \ldots, N\}$ then one can define $\mathbf{P}$ the probabilities on the macrostates, which are related to $\mathbf{p}$ via $\mathbf{P} = \mathbf{A}^T\mathbf{p}$, where $\mathbf{A}$ is an $n \times N$ aggregation matrix with entries $A_{iI}$ equal to 1 if microstate $i \in I$ and zero otherwise. The macrostates probabilities also evolve according to a memory kernel equation.

$$\frac{d\mathbf{P}}{dt} = \int_0^t \mathbf{R}(t-\tau)\mathbf{P}(\tau)d\tau \tag{2.14}$$

By Laplace transforming equation 2.14 and rearranging yields the propagator $(s\mathbf{I}_N - \hat{\mathbf{R}}(s))^{-1}$ in Laplace space in equation 2.15.

$$\hat{\mathbf{P}}(s) = (s\mathbf{I}_N - \hat{\mathbf{R}}(s))^{-1}\mathbf{P}(0) \tag{2.15}$$

This Laplace transformed propagator can be used to express the correlator of the coarse-grained system as in equation 2.16.

$$\hat{\tilde{\mathbf{C}}}^{\mathrm{CG}}(s) = (s\mathbf{I}_N - \hat{\mathbf{R}}(s))^{-1}\mathbf{D}_N - \frac{1}{s}\mathbf{D}_N\mathbf{D}_N^T. \tag{2.16}$$

Here $\mathbf{D}_N$ is the diagonal matrix with the stationary solution of Eq. 2.14 $\mathbf{P}^{\mathrm{eq}}$ on the

diagonal.

Two key questions are which projection corresponds to the clustering protocol $\mathbf{A}$ and how the rate matrix of the coarse-grained system $\hat{\mathbf{R}}(s)$ is related to that of the original system $\mathbf{K}$. Defining the relation between $\mathbf{u}$ and $\mathbf{P}$ to be described by an $n \times N$ matrix $\mathbf{H}$ such that $\mathbf{u} = \mathbf{HP}$, one has from $\mathbf{u} = \mathcal{P}\mathbf{p}$ and $\mathbf{P} = \mathbf{A}^T\mathbf{p}$ that $\mathcal{P} = \mathbf{HA}^T$. The condition that $\mathcal{P}^2 = \mathcal{P}$ (necessary for a projection operator[6]) yields $\mathbf{A}^T\mathbf{H} = \mathbf{I}_N$. Using this relation and combining Eq. 2.15 with the Laplace transform of equation 2.12 (2.17) gives $\hat{\mathbf{R}}(s) = \mathbf{A}^T\hat{\mathbf{M}}(s)\mathbf{H}$.

$$s\hat{\mathbf{u}}(s) - \mathbf{u}(0) = \hat{\mathbf{M}}(s)\mathbf{u}(s) \tag{2.17}$$

Laplace transforming 2.13, we obtain equation 2.18

$$\hat{\mathbf{R}}(s) = s\mathbf{A}^T\mathbf{K}(s\mathbf{I}_n - \mathbf{K} + \mathbf{HA}^T\mathbf{K})^{-1}\mathbf{H} \tag{2.18}$$

$\mathbf{H}$ must be chosen to ensure that the stationary solution of 2.14 is $\mathbf{P}^{\mathrm{eq}} = \mathbf{A}^T\mathbf{p}^{\mathrm{eq}}$. This choice is not unique, however a sufficient condition is that $\mathbf{P}^{\mathrm{eq}}$ satisfies detailed balance with $\hat{\mathbf{R}}(s)$ for all $s$, i.e.

$$\hat{\mathbf{R}}(s)\mathbf{D}_N = \mathbf{D}_N\hat{\mathbf{R}}^T(s)$$

The choice of equation 2.19 fulfils this requirement for all $s$.

$$\mathbf{H} = \mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1} \tag{2.19}$$

This can be easily checked for the limit $s \to \infty$, where 2.18 evaluates to equation 2.20.

$$\hat{\mathbf{R}}(\infty) = \mathbf{A}^T\mathbf{KH} \tag{2.20}$$

[6]Projecting a system with the same projector a second time should not change the system.

Substituting equation 2.19 gives equation 2.21.

$$\hat{\mathbf{R}}(\infty)\mathbf{D}_N = \mathbf{A}^T\mathbf{K}\mathbf{D}_n\mathbf{A}. \tag{2.21}$$

This is equal to $\mathbf{D}_N\hat{\mathbf{R}}^T(\infty)$ as long as the rate matrix of the original system $\mathbf{K}$ satisfies detailed balance with $\mathbf{p}^{\text{eq}}$, i.e. $\mathbf{K}\mathbf{D}_n = \mathbf{D}_n\mathbf{K}^T$. From now on we will restrict ourselves to choice 2.19, which preserves the detailed balance condition assumed in the original system, thus making the coarse-grained dynamics equilibrium. For the choice 2.19, we obtain that our projection operator must be as in equation 2.22.

$$\mathcal{P} = \mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1}\mathbf{A}^T \tag{2.22}$$

And also $\mathbf{u}_i(t) = [\mathbf{p}_i^{eq}\mathbf{P}_I(t)]/\mathbf{P}_I^{eq}\ \forall\ i \in I$ so that the elements of $\mathbf{u}$ tend to the same limit as the elements of $\mathbf{p}$. Substituted into equation 2.18, this choice for $\mathbf{H}$ gives the clustering relation first obtained in [105].

$$\hat{\mathbf{R}}(s) = s\mathbf{A}^T\mathbf{K}(s\mathbf{I}_n - \mathbf{K} + \mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1}\mathbf{A}^T\mathbf{K})^{-1}\mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1} \tag{2.23}$$

This result is remarkable as it shows how to construct the rate matrix of a low-dimensional dynamics purely in terms of the rate matrix $\mathbf{K}$ of the original high-dimensional dynamics and a choice of clustering $\mathbf{A}$. To obtain a physically intuitive interpretation of this result, equation 2.23 is rearranged to be of the same form as the Laplace transform of a correlation function, as in equation 2.8. This expression can be simplified down using the Woodbury inversion formula (equation 2.24) and identifying $\mathbf{M} = (s\mathbf{I}_n - \mathbf{K})$, $\mathbf{U} = \mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1}$ and $\mathbf{V} = \mathbf{A}^T\mathbf{K}$..

$$(\mathbf{M} + \mathbf{U}\mathbf{V})^{-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{U}(\mathbf{I}_n + \mathbf{V}\mathbf{M}^{-1}\mathbf{U})\mathbf{V}\mathbf{M}^{-1} \tag{2.24}$$

In simplifying down it is useful to notice that $\mathbf{V}\mathbf{M}^{-1}\mathbf{U} = s\mathbf{A}^T\mathbf{M}^{-1}\mathbf{U} - \mathbf{I}_n$. Using these relations, it is straightforward to obtain the simpler version of equation 2.23 given in

2.25.

$$\hat{\mathbf{R}}(s) = s\mathbf{I}_n - (\mathbf{A}^T(s\mathbf{I}_n - \mathbf{K})^{-1}\mathbf{D}_n\mathbf{A}\mathbf{D}_N^{-1})^{-1} \tag{2.25}$$

This can be rearranged in to the form of equation 2.26.

$$\mathbf{A}^T(s\mathbf{I}_n - \mathbf{K})^{-1}\mathbf{D}_n\mathbf{A} = (s\mathbf{I}_N - \hat{\mathbf{R}}(s))^{-1}\mathbf{D}_N \tag{2.26}$$

Since $s^{-1}\mathbf{A}^T\mathbf{D}_n\mathbf{D}_n^T\mathbf{A} = s^{-1}\mathbf{D}_N\mathbf{D}_N^T$ can be subtracted from both sides, we can recognise the Laplace transformed correlation function using equation 2.16.

$$\mathbf{A}^T\left((s\mathbf{I}_n - \mathbf{K})^{-1}\mathbf{D}_n - \frac{1}{s}\mathbf{D}_n\mathbf{D}_n^T\right)\mathbf{A} = \hat{\bar{\mathbf{C}}}^{CG}(s) \tag{2.27}$$

Using equation 2.8, leads to our final result of 2.28 showing that the condition which arises naturally from attempting to reproduce the high dimensional kinetics on a low dimensional space is to equate Laplace transformed correlation functions.

$$\sum_{i\in I}\sum_{j\in J}\int_0^\infty \bar{C}_{ij}(t)e^{-st}dt = \int_0^\infty \bar{C}_{IJ}^{CG}(t)e^{-st}dt, \tag{2.28}$$

### 2.5.1 Markovian Low-dimensional Dynamics

In the previous section the correlation element protocol arose naturally out of enforcing a projected dynamics to preserve detailed balance. To arrive to a Markovian system that preserves some properties of the correlation functions of the original system requires a choice of the Laplace parameter $s$. It can be seen from equation 2.28 that the choice of $s$ effectively dictates which times contribute the most on each side of the equation. Taking a particular value for $s$ will have the effect of using correlation functions at all times in the sum but providing a heavier weight to some than others. Generally however, the Laplace parameter $s$ is a complex number such that $s = a + ib$ with $a$ and $b$ being real numbers. The various choices of $s$ are summarised below.

- $s = 0$: Equally weight correlations at all times. Makes total area under correlation

functions equal.

- $s \to \infty$: Equate correlation functions at $t = 0$.
- $s > 0$: Exponentially decreasing weight for correlation functions with increasing time. Bigger $s$ will more heavily weight towards early correlation functions.
- $s < 0$: Exponentially increasing weight for correlation functions with increasing time. Biased towards ensuring correlation functions are equal at long times.
- $s \in \mathbb{C}$: This choice is possible but less physically intuitive. It may be applicable for systems with variables which periodically correlate and decorrelate.

Of these options, there are two choices of particular physical interest, $s \to 0$ (Hummer-Szabo [105]) and $s \to \infty$ (Local Equilibrium).

In the case where $s$ goes to 0, we obtain the Hummer-Szabo definition which ensures that the area under the correlation functions over all time is equal:

$$\sum_{i \in I} \sum_{j \in J} \int_0^\infty \bar{C}_{ij}(t) dt = \int_0^\infty \bar{C}_{IJ}^{CG}(t) dt. \tag{2.29}$$

At the other end of the spectrum, one can obtain the Local Equilibrium condition by taking the limit of $s$ tending to infinity. In this limit $e^{-st}$ becomes a delta function $\delta_{0t}$ such that only the initial correlation functions are equated.

$$\sum_{i \in I} \sum_{j \in J} \bar{C}_{ij}(0) = \bar{C}_{IJ}^{CG}(0) \tag{2.30}$$

In the following sections we will focus on these two definitions. We are less interested in the cases above which focus on equating the long time correlation functions ($s < 0$) because all of these methods will guarantee that the long time correlation functions will be equal. This is because we made the strict requirement in the previous section that the equilibrium populations in the high and low-dimensional representations must be equal.

On the other hand, we can suggest similar conditions which do not naturally arise from the previous protocol but may still be useful. For example the typical case of MSM construction which equates at some chosen time (the lagtime $\tau$) the correlation functions is not covered under any of these conditions.

To cover this we can first define a more general condition with the time integral of the correlation functions to be equal between two selected times, $\tau_1$ and $\tau_2$, for the coarse-grained and full dimensional dynamics:

$$\sum_{i\in I}\sum_{j\in J}\int_{\tau_1}^{\tau_2}\bar{C}_{ij}(t)dt = \int_{\tau_1}^{\tau_2}\bar{C}_{IJ}^{CG}(t)dt. \tag{2.31}$$

Then as a special case, we can select a specific time $\tau_1 = \tau$ and set $\tau_2 = \tau + \epsilon$, where the limit $\epsilon \to 0$ is taken, for which the typical MSM condition returns:

$$\sum_{i\in I}\sum_{j\in J}\bar{C}_{ij}(\tau) = \bar{C}_{IJ}^{CG}(\tau) \tag{2.32}$$

## 2.6 Relaxation Times as a Variational Parameters

In the previous section we addressed our first question by examining a variety of protocols that can be used for defining the kinetics on a low-dimensional system given a high-dimensional propagator. The second question we are interested in answering is whether the relaxation timescales can be used as variational parameters to assess the quality of the clustering. Ideally one would like to show that every eigenvalue is variational under the clustering projection. This is analytically challenging to show in general so we instead provide quantitative results for the variational nature of i) the second (first non-zero) eigenvalue and ii) the sum over all eigenvalues to show that they are valid variational parameters. After showing these two analytically, we will assume that each individual eigenvalue is variational[7].

It is reasonable to expect that a good coarse-graining will satisfy a variational

[7]While we do not explicitly prove this more the general case, it is empirically observed to be true.

principle in its preserved timescales. A coarse-graining has the effect of smoothing out and removing any timescales/free energy barriers associated with intrastate dynamics and as such the observed barrier crossing should be made faster rather than slower[8].

Firstly we demonstrate that the rate matrix obtained from the coarse-graining is variational in its second eigenvalue in both the HS and LE conditions described previously. The proof offered is an intuitive, element-wise approach.

### 2.6.1 Hummer-Szabo $\tau_2$ Variational Principle

The Hummer-Szabo condition is given by

$$\sum_{i\in I}\sum_{j\in J}\int_0^\infty \bar{C}_{ij}(\tau)d\tau = \int_0^\infty \bar{C}_{IJ}^{CG}(\tau)d\tau \tag{2.33}$$

The correlation functions here can be replaced by the explicit expression in terms of the propagator.

$$\sum_{i\in I}\sum_{j\in J}\int_0^\infty ([e^{\mathbf{K}\tau}]_{ij}p_j^{\mathrm{eq}} - p_i^{\mathrm{eq}}p_j^{\mathrm{eq}})dt = \int_0^\infty ([e^{\mathbf{R}\tau}]_{IJ}P_J^{eq} - P_I^{eq}P_J^{eq})dt. \tag{2.34}$$

Performing a spectral decomposition of the Hummer Szabo condition and using $\mu_n$ and $\mathbf{\Phi}_n^R$ to denote the $n$th eigenvalue and eigenvector respectively in the reduced system and using $\lambda_n$ and $\psi_n^R$ to denote the $n$th eigenvalue and eigenvector respectively in the full system to obtain equation 2.35.

$$\int_0^\infty \sum_{n'=2}^{N} e^{\mu_{n'}t}\Phi_{n'}^R(I)\Phi_{n'}^R(J)dt =$$

$$\sum_{i\in I}\sum_{j\in J}\int_0^\infty \sum_{n'=2}^{n} e^{\lambda_{n'}t}\psi_{n'}^R(i)\psi_{n'}^R(j)dt. \tag{2.35}$$

[8]Imagine a cyclist pedaling uphill on a smooth road vs on a mountain path. Removing the effect of friction will make the uphill crossing faster.

Next, we can explicitly perform the time integrals on both sides of the equation.

$$\sum_{n'=2}^{N} \frac{-1}{\mu_{n'}} \Phi_{n'}^{R}(I)\Phi_{n'}^{R}(J) = \sum_{i\in I}\sum_{j\in J}\sum_{n'=2}^{n} \frac{-1}{\lambda_{n'}} \psi_{n'}^{R}(i)\psi_{n'}^{R}(j). \tag{2.36}$$

Multiplying both sides by $\Phi_2^L(I)\Phi_2^L(J)$ and summing over all macrostates $I$ and $J$ the second eigenvalue can be isolated due to the orthogonality of eigenvectors.

$$\frac{-1}{\mu_2} = \sum_{I,J}\sum_{i\in I}\sum_{j\in J}\sum_{n'=2}^{n} \frac{-1}{\lambda_{n'}} \psi_{n'}^{R}(i)\psi_{n'}^{R}(j)\Phi_2^L(I)\Phi_2^L(J). \tag{2.37}$$

Since $I$ and $J$ (and by extension $i$ and $j$) are indices that run over the same values we can simplify down by grouping terms together to obtain equation 2.38 with the definition $a_{n'} = (\sum_I \sum_{i\in I} \psi_{n'}^R(i)\Phi_2^L(I))^2$.

$$\frac{-1}{\mu_2} = \sum_{n'=2}^{n} \frac{-1}{\lambda_{n'}} (\sum_{I}\sum_{i\in I} \psi_{n'}^{R}(i)\Phi_2^L(I))^2 = \sum_{n'=2}^{n} \frac{-1}{\lambda_{n'}} a_{n'} \tag{2.38}$$

From the orthogonality and normalization of the eigenvectors, it can be shown that $\sum_{n'=2}^{n} a_{n'} = 1$, giving 2.39.

$$\frac{-1}{\mu_2} \leq \frac{-1}{\lambda_2} \sum_{n'=2}^{n} a_{n'} = \frac{-1}{\lambda_2} \tag{2.39}$$

Since the negative inverse of the eigenvalue is the relaxation time, the slowest relaxation time of the dimensionally reduced matrix $\mathbf{R}$ obtained via HS is always less than or equal that of the original system $\mathbf{K}$.

$$\tau_2^R \leq \tau_2^K \tag{2.40}$$

### 2.6.2 Local Equilibrium $\tau_2$ Variational Principle

Next the slowest relaxation time variational principle is proven for the local equilibrium condition. The local equilibrium condition corresponds to enforcing that the number

of transitions occurring at equilibrium is exact at short times.

$$\sum_{i\in I}\sum_{j\in J}\bar{\mathbf{C}}(\tau)_{ij} = \bar{\mathbf{C}}(\tau)_{IJ} \tag{2.41}$$

In the case where $\tau$ is much smaller than the slowest timescale of the rate matrix[9], then equation 2.41 can be simplified to equation 2.42.

$$R_{IJ}P_J = \sum_{i\in I}\sum_{j\in J}K_{ij}p_j. \tag{2.42}$$

From here we can proceed analogously to the proof for Hummer-Szabo and both sides of the equation can be spectrally decomposed.

$$\sum_{n'=2}^{N}\mu_{n'}\Phi^R_{n'}(I)\Phi^R_{n'}(J) = \sum_{i\in I}\sum_{j\in J}\sum_{n'=2}^{n}\lambda_{n'}\psi^R_{n'}(i)\psi^R_{n'}(j) \tag{2.43}$$

Multiplying both sides by $\mathbf{\Phi}^L_2(I)\mathbf{\Phi}^L_2(J)$ and summing over all macrostates $I$ and $J$, the second eigenvalue can be isolated.

$$\mu_2 = \sum_{I,J}\sum_{i\in I}\sum_{j\in J}\sum_{n'=2}^{n}\lambda_{n'}\psi^R_{n'}(i)\psi^R_{n'}(j)\Phi^L_2(I)\Phi^L_2(J) \tag{2.44}$$

Exactly as before, it follows that $\mu_2 < \lambda_2$ and so (since the eigenvalues are both negative) it follows that $\frac{-1}{\mu_2} < \frac{-1}{\lambda_2}$. Hence the local equilibrium condition satisfies the variational principle for its second eigenvalue.

$$\tau^R_2 \leq \tau^K_2 \tag{2.45}$$

### 2.6.3 Interlude: Kemeny Constant

In this next section we want to consider the sum over all relaxation times as a variational parameter. This seems a reasonable parameter to consider since this would

[9]To make the step from correlation matrix to rate matrix we need $\tau$ to be small enough that it becomes reasonable to approximate $e^{\mathbf{K}\tau}$ by Taylor expanding to first order as $\mathbf{I} + \mathbf{K}\tau$.

involve aiming to preserve all of the timescales present in the coarse-grained projection. However before we examine this, we make a slight diversion since the sum of all relaxation times has some interesting properties relating to correlation functions and mean first passage times. This summation of all timescales in a Markov chain is known as the Kemeny constant, $\zeta$. It was first described by Kemeny and Snell in their textbook Finite Markov Chains [106].

$$\zeta = \sum_j t'_{ji} p_j^{eq} = \sum_{n=2}^{N} \tau_n \tag{2.46}$$

Here $t'_{ji}$ is the mean first passage time from state $i$ to state $j$. This is called the Kemeny constant due to its lack of dependence on the starting state index $i$.

To obtain the above result of the Kemeny constant, we begin by considering how to write the continuous-time mean first passage time between a pair of states $i$ and $j$ in a Markov system described by a rate matrix $\mathbf{K}$. The transition probability can be written as $e^{\mathbf{K}\tau}$. We will take a timestep unit of $\tau$ and consider the limit to zero at the end.

$$t_{ji} = [e^{\mathbf{K}\tau}]_{ji}\tau + \sum_{k\neq j}[e^{\mathbf{K}\tau}]_{ki}(t_{jk}+\tau) = \tau + \sum_{k\neq j}[e^{\mathbf{K}\tau}]_{ki}t_{jk} \tag{2.47}$$

We can rewrite equation 2.47 in a more convenient form as in equation 2.48 (defining $t'_{ji} = \frac{t_{ji}}{\tau}$ and the Kronecker delta $\delta_{ki}$).

$$\sum_k (\delta_{ki} - [e^{\mathbf{K}\tau}]_{ki})t'_{jk} = 1 - [e^{\mathbf{K}\tau}]_{ji}t'_{jj} \tag{2.48}$$

This leads to the more convenient matrix form of equation 2.49 where we have defined $\mathbf{t'}_j^T = (t'_{j1}, \ldots, t'_{jN})$ as the row vector with the MFPTs to $j$ as components.

$$\mathbf{t}'_j(\mathbf{I} - [e^{\mathbf{K}\tau}]) = (1 - [e^{\mathbf{K}\tau}]_{j1}t'_{jj}, ..., 1 - [e^{\mathbf{K}\tau}]_{jN}t'_{jj}) \tag{2.49}$$

We can then write the mean first passage time as a linear combination of the left eigenvectors of $T$[10]. Using ${\mathbf{t}'_j}^T = \sum_\ell a_{j\ell}\phi^{(\ell)}$, substituting in to equation 2.49 and considering component $r$ produces equation 2.50 (where we have used that $\lambda_\ell$ is the eigenvalue associated with the left eigenvector $\phi_r^{(\ell)}$).

$$\sum_\ell a_{j\ell}(1 - e^{\lambda_\ell \tau})\phi_r^{(\ell)} = 1 - [e^{\mathbf{K}\tau}]_{jr} t'_{jj} \tag{2.50}$$

By multiplying by $\phi_r^{(s)}$, the $r$-th component of the $s$-th right eigenvector of $\mathbf{T}$ and summing over $r$, the orthonormality (and normalization) of the eigenvectors allows us to obtain the mean first passage times in terms of the eigenvalues and vectors of the transition matrix (and a set of constants $a_{j\ell}$ which remain to be determined).

$$\sum_{\ell>1} a_{j\ell}(1 - e^{\lambda_\ell \tau})\delta_{\ell s} = \delta_{s1} - e^{\lambda_s \tau}\psi_j^{(s)} t'_{jj} \tag{2.51}$$

Taking $s = 1$, one finds that in the discrete time formalism for mean first passage times, the diagonal elements (or recurrence time) $t'_{jj}$ are given by the inverse of the equilibrium probablities [107].

$$t'_{jj} = \frac{1}{p_j^{eq}} \tag{2.52}$$

Inserting this in to equation 2.51 provides an expression for the $a$ values in terms of just the eigenvalues/vectors.

$$a_{js} = -\frac{1}{p_j^{\mathrm{eq}}} \frac{e^{\lambda_s \tau}}{1 - e^{\lambda_s \tau}} \psi_j^{(s)} \tag{2.53}$$

[10]This assumes that $T$ has a complete set of orthonormal vectors which is only guaranteed if $T$ satisfies detailed balance.

By substituting this back in to the expression for the mean first passage time in terms of the basis vectors, one can obtain equation 2.54.

$$t'_{jk} = a_{j1} - \frac{1}{p_j^{\text{eq}}} \sum_{\ell>1} \frac{e^{\lambda_\ell \tau}}{1 - e^{\lambda_\ell \tau}} \psi_j^{(\ell)} \phi_k^{(\ell)} \tag{2.54}$$

The $a_{j1}$ term can be identified by setting $j = k$ and finally we can arrive at equation 2.55 for the MFPT in terms of the eigenvalues and eigenvectors.

$$t'_{ji} = \frac{1}{p_j^{\text{eq}}} \left[ \tau + \sum_{\ell>1} \frac{e^{\lambda_\ell \tau}}{1 - e^{\lambda_\ell \tau}} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}) \right] \tag{2.55}$$

Taking the limit of $\tau$ to zero simplifies equation 2.55.

$$t'_{ji} = \frac{1}{p_j^{\text{eq}}} \left[ \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}) \right] \tag{2.56}$$

And so the $t_{jj}$ element is zero since in continuous time there is no time in which to leave and return to the state,

$$t_{jj} = 0 \tag{2.57}$$

We now have an equation for the mean first passage time in terms of the eigenvalues and eigenvectors of the Markovian rate matrix. Now we can observe that if we multiply equation 2.56 on both sides by $p_j^{eq}$ and sum over all $j$ that from the orthogonality of eigenvectors we arrive at equation 2.58.

$$\sum_j t'_{ji} p_j^{eq} = \sum_{l>1} \tau_l \tag{2.58}$$

Interestingly we have an index $i$ on the left hand side which does not appear on the right, so this expression is independent of the particular choice of $i$. It is for this index independence that the sum of timescales is called the Kemeny constant. Following the

original derivation by Kemeny, this quantity has generated a huge amount of academic interest in attempts to offer an interpretation for the invariance under the choice of starting state [108–111].

Next we offer a physical interpretation of Kemeny constant in terms of correlation functions. A similar interpretation was proposed by Bini [110] in terms of the number of lost transitions to $j$ that occur as a result of having started in $i$ rather than $j$, our interpretation introduces correlation functions to formalise the interpretation of Kemenys constant as describing the invariance of the decay to equilibrium. We can start with our continuous time expression for the mean first passage time.

$$t'_{ji} = \frac{1}{p_j^{\text{eq}}} \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}) \tag{2.59}$$

Adding and subtracting $\psi_j^{(1)}$ and bearing in mind that $\phi_i^{(1)} = 1 \; \forall \; i$, we can reformulate the vector products in terms of matrix elements

$$\begin{aligned} t'_{ji} &= \frac{1}{p_j^{\text{eq}}} \left[ -\psi_j^{(1)} \phi_j^{(1)} + \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} \phi_j^{(\ell)} \right. \\ &\quad \left. + \psi_j^{(1)} \phi_i^{(1)} - \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} \phi_i^{(\ell)} \right] \end{aligned} \tag{2.60}$$

$$= \frac{1}{p_j^{\text{eq}}} \left[ (\mathbf{P}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1}_{jj} - (\mathbf{P}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1}_{ji} \right] \tag{2.61}$$

where we have used $\mathbf{P}^{\text{eq}} = \psi^{(1)}$ and $\mathbf{1}_n^T = \phi^{(1)}$.

Now we can write the mean first passage time in terms of correlation functions.

$$p_j^{\text{eq}} t'_{ji} = \frac{\int_0^\infty C_{jj}^{\text{eq}}(\tau) d\tau}{p_j^{\text{eq}}} - \frac{\int_0^\infty C_{ji}^{\text{eq}}(\tau) d\tau}{p_i^{\text{eq}}} \tag{2.62}$$

And so the Kemeny constant can be expressed in terms of the time-integrated

correlation functions.

$$\sum_j p_j^{eq} t'_{ji} = \sum_j \left[ \frac{\int_0^\infty C_{jj}^{\mathrm{eq}}(\tau) d\tau}{p_j^{\mathrm{eq}}} - \frac{\int_0^\infty C_{ji}^{\mathrm{eq}}(\tau) d\tau}{p_i^{\mathrm{eq}}} \right] \tag{2.63}$$

Expressing the correlation function in terms of the rate matrix exponential we can arrive at equation 2.64 and see that the Kemeny constant can be interpreted as the additional time required for the system to converge to equilibrium as a result of starting in $i$ rather than $j$. Since we are summing over every $j$, every possible dynamics in the system will be considered (hence the sum of relaxation times interpretation) and the choice of $i$ will not matter.

$$\sum_j p_j^{eq} t'_{ji} = \sum_j \left[ \int_0^\infty [e^{\mathbf{K}\tau}]_{jj} - p_j^{eq} d\tau - \int_0^\infty [e^{\mathbf{K}\tau}]_{ji} - p_j^{eq} d\tau \right] \tag{2.64}$$

### 2.6.4 Hummer-Szabo Kemeny Variational Principle

We would like to show that when we use the HS protocol to perform a clustering, the Kemeny constant of the clustered system is always less than that of the original unclustered system[11]. This would enable us to reliably use the Kemeny constant as a variational parameter for clustering.

The Kemeny constant was used in a recent study examining community detection in general networks [112]. This study was reliant on the argument that highly connected networks will tend to have low values of Kemeny (short MFPTs) while poorly connected networks will have high Kemeny values (high MFPTs). Here we formalise this concept by showing the variational behaviour of the Kemeny constant under HS clustering and also offering an analytically exact interpretation of its optimization. We begin by considering the MFPT of the clustered system in terms of the correlation functions as

[11]To avoid confusion later, it is important to remember that the 'constant' part of the Kemeny constant is that its MFPT representation is independent of a microstate index. However the Kemeny constant does not remain constant after Hummer-Szabo clustering.

derived previously in equation 2.62.

$$P_J^{eq} t_{JI} = \int_0^\infty \frac{\bar{C}_{JJ}(\tau)}{p_J^{eq}} d\tau - \int_0^\infty \frac{\bar{C}_{JI}(\tau)}{p_I^{eq}} d\tau \tag{2.65}$$

The HS protocol provides us with a means to connect the cluster correlation functions with the original.

$$P_J^{eq} t_{JI} = \frac{1}{p_J^{eq}} \sum_{j \in J, j' \in J} \int_0^\infty \bar{C}_{jj'}(\tau) d\tau - \frac{1}{p_J^{eq}} \sum_{j \in J, i \in I} \int_0^\infty \bar{C}_{ji}(\tau) d\tau \tag{2.66}$$

Both terms on the right can be substituted for using equation 2.63.

$$P_J^{eq} t_{JI} = \frac{1}{p_J^{eq}} \sum_{j \in J, j' \in J} p_{j'}^{eq} \Big[ \int_0^\infty \frac{\bar{C}_{jj}(\tau)}{p_j^{eq}} d\tau - p_j^{eq} t_{jj'} \Big]$$

$$- \frac{1}{p_I^{eq}} \sum_{j \in J, i \in I} p_i^{eq} \Big[ \int_0^\infty \frac{\bar{C}_{jj}(\tau)}{p_j^{eq}} d\tau - p_j^{eq} t_{ji} \Big] \tag{2.67}$$

Simplifying equation 2.67 results in equation 2.68.

$$P_J^{eq} t_{JI} = \frac{1}{p_I^{eq}} \sum_{j \in J, i \in I} p_i^{eq} p_j^{eq} t_{ji} - \frac{1}{p_J^{eq}} \sum_{j \in J, j' \in J} p_{j'}^{eq} p_j^{eq} t_{jj'} \tag{2.68}$$

Finally summing over all macrostates $J$ produces a bound for the Kemeny constant given by equation 2.69.

$$\zeta^{HS} = \zeta^{orig} - \sum_J p_J^{eq} \sum_{j \in J, j' \in J} \frac{p_{j'}^{eq}}{p_J^{eq}} \frac{p_j^{eq}}{p_J^{eq}} t_{jj'} \tag{2.69}$$

This bound represents the expectation value of mean first passage time if two states are drawn from within the same cluster with equilibrium probability. As such, this becomes smaller as the states become increasingly metastable (interconversion is fast) and vanishes to zero only when each macrostate consists of one microstate (i.e. no clustering performed).

### 2.6.5 Choice of Clustering Parameter

Now that we have demonstrated the slowest relaxation time is variational for HS and LE and the Kemeny constant is variational for HS, we proceed to assume that the empirical observation of variational eigenvalues holds true in general. Given this assumption, what variational parameters should we choose? In principle any linear combination of the relaxation times would be feasible (equation 2.70 where $c_n \in \mathbf{R}$).

$$\sum_{n=2}^{N} c_n \tau_n \tag{2.70}$$

Given this endless combination of possibilities, what should we choose to investigate? The starting goal of this project was to develop a method which would be automatic and would (as much as possible) remove the need for complicated or non-intuitive parameter choosing. To this end, we restrict ourselves to $c_n$ equal to zero or one. In particular we will examine using each individual relaxation time $\tau_2$ up to $\tau_N$ and the truncated summation $\sum_{n=2}^{n'} \tau_n$ (for some $n' \leq N$).

These two choices seem reasonable, the single relaxation time optimization should favor preserving the associated kinetic process while the truncated summation will favor preserving multiple kinetic processes in accordance with their relative size (so fast processes will have little influence on the summation optimization). In the results section of this chapter we will examine, for a number of simple test cases, the effect of using the Hummer-Szabo clustering approach to variationally optimize the above parameters.

## 2.7 Results

With the variational proofs of the previous section, we examine qualitatively on a number of illustrative examples, the effect of implementing our suggested procedure. We use the relaxation timescale based variational parameters in conjunction with the Hummer-Szabo clustering to attempt to identify metastable and transition states and

compare our results to the commonly used PCCA+ algorithm described earlier in the chapter. We will examine three artificially produced potential energy landscapes.

- A smooth potential with multiple wells of varying depth.
- A noisy version of the aforementioned multiwell potential.
- A two-dimensional potential energy surface with multiple wells.

To construct our toy model potentials, we defined our potential of interest $V(x)$ (or $V(x, y)$) and discretized the $x$ coordinate in to sufficiently fine states such that the difference in potential between neighbouring states is small. From this discretized potential we then defined our $\mathbf{K}$ matrix using an Arrhenius rate model as in equation 2.71. $A$ is the Arrhenius factor, it has units of inverse time and controls the timescale of the rate matrix. For the purposes of our study here, this value is arbitrary as it controls the magnitude of the relaxation timescales but not their relative size. $k_B$ and $T$ are the Boltzmann constant and temperature respectively.

$$K_{ji} = Ae^{-\frac{V_j - V_i}{k_B T}} \tag{2.71}$$

With this $\mathbf{K}$ matrix we can then implement our iterative procedure, searching through each possible clustering with HS and finding that which optimizes our chosen timescale parameter.

### 2.7.1 Smooth Multiwell Potential

The first potential we examine is a smooth potential with four wells of varying depth. We examine clustering in to four and five states. For the four states as shown in figure 2.3 we compare using i)$\tau_2$, ii)$\tau_3$, iii)$\tau_4$ and iv) PCCA+. We observe in figure 2.3 that all the parameter choices identify the four metastable states in the system. By extension, since the individual timescales find the same clustering, the truncated summations ($\sum_{n=2}^{3} \tau_n$ and $\sum_{n=2}^{4} \tau_n$) will also find the four metastable states.

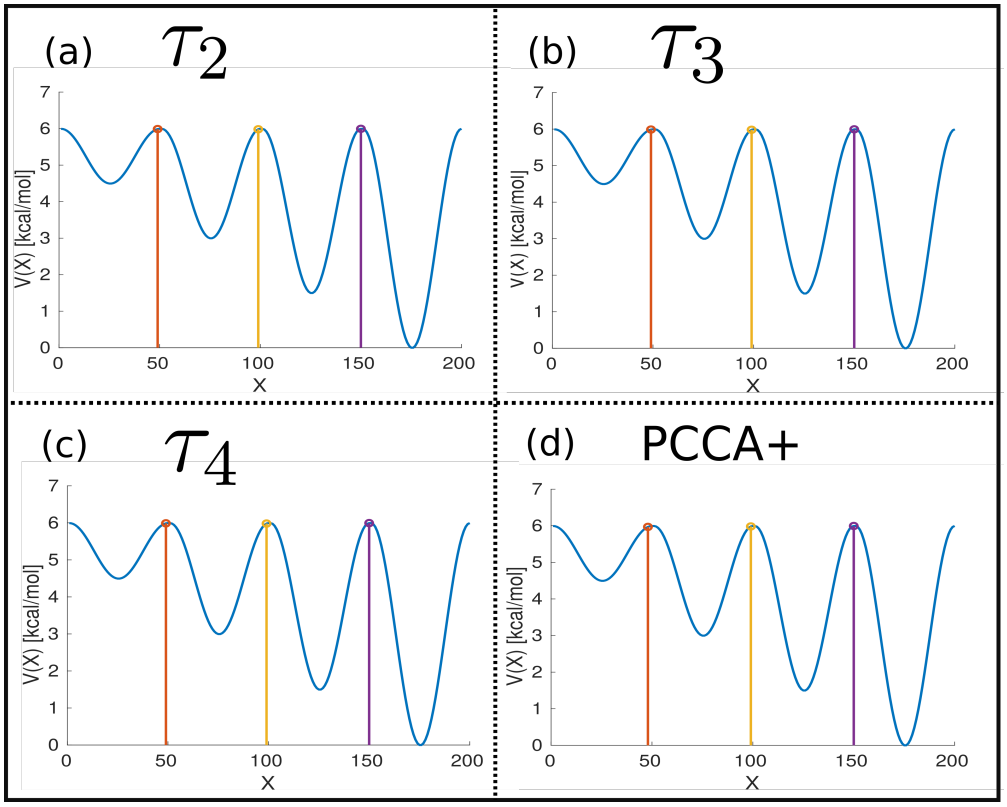


Figure 2.3: Eigenvalue clustering vs PCCA+ for smooth potential four state clustering

Following from this we examine next the first potential again but now with a five state clustering as seen in figure 2.4. Interestingly, every timescale based clustering now identifies a transition state which connects the regions whose dynamics are governed by the smallest $\tau_n$ used. For example, the $\tau_2$ clustering again finds a transition state centered upon the highest free energy boundary, the $\tau_3$ (and $\sum_{n=2}^{3} \tau_n$) finds a transition state on the second largest boundary and similarly for $\tau_4$ (and $\sum_{n=2}^{4} \tau_n$). The PCCA+ in contrast attempts to split up the least stable state in to two pieces. Meanwhile, the Kemeny constant ($\sum_{n=2}^{5} \tau_n$) (not included in figure 2.4) identifies an identical clustering to PCCA+.

It is perhaps unsurprising that each timescale favors placing a transition state in the

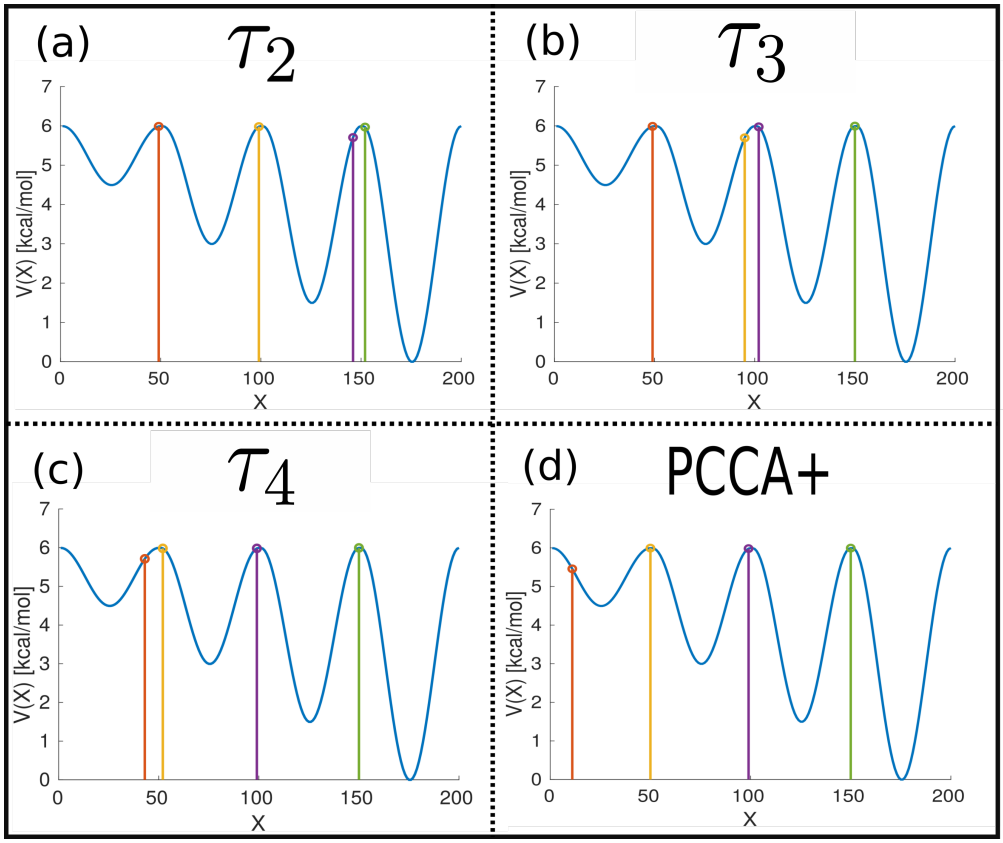


Figure 2.4: Eigenvalue clustering vs PCCA+ for smooth potential five state clustering

region which describes its dynamics. What is more unexpected is that for the truncated summation of timescales, it is the smaller timescale which dictates the transition state region. It appears that, for example in using $\sum_{n=2}^{3} \tau_n$, placing the bottleneck on the second highest barrier is more beneficial to $\tau_2$ than placing on the highest barrier would be to $\tau_3$. This will be dependent on the relative magnitude of the relaxation times. Clearly as $\tau_3$ becomes very small then it will no longer be favorable to place a transition state which limits according to the $\tau_3$ dynamics.

These timescale based variational parameters appear promising for the automatic identification of both metastable and transition state regions. We have also seen that for the simple 1-D case, the truncated summations produce the same clustering as

that of their smallest timescale. Meanwhile the Kemeny constant finds an identical clustering to the PCCA+. For the rest of this results section we examine two more test cases to deepen our understanding and test our method further. First we see how robust these results are to noise in the potential by adding random Gaussian noise. Secondly we examine how these methods perform on higher-dimensional potentials.

### 2.7.2 Noisy Multiwell Potential

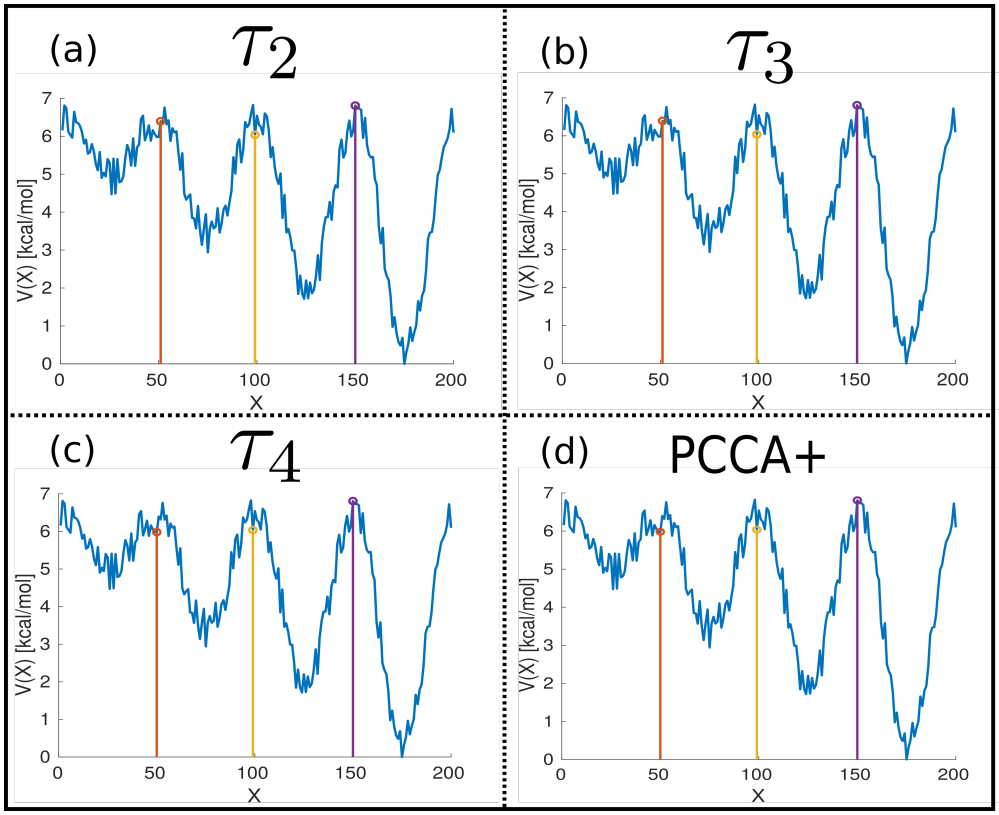


Figure 2.5: Eigenvalue clustering vs PCCA+ for noisy potential four state clustering

In the previous section, we considered a very idealised smooth potential. In reality these free energies are likely to be noisy due to insufficient sampling either along the coordinate of interest or on some faster degree of freedom. To try to account for

these more realistic situations we add some random Gaussian noise to the potential considered previously and examine the same clusterings. As before we cluster in to four and five states using the same selection of timescale based variational parameters and PCCA+. For the four state clustering presented in figure 2.5, effectively the same results are observed with all methods identifying the four metastable states. The truncated sum (again omitted for brevity) produces identical results.

More interesting is to examine the five state clustering results. The timescale based clusterings reproduce their bottleneck states as before while now PCCA+ has moved from splitting the least metastable state in two to identifying what appears to be the next most metastable state in the system (and the Kemeny constant (not shown) produces the same result). This demonstrates that in more realistic noisy cases, PCCA+ and the Kemeny constant are geared towards finding metastable states and as such will find metastable states that are perhaps not due to the underlying potential but rather are due to the noise. Importantly, the results of the single timescale variational clustering appear to be robust to the addition of noise to the system.

### 2.7.3 Two-dimensional Potential Energy Surface

In this final test case we apply the proposed clustering algorithm to a two-dimensional potential energy surface with three metastable wells (as shown in figure 2.7). The three well potential considered is given in equation 2.72 (in units of kcal/mol) with $x$ and $y$ both taking values in the range $[-0.6\pi, 0.6\pi]$. $c$ is a constant value such that the function takes a minimum value of 0 on this domain of values.

$$v(x,y) = -8(e^{-2(x-1.3)^2-2(y-1.3)^2} + e^{-2(x-1.2)^2-2(y+1.2)^2} + e^{-2(x+1.3)^2-2(y+0.9)^2}) + c \tag{2.72}$$

This is an interesting test case as for higher dimensional models the number of

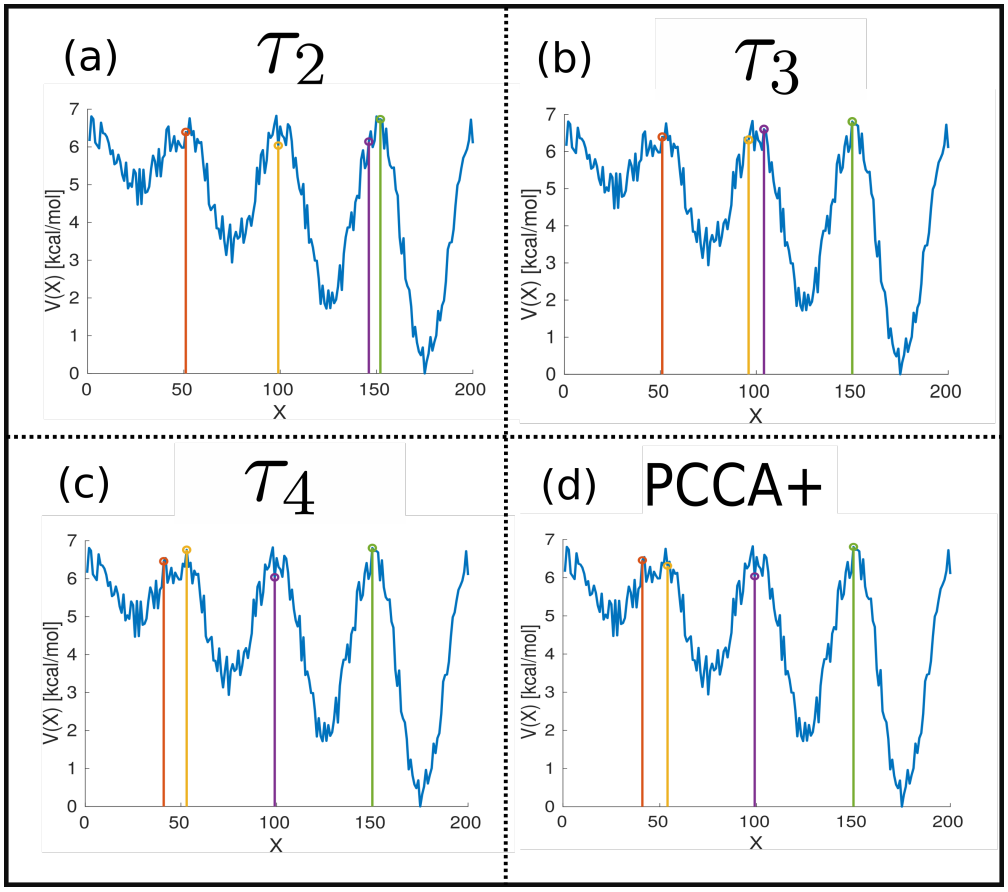


Figure 2.6: Eigenvalue clustering vs PCCA+ for noisy potential five state clustering

possible permutations grows rapidly. The kinetic pathways can also become linked together, as opposed to the 1-D case where the various transition states were completely separate. It is in this case where it becomes interesting to examine both the single relaxation times and the truncated summations since they produce distinct clusterings and the difference in those clusterings serves to deepen our understanding of the forces at work in this parameter optimization. Given that our potential contains three metastable states we will consider three, four and five state clustering.

There is a wide combination of cluster numbers and parameter choices which we could present here, but not all of these choices lead to sensible or useful results. Here

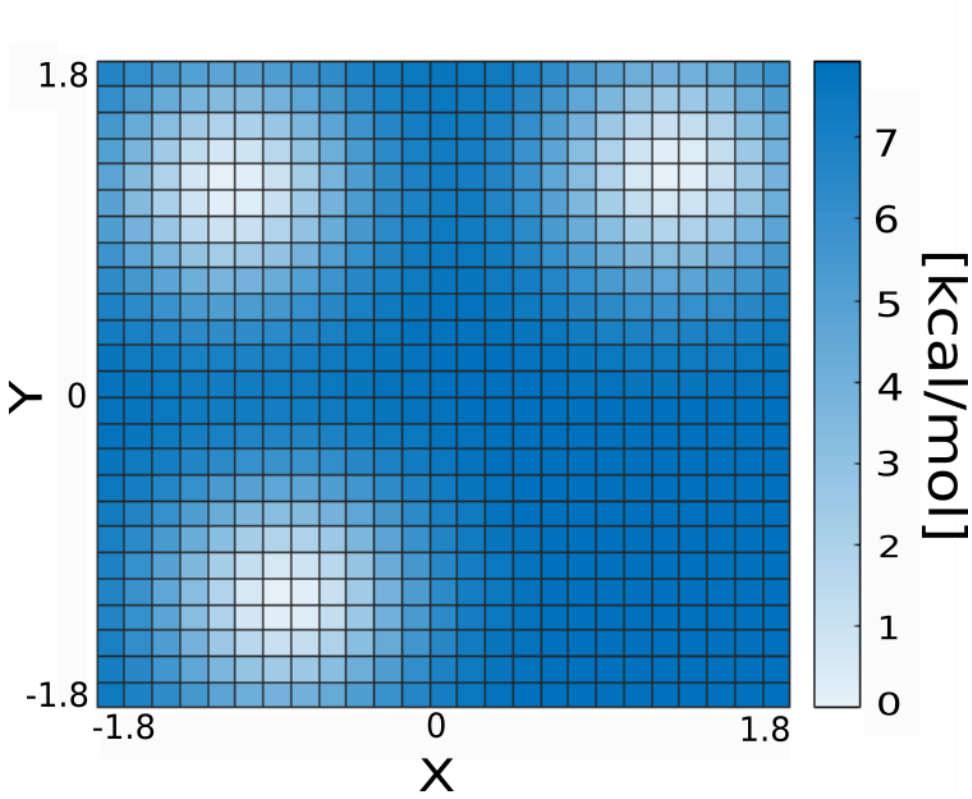


Figure 2.7: 2D potential energy surface

in figure 2.9 we present a selection of figures which illustrate our suggested protocol for applying this eigenvalue procedure to multi-dimensional models. First we cluster using the Kemeny constant to identify the maximally metastable clustering. If we cluster with Kemeny while seeking for more clusters than are present in the system then we will find that one of the returned clusters is several orders of magnitude smaller than the others. This allows us to assess how many metastable states are present. Once it has been determined that there are three metastable states we can extend to identifying transition states by searching for four or five clusters with kinetic parameters other than Kemeny. But which parameters should be used to find these transition regions?

We find that it is not advisable to cluster using single eigenvalues (other than $\tau_2$).

For example clustering in to four states using $\tau_3$ produces figure 2.8. This groups states in a bad way but given our parameter choice we can understand why it has grouped in the way it has. Since the clustering cares only about maximising $\tau_3$ and has no interest in $\tau_2$, it extends the metastable state for the most stable cluster in to being a transition state region to make the $\tau_3$ process slower. This non-sensible distortion was not possible in the one-dimensional case since there was no way for the most stable state to deform and contribute to the $\tau_3$ process.

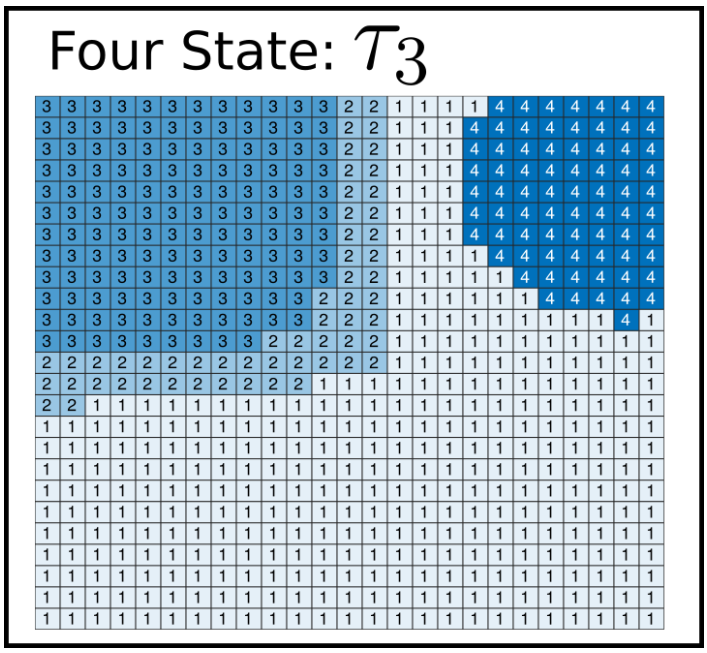


Figure 2.8: 2D clustering in four states with $\tau_3$

If we look instead at figure 2.9 (b) and (c), here we cluster in to four states using $\tau_2$ and $\sum_{n=2}^{3} \tau_n$ respectively. Again the resultant clustering can be interpreted in our choice of parameter. In maximising $\tau_2$, we see that a transition state region is placed around the most stable state. However the optimization assigns some of the high potential region in the lower right corner to one of the metastable clusters. By instead using $\sum_{n=2}^{3} \tau_n$, the system now has an incentive to care about the nature of the boundary between states 3 and 4 and reassigns this high potential region to belong to

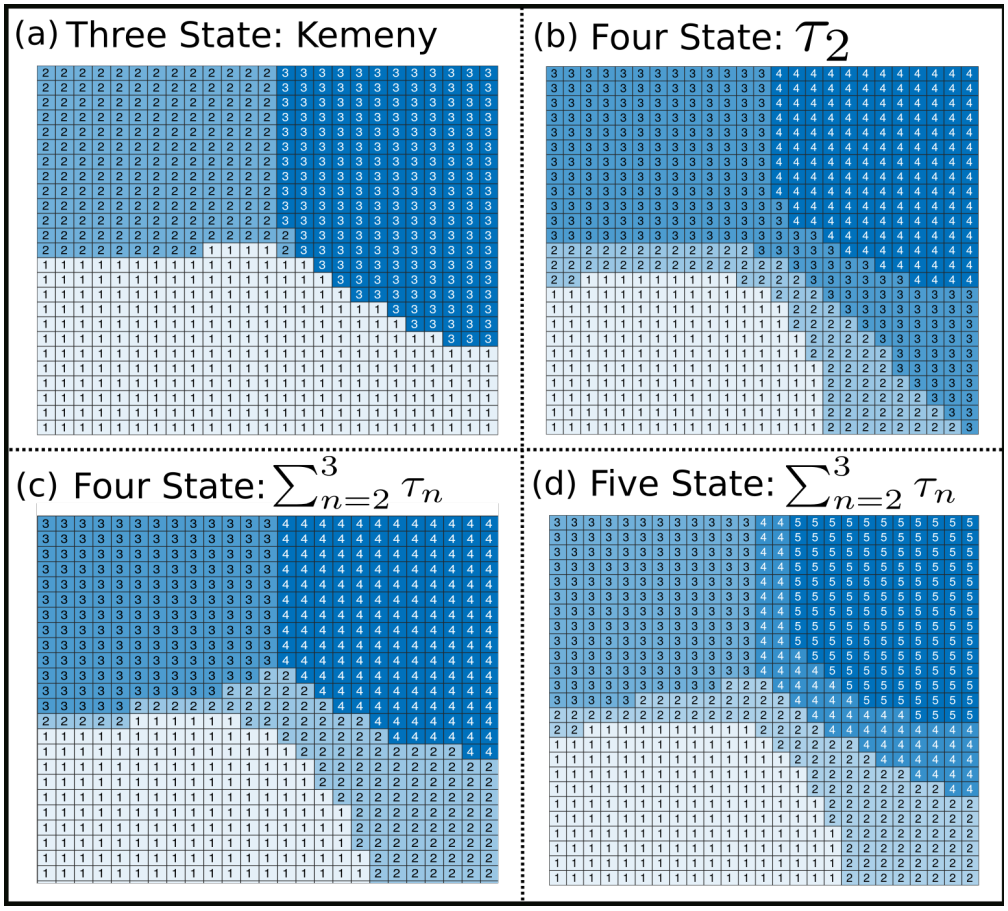


Figure 2.9: 2D potential energy clustering

the transition state such that transitions from 3 to 4 only happen through the pathway connecting them.

Finally if we choose to do a five state optimization using $\sum_{n=2}^{3} \tau_n$ (figure 2.9 (d)) as our variational parameter (since these are the two timescales governing metastable transitons) then we can identify our three metastable states as well as two transition state regions describing the dynamics.

## 2.8 Conclusions

We have examined the impact of performing a timescale based clustering with the Hummer-Szabo method. Given our observations, we make the following suggested protocol for implementing timescale based clustering.

1. Perform coarse-graining using Kemenys constant for increasing numbers of clusters until a state is returned which is much smaller in magnitude than the other clusters. This provides the number of metastable states $N_{stable}$.

2. If the potential is 1-D, then use the kinetic parameters $\tau_2$, ..., $\tau_{N_{stable}-1}$ to find the transition states linking the key metastable regions.

3. If however the potential is multidimensional, cluster in to $N_{stable} + 1$ up to $2N_{stable} - 1$ clusters using $\sum_{n=2}^{N_{stable}-1} \tau_n$ as a parameter to ensure that all the timescales are accounted for.

For 1-D systems we found that the approach is able to automatically and robustly identify both metastable states and transition state regions. The truncated sum of timescales up to $\tau_n$ as well as the individual timescales will find all metastable regions as well as a transition state along the pathway connecting the states whose dynamics is described by timescale $\tau_n$. In contrast, the PCCA+ algorithm, does not explicitly pick up any transition state although in a noisy potential it will pick up increasingly small metastable states. It may happen though that (as in our presented example) these small metastable states are just artefacts of the distorted potential energy obtained from the insufficient sampling of microstates. In the case of the 2-D systems, we saw that due to the merging of kinetic pathways, we needed to be more careful regarding our choice of kinetic parameters.

The important take away from this chapter is that we have at this point obtained promising initial results that relaxation timescales can be used to automatically identify metastable and transition states. We can identify multiple transition state regions

from which to reinitialise simulations even when the potential energy is noisy. This is especially promising because it would be unhelpful for improving simulations if we were to require a well-converged potential in order to identify the transition states.

With this new method for identifying clusters in kinetic models, in the next two chapters we will first try and develop an analytic framework for thinking about and interpreting the results of this chapter and secondly we will develop an algorithmic approach for efficiently implementing this procedure on general networks.

*"One of the principal objects of theoretical research in any department of knowledge is to find the point of view from which the subject appears in its greatest simplicity."*

J. Willard Gibbs

# 3

# Mean First Passage Time Analysis

## 3.1 Introduction

We now know that when implementing either the HS or LE clustering protocol that the slowest relaxation time will be variational (as will be the Kemeny constant). The next question is to ask precisely what choice of microstate clustering is obtained when using this procedure. In this chapter we will investigate the clustering which optimises the slowest relaxation time in some particular cases since we already have an intuitive interpretation for the Kemeny constant.

We will go through the following steps:

1. Assume our system has an underlying dynamics which is well described by a 1-D Smoluchowski equation.

2. Write the slowest relaxation time in terms of time-integrated correlation functions.

3. Use a useful result of Perico and Szabo [113] to write the time-integrated correlation function as an integral over space of the potential energy.

4. Rewrite this spatial integral in terms of mean first passage times to an absorbing boundary.

5. Differentiate the mean first passage times with respect to the position of the absorbing boundary to find a condition which maximises the slowest relaxation time.

We are going to do this for two analytically tractable cases, i) a two state clustering on an arbitrary potential and ii) a three state clustering on a symmetric potential. We postulate that this framework for interpreting relaxation times as mean first passage times holds great promise for developing intuitive explanations for relaxation timescale based clustering and could potentially be extended to better understand the results of higher dimensional clustering.

## 3.2 Theory

### 3.2.1 Smoluchowski Equation

The first step is to make the assumption that the dynamical system which is being investigated is well described by a single variable Smochulowski equation. This is known to be a valid assumption for many systems of biological interest [114–116]. The

objective of performing this analysis is to gain interpretable insight, so a simple 1-D model is favoured. Although the results presented here are one dimensional it is likely that the equations can be generalized to multidimensional dynamics. The 1-D Smoluchowski equation along a coordinate $x$ is written as a Fokker-Planck evolution of conditional probabilities as in equation 3.1.

$$j\left(x,t|x_0,t_0\right) = De^{-\beta V(x)}\nabla e^{\beta V(x)}p\left(x,t|x_0,t_0\right) \tag{3.1}$$

Here $j$ is the probability flux between states, $V$ is the potential which governs the dynamics and $D$ is the diffusion constant. $\beta$ is equal to $1/k_BT$ where $k_B$ is the Boltzmann constant and $T$ is the temperature.

For the analysis/interpretation framework which we are going to layout in this chapter, we will need to have expressions for the mean first passage times between regions/to hit absorbing boundaries for the Smoluchowski equation. For the sake of brevity, we provide these results without derivation as they have been derived and applied in many existing works in the context of reaction rate theory [117]. The mean first passage time from a starting point $x_0$ to an absorbing boundary at $a$ (given a reflecting boundary at $b$) is given by equation 3.2.

$$\left\langle t\left(x_0\right)\right\rangle = \int_a^{x_0}\frac{e^{\beta V(\xi)}}{D(\xi)}d\xi\int_\xi^b e^{-\beta V(\eta)}d\eta \tag{3.2}$$

If the starting position is not a single position but a region of space ($x_0 = [a,b]$) then the mean first passage time from the region to the absorbing boundary is given instead by equation 3.3.

$$\left\langle t\left(x_0\right)\right\rangle = \int_a^{x_0}\frac{e^{\beta V(\xi)}}{D(\xi)}d\xi\frac{\left[\int_\xi^b e^{-\beta V(\eta)}d\eta\right]^2}{\int_a^b e^{-\beta V(\eta')}d\eta'} \tag{3.3}$$

### 3.2.2 Relaxation Times and Correlation Functions

As we saw from Chapter 2 with our examination of the Kemeny constant, there is a close relationship between time-integrated correlation functions and relaxation timescales. This is intuitively reasonable as both quantities have the interpretation of describing a systems speed of convergence to equilibrium.

Since the Kemeny constant has an immediate link to MFPTs and we already have an interpretation in terms of metastability maximisation, we will not consider it here but will focus on trying to describe the clusterings obtained via using the slowest relaxation time, $\tau_2$, as a variational parameter as this leads to the less immediately intuitive transition states.

#### Two State Clustering

For a two state clustering, from the fluctuation-dissipation theorem it can be shown that the relaxation time is related to the time correlation function of the $\theta(x)$ function defined in section 2.4 [99] where $\delta\theta_1(x) = \theta_1(x) - \langle\theta_1\rangle$ and $\langle\theta_1\rangle = \frac{\int_{-\infty}^{a} e^{-\beta V(x)}dx}{\int_{-\infty}^{\infty} e^{-\beta V(x)}dx}$.

$$\int_0^\infty \bar{C}_{11}(t)dt = \int_0^\infty \frac{\langle\delta\theta_1(0)\delta\theta_1(t)\rangle}{\langle\delta\theta_1(0)^2\rangle}dt \tag{3.4}$$

$$\int_0^\infty \frac{\langle\delta\theta_1(0)\delta\theta_1(t)\rangle}{\langle\delta\theta_1(0)^2\rangle}dt = \int_0^\infty \frac{[e^{\mathbf{K}t}]_{11}P_1^{eq} - P_1^{eq}P_1^{eq}}{P_1^{eq}(1-P_1^{eq})} \tag{3.5}$$

We can now spectrally decompose the rate matrix and simplify the integrand.

$$= \int_0^\infty \frac{\sum_{n=1}^{2} e^{\lambda_n t}\psi_n^L(1)\psi_n^R(1) - P_1^{eq}}{(1-P_1^{eq})}dt \tag{3.6}$$

$$= \int_0^\infty \frac{e^{\lambda_2 t}\psi_2^L(1)\psi_2^R(1)}{P_2^{eq}}dt \tag{3.7}$$

For a two state system, some simple relations for the eigenvectors (normalization and orthogonality) make it straightforward to show that the term multiplying the

exponential is equal to 1 and so one is left with just the integral.

$$= \int_0^\infty e^{\lambda_2 t} dt = \frac{-1}{\lambda_2} = \tau_2 \tag{3.8}$$

So we can write the relaxation time as a time-integral of correlation functions.

$$\tau_2 = \int_0^\infty \bar{C}_{11}(t) dt = \int_0^\infty \frac{\langle \delta\theta_1(0)\delta\theta_1(t)\rangle}{\langle \delta\theta_1(0)^2\rangle} dt \tag{3.9}$$

**Three State Symmetric Clustering**

We can also write the slowest relaxation time of a three state clustering in terms of time-integrated correlation functions if we assume the underlying potential is symmetric. By assuming the potential is symmetric, we reduce the two boundary optimization to a single boundary optimization. This reduces the number of free parameters in our correlation matrix.

We begin by considering our 1-D symmetric potential from $-\infty$ to $\infty$ which is divided in to three regions labeled as 1, 2 and 3 with the boundaries between these regions placed at $-a$ and $a$[1]. We use our earlier definition of an associated number function $\delta\theta_i(t)$. If we now consider our correlation function as being generated by an underlying 3 state rate matrix $R$ and associated left and right eigenfunctions [118]:

$$\int_0^\infty C_{IJ}(t) dt = \int_0^\infty \left( [e^{tR}]_{IJ} P_J^{eq} - P_I^{eq} P_J^{eq} \right) dt \tag{3.10}$$

Doing a spectral decomposition of the right hand side of the equation:

$$\int_0^\infty C_{IJ}(t) dt = \int_0^\infty \sum_{n=2}^{3} [e^{\lambda_n t}] \psi_n^R(I) \psi_n^L(J) P_J^{eq} dt \tag{3.11}$$

[1]Since the potential is symmetric and centered upon 0, we must have that the two boundaries are equally spaced on each side (i.e. that regions 1 and 3 are identical), so our two boundary optimization is really a single boundary problem.

$$= \int_0^\infty \sum_{n=2}^{3} [e^{\lambda_n t}] \psi_n^R(I) \psi_n^R(J) dt \tag{3.12}$$

Then writing these elements out explicitly and exploiting that $\psi_2^R(2) = 0$ (and calling the matrix $C$ for a shorthand notation).

$$C = \int_0^\infty \begin{pmatrix} e^{\lambda_2 t}(\psi_2^R(1))^2 + e^{\lambda_3 t}(\psi_3^R(1))^2 & e^{\lambda_3 t}\psi_3^R(1)\psi_3^R(2) & e^{\lambda_2 t}\psi_2^R(1)\psi_2^R(3) + e^{\lambda_3 t}\psi_3^R(1)\psi_3^R(3) \\ e^{\lambda_3 t}\psi_3^R(1)\psi_3^R(2) & e^{\lambda_3 t}(\psi_3^R(2))^2 & e^{\lambda_3 t}\psi_3^R(2)\psi_3^R(3) \\ e^{\lambda_2 t}\psi_2^R(1)\psi_2^R(3) + e^{\lambda_3 t}\psi_3^R(1)\psi_3^R(3) & e^{\lambda_3 t}\psi_3^R(2)\psi_3^R(3) & e^{\lambda_2 t}(\psi_2^R(3))^2 + e^{\lambda_3 t}(\psi_3^R(3))^2 \end{pmatrix} dt$$

Since we are considering the special case of a symmetric potential we have the following identities:

$$\psi_2^R(1) = -\psi_2^R(3) \tag{3.13}$$

$$\psi_3^R(1) = \psi_3^R(3) \tag{3.14}$$

$$2\psi_3^R(1) + \psi_3^R(2) = 0 \implies \psi_3^R(2) = -2\psi_3^R(1) \tag{3.15}$$

$$C = \int_0^\infty \begin{pmatrix} e^{\lambda_2 t}(\psi_2^R(1))^2 + e^{\lambda_3 t}(\psi_3^R(1))^2 & -2e^{\lambda_3 t}(\psi_3^R(1))^2 & -e^{\lambda_2 t}(\psi_2^R(1))^2 + e^{\lambda_3 t}(\psi_3^R(1))^2 \\ -2e^{\lambda_3 t}(\psi_3^R(1))^2 & 4e^{\lambda_3 t}(\psi_3^R(1))^2 & -2e^{\lambda_3 t}(\psi_3^R(1))^2 \\ -e^{\lambda_2 t}(\psi_2^R(1))^2 + e^{\lambda_3 t}(\psi_3^R(1))^2 & -2e^{\lambda_3 t}(\psi_3^R(1))^2 & e^{\lambda_2 t}(\psi_2^R(1))^2 + e^{\lambda_3 t}(\psi_3^R(1))^2 \end{pmatrix} dt$$

We can recognise that $C$ has the form given below with only two free parameters, $X$ and $Y$.

$$C = \begin{pmatrix} X & Y & -X-Y \\ Y & -2Y & Y \\ -X-Y & Y & X \end{pmatrix}$$

A matrix of this form will have eigenvalues given by 3.16.

$$\mu = \begin{cases} 0 \\ 2X+Y \\ -3Y \end{cases} \tag{3.16}$$

Using this we find that the second eigenvalue of $C$ is given by:

$$\mu_2 = \int_0^\infty 2e^{\lambda_2 t}(\psi_2^R(1))^2 dt \tag{3.17}$$

From the normalization of eigenvectors (and again exploiting symmetry) we can write the square of the first element of the second right eigenvector in terms of the equilibrium probabilities.

$$\sum_{i=1}^{3} \frac{(\psi_2^R(i))^2}{P_i^{eq}} = 1 \implies (\psi_2^R(1))^2 = \frac{P_1^{eq}}{2} \tag{3.18}$$

Substituting this value for $\psi_2^R(1)$ in and performing the integration produces equation 3.19.

$$\mu_2 = \frac{-P_1^{eq}}{\lambda_2} \implies \lambda_2 = \frac{-P_1^{eq}}{\mu_2} \tag{3.19}$$

If we substitute in the our known expression for $\mu_2$ in terms of correlation function elements (from equation 3.16) then the quantity to be analytically maximised is given in terms of the equilibrium probability and the two unique integrated correlation function

elements.

$$\lambda_2 = \frac{-P_1^{eq}}{(2C_{11} + C_{12})} \tag{3.20}$$

Equivalently, we can write equation 3.20 in terms of relaxation time by inverting.

$$\tau_2 = -\frac{(2C_{11} + C_{12})}{P_1^{eq}} = -\frac{1}{P_1^{eq}}\left(2\int_0^\infty \langle \delta\theta_1(0)\delta\theta_1(t)\rangle dt + \int_0^\infty \langle \delta\theta_1(0)\delta\theta_2(t)\rangle dt\right) \tag{3.21}$$

We have explicit expressions for the slowest relaxation time in a two state system and a three state symmetric system and so we can proceed to convert these expressions in to mean first passage times and extremise with respect to the boundary position.

### 3.2.3 Correlation Functions and Spatial Integrals

The final piece of theory we need to introduce in advance of our results section is the link between time-integrated correlation functions and spatial integrals over the potential energy. Perico and Szabo [113] showed that (where $\beta = \frac{1}{K_B T}$) the time integral of an autocorrelation function can be written as a spatial integral over the whole coordinate as in equation 3.22.

$$\int_0^\infty \langle \delta\theta(x(0))\delta\theta(x(t))\rangle dt = \int_{-\infty}^\infty \frac{dx}{De^{-\beta v(x)}} \frac{\left[\int_x^\infty \delta\theta(y)e^{-\beta v(y)}dy\right]^2}{\int_{-\infty}^\infty e^{-\beta v(x)}dx} \tag{3.22}$$

This relation allows us to now directly connect our relaxation time to a spatial integral with a dependence on the boundary positions.

## 3.3 Results

### 3.3.1 Two State Relaxation Time

To calculate the relaxation time in terms of populations and MFPTs, we need to calculate two quantities $\int_0^\infty \langle \delta\theta_1(0)\delta\theta_1(t)\rangle dt$ and $\langle \delta\theta_1(0)^2\rangle$. Using equation 3.22 to rewrite our two-state relaxation time, we have equation 3.23.

$$\int_0^\infty \langle \delta\theta_1(0)\delta\theta_1(t)\rangle dt = \int_{-\infty}^{\infty} \frac{dx}{De^{-\beta v(x)}} \frac{\left[\int_x^\infty \delta\theta_1(y)e^{-\beta v(y)}dy\right]^2}{\int_{-\infty}^{\infty} e^{-\beta v(x)}dx} \tag{3.23}$$

Next we split the integral into two segments, one running from $-\infty$ to $a$ (region 1) and the other from $a$ to $\infty$ (region 2). We also make use of the fact that $\int_{-\infty}^{\infty} \delta\theta_1(y)e^{-\beta v(y)}dy = 0$ to change the limits of the inner $y$ integral.

$$C_{11} = \int_{-\infty}^{a} \frac{dx}{De^{-\beta v(x)}} \frac{\left[\int_{-\infty}^x \delta\theta_1(y)e^{-\beta v(y)}dy\right]^2}{\int_{-\infty}^{\infty} e^{-\beta v(x)}dx} + \int_a^{\infty} \frac{dx}{De^{-\beta v(x)}} \frac{\left[\int_x^\infty \delta\theta_1(y)e^{-\beta v(y)}dy\right]^2}{\int_{-\infty}^{\infty} e^{-\beta v(x)}dx} \tag{3.24}$$

We exploit the properties of the number function and define some state normalised probabilities $p_1(x)$ and $p_2(x)$.

$$p_1(x) = \frac{e^{-\beta V(x)}}{\int_{-\infty}^{a} e^{-\beta V(x)}dx} \tag{3.25}$$

$$p_2(x) = \frac{e^{-\beta V(x)}}{\int_{a}^{\infty} e^{-\beta V(x)}dx} \tag{3.26}$$

With these normalised probabilities, one can rewrite the time-integrated correlation function in the form of equation 3.27.

$$\int_0^\infty \langle \delta\theta_1(0)\delta\theta_1(t)\rangle dt = \langle\theta_1\rangle^2\langle\theta_2\rangle \int_{-\infty}^{a} \frac{dx}{Dp_1(x)}\left[\int_{-\infty}^{x} p_1(y)dy\right]^2 + (\langle\theta_2\rangle)^2\langle\theta_1\rangle \int_a^\infty \frac{dx}{Dp_2(x)}\left[\int_x^\infty p_2(y)dy\right]^2 \quad (3.27)$$

Similarly $\langle \delta\theta_1(0)^2\rangle$ is expressed as:

$$\langle \delta\theta_1(0)^2\rangle = \langle\theta_1\rangle\langle\theta_2\rangle \quad (3.28)$$

In our current notation, the expected value of the theta function is equivalent to the population of the state so we replace these with $p_1$ and $p_2$. To avoid confusion we emphasise the distinction between $p_1$ and $p_1(x)$, $p_1$ is the population of state 1 while $p_1(x)$ is the population of position $x$ relative to the population of state 1. With these definitions we have the relation that $p_1 p_1(x) = p(x)$ (or equivalently that $p_1(x) = \frac{p(x)}{p_1}$ so $p_1(x)$ is the population of the state relative to that of state 1). Using this and combining equations 3.27 and 3.28 with equation 3.9 we can obtain a more easily differentiable expression in the form of equation 3.29.

$$\tau_2(a) = p_2 \int_{-\infty}^{a} \frac{dx}{Dp_1(x)}\left[\int_{-\infty}^{x} p_1(y)dy\right]^2 + p_1 \int_a^\infty \frac{dx}{Dp_2(x)}\left[\int_x^\infty p_2(y)dy\right]^2 \quad (3.29)$$

The mean first passage times to the barrier from each side are given by the integrals in the above equation (see reference [119] for details).

$$t_{a2} = \int_a^\infty \frac{dx}{Dp_2(x)}\left[\int_x^\infty p_2(y)dy\right], \quad (3.30)$$

In equation 3.30, $t_{a2}$ is the expected time to reach a boundary $a$ with a starting pointing chosen randomly from state 2 with equilibrium probability ($t_{a1}$ is equivalently defined).

With this, 3.29 can be written in the compact form of equation 3.31.

$$\tau_2(a) = p_2 t_{a1} + p_1 t_{a2} \tag{3.31}$$

Since we have managed to write the relaxation time in terms of mean first passage times, we can differentiate with respect to the boundary position and find the condition which will maximise the relaxation time.

### 3.3.2 Optimization of Two State Boundary Position

The optimal barrier will be such that the $\tau_2$ is maximized with respect to the barrier position $a$.

$$\frac{d\tau_2}{da} = 0 \tag{3.32}$$

This requires the calculation of the derivative of each of the four components of equation 3.31. The derivatives of the $p$ terms are straightforward to calculate.

$$\frac{dp_2}{da} = \frac{-e^{-\beta v(a)}}{\int_{-\infty}^{\infty} e^{-\beta v(x)} dx} = -p(a) \tag{3.33}$$

$$\frac{dp_1}{da} = \frac{e^{-\beta v(a)}}{\int_{-\infty}^{\infty} e^{-\beta v(x)} dx} = p(a) \tag{3.34}$$

The more difficult term to differentiate is the mean first passage time. This requires use of the Leibniz integral rule [120] for differentiation with respect to the limits of an integral.

$$\frac{d}{dx}\left(\int_{a(x)}^{b(x)} f(x,t)\,\mathrm{d}t\right) = f(x,b(x)) \cdot b'(x) - f(x,a(x)) \cdot a'(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x,t) \tag{3.35}$$

Applying this gives the following result for the mean first passage terms.

$$\frac{dt_{a2}}{da} = \frac{d}{da}\left( \int_a^\infty \frac{dx}{Dp_2(x)} \left[ \int_x^\infty p_2(y)dy \right]^2 \right) \tag{3.36}$$

$$= \frac{-1}{Dp_2(a)} + \int_a^\infty dx \left[ \frac{d}{da}\left( \frac{1}{Dp_2(x)} \right) \left[ \int_x^\infty p_2(y)dy \right]^2 + \left( \frac{1}{Dp_2(x)} \right) \frac{d}{da} \left[ \int_x^\infty p_2(y)dy \right]^2 \right] \tag{3.37}$$

$$= \frac{-1}{Dp_2(a)} + \int_a^\infty dx \left[ \frac{-e^{-\beta v(a)}}{De^{-\beta v(x)}} \left[ \int_x^\infty p_2(y)dy \right]^2 + \frac{2}{Dp_2(x)} \frac{e^{-\beta v(a)}}{\int_a^\infty e^{-\beta v(x)}dx} \left[ \int_x^\infty p_2(y)dy \right]^2 \right] \tag{3.38}$$

$$= \frac{-1}{Dp_2(a)} - p_2(a)t_{a2} + 2p_2(a)t_{a2} = \frac{-1}{Dp_2(a)} + p_2(a)t_{a2} \tag{3.39}$$

A similar expression is found for the mean first passage time from the other side of the boundary.

$$\frac{dt_{a1}}{da} = \frac{1}{Dp_1(a)} - p_1(a)t_{a1} \tag{3.40}$$

Substituing all these into equation 3.32 and simplifying it is found that the relaxation time is maximized at the condition given in equation 3.41.

$$p_2(a)t_{a2} = p_1(a)t_{a1} \tag{3.41}$$

Equivalently this can be written in terms of the equilibrium probabilities of the two

clusters in a 'flux to the boundary' form.

$$\frac{p_1}{t_{a1}} = \frac{p_2}{t_{a2}} \tag{3.42}$$

This gives us back the intuitive result that the relaxation time will be optimized by a boundary which ensures that the flux through the boundary is equal in both directions.

### 3.3.3 Three State Symmetric Relaxation Time

In this section we present an equivalent derivation for the three state case we examined earlier. We have the relaxation time in terms of the coarse-grained time-integrated correlation functions. We first compute the relaxation time in terms of mean first passage times similarly to before.

$$\tau_2 = -\frac{(2C_{11} + C_{12})}{p_1} = -\frac{1}{p_1}\left(2\int_0^\infty \langle \delta\theta_1(0)\delta\theta_1(t)\rangle dt + \int_0^\infty \langle \delta\theta_1(0)\delta\theta_2(t)\rangle dt\right) \tag{3.43}$$

As in the previous derivation for the two boundary case, the $C_{11}$ term behaves as though we are doing a two state optimisation (1 vs 2+3) and can be written completely analogously to equation 3.29.

$$C_{11} = (1-p_1)^2 \int_{-\infty}^{-a} \frac{dx}{Dp(x)}\left[\int_{-\infty}^{x} p(y)dy\right]^2 + p_1^2 \int_{-a}^{\infty} \frac{dx}{Dp(x)}\left[\int_{x}^{\infty} p(y)dy\right]^2 \tag{3.44}$$

The time-integrated correlation function $C_{12}$ can also be computed.

$$C_{12} = -p_2^2 \int_{-\infty}^{-a} \frac{dx}{Dp(x)}\left[\int_{-\infty}^{x} p(y)dy\right]^2 - 2p_1^2 \int_{-a}^{a} \frac{dx}{Dp(x)}\left[\int_{x}^{\infty} p(y)dy\right]^2 \tag{3.45}$$

$$+p_1^2 \int_{-a}^{a} \frac{dx}{Dp(x)} \left[ \int_x^{\infty} p(y)dy \right]$$

By substituting the expressions for $C_{11}$ and $C_{12}$ in to equation 3.43 we find that we get a significant amount of cancellation such that our relaxation time can be written simply as 3.46.

$$\tau_2 = \int_{-\infty}^{-a} \frac{dx}{Dp_1(x)} \left[ \int_{-\infty}^{x} p_1(y)dy \right]^2 + p_1 \int_{-a}^{a} \frac{dx}{Dp(x)} \left[ \int_x^{\infty} p(y)dy \right] \tag{3.46}$$

As with the two state case, we can recognise these integral quantities as mean first passage times and rewrite $\tau_2$ concisely as equation 3.47.

$$\tau_2 = t_{-a1} + p_1 t_{-aa} \tag{3.47}$$

$p_1$ is the probability to be in state 1. $t_{-aa}$ is the mean survival time for a particle starting at $a$ to hit the boundary at $-a$. $t_{-a1}$ is the mean first passage time starting from the region 1 to reach the boundary at $-a$.

### 3.3.4 Optimization of Three State Relaxation Time

Differentiating the relaxation time equation 3.43 with respect to the boundary position $a$ and equating to zero results in the equation 3.48 for optimal boundary placement.

$$p(a)(2C_{11} + C_{12}) + p_1 \frac{d(2C_{11} + C_{12})}{da} = 0 \tag{3.48}$$

We have expressions for $C_{11}$ and $C_{12}$ from the preceding section and simply need to calculate their derivatives. We begin by differentiating equations 3.44 and 3.45.

$$\frac{dC_{11}}{da} = 2p(a) \left[ (1 - p_1) \int_{-\infty}^{-a} \frac{dx}{Dp(x)} \left[ \int_{-\infty}^{x} p(y)dy \right]^2 - p_1 \int_{-a}^{\infty} \frac{dx}{Dp(x)} \left[ \int_x^{\infty} p(y)dy \right]^2 \right] \tag{3.49}$$

$$\frac{dC_{12}}{da} = -4(1-2\langle n_1\rangle)p(a)\int_{-\infty}^{-a}\frac{dx}{Dp(x)}\left[\int_{-\infty}^{x}p(y)dy\right]^2 + 4\langle n_1\rangle p(a)\int_{-a}^{a}\frac{dx}{Dp(x)}\left[\int_{x}^{\infty}p(y)dy\right]^2$$

$$-2\langle n_1\rangle p(a)\int_{-a}^{a}\frac{dx}{Dp(x)}\left[\int_{x}^{\infty}p(y)dy\right] \tag{3.50}$$

Putting the above together in equation (15) and cancelling terms results in the simple expression in equation 3.51 for the condition of optimal relaxation time.

$$p_1\int_{-a}^{a}\frac{dx}{Dp(x)}\left[\int_{x}^{\infty}p(y)dy\right] = \int_{-\infty}^{-a}\frac{dx}{Dp_1(x)}\left[\int_{-\infty}^{x}p_1(y)dy\right]^2 \tag{3.51}$$

We can again interpret these quantities in terms of mean first passage times and see that this equates the two quantities which we found previously to constitute our slowest relaxation time.

$$p_1 t_{-aa} = t_{-a1} \tag{3.52}$$

In the results section of this chapter, we will see that this condition, whilst not intuitive does result in the correct placement of the optimized boundaries. It is not clear how to interpret this result, we speculate the condition of making the potential symmetric in order to make the calculation tractable collapses the terms of the equation in a way that obscures the deeper laws which are governing the optimal clustering.

Rewriting equation 3.52 in a flux representation one can see that the condition can be stated as the flux of transitions hitting the boundary is equal to the rate of transitions through the middle state.

$$\frac{p_1}{t_{-a1}} = \frac{1}{t_{-aa}} \tag{3.53}$$

## 3.4 Estimating MFPTs from MD data

We have derived analytically exact expressions for the MFPT conditions which describe the 1-D clustering found by maximising the slowest relaxation time found by using the Hummer-Szabo clustering in a two state and three state symmetric case. To test the derived equations on a realistic model, we need to be able to estimate MFPT quantities from discrete datasets. Here we present three possible ways of computing these quantities: using Markov chain theory, explicit counting and discrete approximation of the integral expressions of MFPTs.

### 3.4.1 MFPT from Markov Model

The first option is to construct a maximum likelihood Markov state model from the simulation data [118] and use known theory of Markov chains to compute MFPTs between states. We used the Meyer method [121, 122] to calculate MFPTs from the obtained discrete state Markov model, which requires solving a system of simple linear equations of the form in equation 3.54 (similarly to the derivation of the Kemeny constant in section 2.6.3).

$$t_{ji} = \tau M_{ji}(\tau) + \sum_{j' \neq j} M_{j'i}(\tau) t_{jj'}, \tag{3.54}$$

$M_{ji}(\tau)$ is the Markovian transition probability to make a transition from $i$ to $j$ in the time interval $\tau$, and $t_{ji}$ is the MFPT from state $i$ to $j$.

### 3.4.2 Explicit counting from MD trajectories

The next approach we consider is to use the discrete time trajectory $\mathbf{x} = x_1, x_2, ..., x_n$ with timestep $\tau$ and to explicitly count the mean first passage time from the observed transitions.

For example, if we observe the system to reach the boundary at times $T_1$, $T_2$, $T_3$, ..., $T_k$, starting in state 2, then the number of steps spent in state 1 will be given by

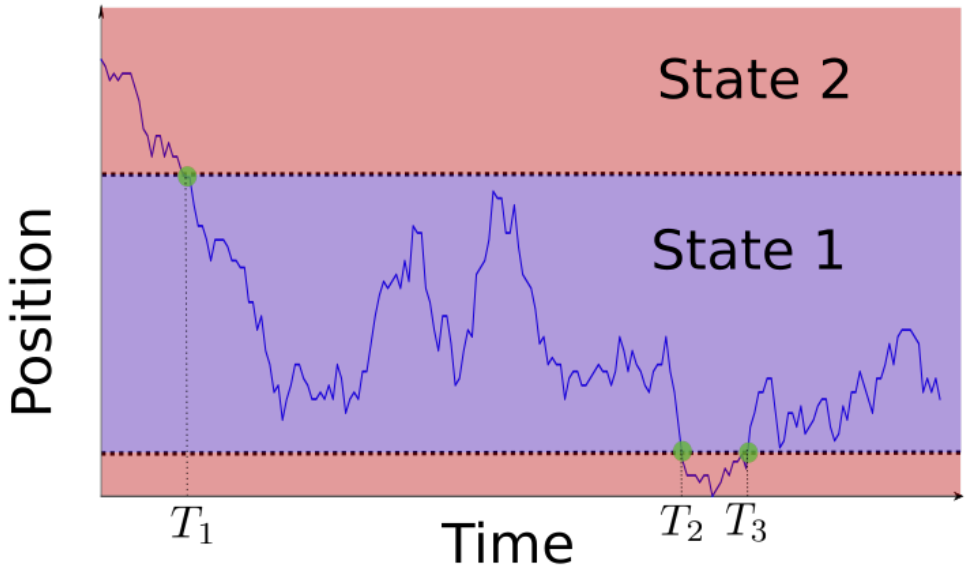


Figure 3.1: Illustration of calculating MFPTs by explicit counting on a periodic coordinate.

$N_1 = \frac{T_2 - T_1}{\tau}$, $N_2 = \frac{T_4 - T_3}{\tau}$. Whenever the system crosses the boundary, it enters the other state and so we use every second pair of crossing times to count the MFPT. The other sum $(T_3 - T_2, T_5 - T_4)$ is used for the MFPT from the opposite side of the boundary. This idea is demonstrated in figure 3.1.

To perform this calculation in practice we consider the MFPT from each of our microstates to the boundary. Let's say for example that for crossing event $i$ we observe a trajectory of length $N_i$ to reach the boundary. This trajectory is then immediately followed by another trajectory of length $N_i - 1$ and then by $N_i - 2$ and so on until we hit the boundary. We can the approximate the MFPT by taking the sum of all these crossing events. We then divide this by the total simulation time in the state in order to properly normalize. The formal expression of the above text is given in equation 3.55.

$$\frac{\sum_i^{k/2} \sum_{j=1}^{N_i} j}{\sum_i^{k/2} N_i} = \frac{\sum_i^{k/2} (N_i - 1) N_i / 2}{\sum_i^{k/2} N_i} \tag{3.55}$$

For the above equation to hold true, we assume that over long simulation times, each microstate is explored with equilibrium probabilities. In the subsequent results we will assume this to be true as this is typically assumed whenever one is constructing a MSM. This algorithm only requires identifying the boundary crossing times over the trajectories, therefore it can be more efficient than the previous approaches. However, as it requires well converged trajectories, its numerical error due to non-Markovian effects can be larger.

### 3.4.3 Discrete approximation of integrals

As a final alternative method, the derived MFPT relations can be approximated numerically by discretizing the integral expressions by which their are defined. This discretization requires knowledge of the diffusion coefficient $D$, which we have previously assumed to be constant. Since we do not known this value a priori the mean first passage times obtained in this way will have the correct behaviour but not the correct magnitude. Since they will be proportional to their true values, the position at which they are equal will be unchanged (again this assumes a constant $D$). In terms of computational efficiency, this method is much faster than the Hummer-Szabo method as it requires only the equilibrium probabilities of the microstates involved. To illustrate how one might impliment this in practice, a discretized form of the mean first passage time $t_{a1}$ is shown in equation 3.56.

$$t_{a1} = \sum_{i=x_1}^{x_a} \frac{\Delta x}{D p_1(i)} \left[ \sum_{j=x_1}^{i} p_1(j) \Delta x \right]^2 \tag{3.56}$$

The integrals have been replaced with discrete summations over the microstate positions where the microstate spacing is $\Delta x$. One could obtain an estimate for the diffusion coefficient by comparing the MFPTs obtained via discrete approximation with those obtained by one of the previous two methods. We examine this method for diffusion estimation in our results section.

## 3.5 Computational Verification of Results

Thus far in this chapter, we have demonstrated that by assuming a diffusive process with Smoluchowski dynamics we can make a connection betwen the Hummer-Szabo clustering described in the previous chapter and analytic expressions for the mean first passage time.

In this results section we demonstrate that our derived expressions hold true on both synthetic data and molecular dynamics simulation trajectories. In particular we show that the optimal clustering obtained from implementing Hummer-Szabo matches exactly the clustering found by enforcing the MFPT condition derived previously.

### 3.5.1 Analytic Examples

The first examples considered are symmetric potentials, with double and triple well, as shown in Fig. 3.2 and 3.3. These potentials are described by equations 3.57 and 3.58 respectively (where the values $c_0$ and $c_1$ are chosen such that $v(x)$ takes the minimum value 0 in the range $-4\pi$ to $4\pi$).

$$v(x) = -\sin(\frac{x-\pi}{2}) + c_0 \tag{3.57}$$

$$v(x) = \sin(\frac{1.5x-\pi}{2}) + c_1 \tag{3.58}$$

For each of these two potentials we consider two cases, two state clustering and three state clustering. In each case we plot 6 quantities.

- The potential $v(x)$.
- The left side of equation 3.41 (or equation 3.52).
- The right side of equation 3.41 (or equation 3.52).

- The MFPT barrier position (location where the previous two quantities are equal).
- The barrier predicted by implementing the variational HS protocol of chapter 2.
- The barrier predicted by using the PCCA+ clustering method.

**Double-well Potential**

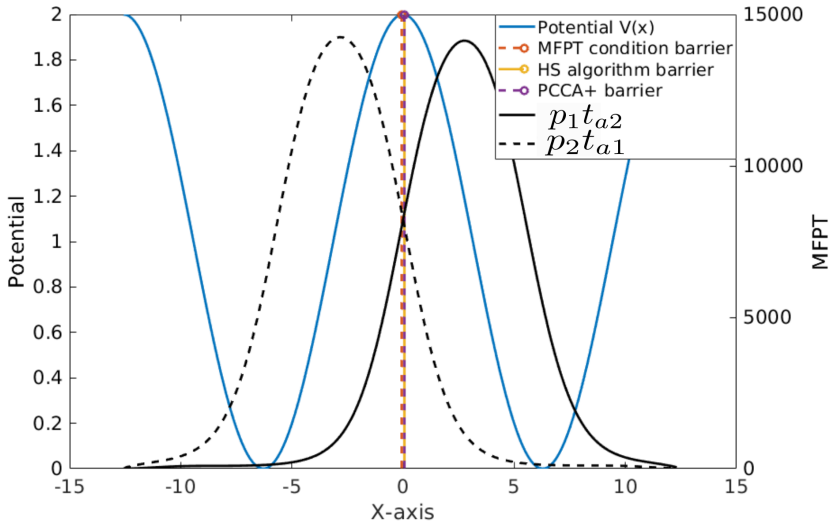


Figure 3.2: Clustering of a Double Well Potential in to Two States

In figures 3.2 and 3.3 we show the result of clustering a 2D potential in to two and three clusters respectively. In the 2 cluster case, we find that the intuitive result of the clustering barrier being placed at the top of the potential at $x = 0$ is reproduced in all three cases (HS, MFPT and PCCA+). However in the 3 cluster case we see a divergence between the results obtained by HS/MFPT vs PCCA+. The HS/MFPT identify two stable states separated by a transition state of finite size at the peak of the potential. Importantly, the size of the transition state/ boundary positions match exactly (the positions are shifted slightly to make both lines visible on the plot).

In contrast, the PCCA+ does not find a transition state. The PCCA+ method assigns to each microstate a probability to occupy each cluster. Implementing the PCCA+ with 3 clusters finds that the states near the peak of the potential have a non-zero probability to occupy the central cluster. However when examining which cluster each microstate has the largest probability for, there is no microstate for which the middle state is the most probable. As such, no microstates get assigned to this middle region.

### Triple Well Potential

In figures 3.4 and 3.5 we show the result of clustering a triple well potential in to two and three clusters respectively.

In figure 3.4 we find that the mean first passage time quantities cross three times. These three crossing correspond to two local maxima and one local minimum (seen as two of the crossing points occur at large values than the other). This is intuitively reasonable since clustering a symmetric triple well in to two states should lump two of

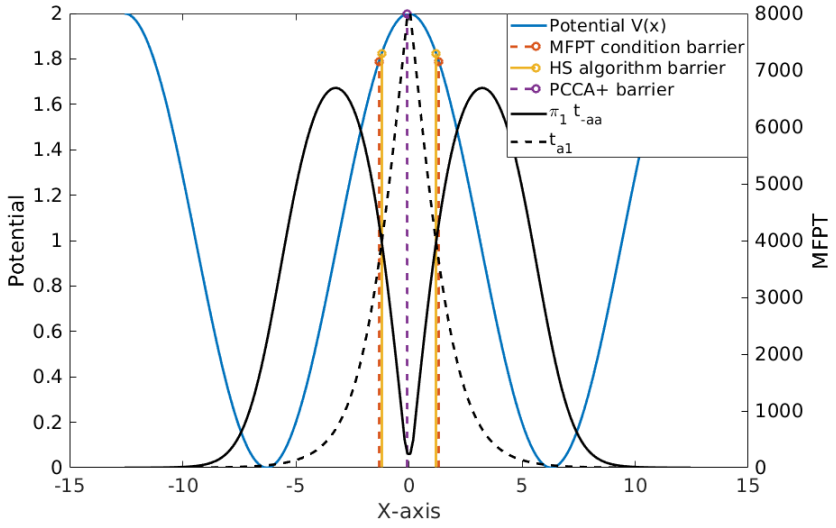


Figure 3.3: Clustering of a Double Well Potential in to Three States

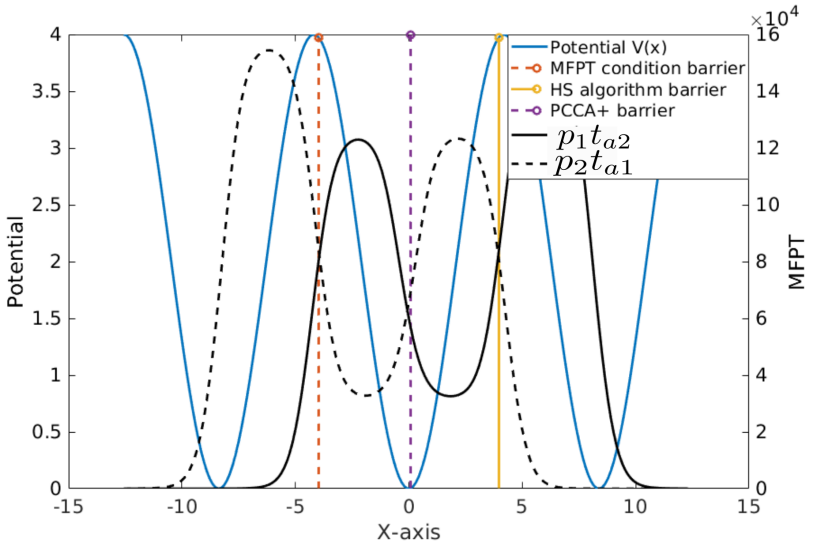


Figure 3.4: Clustering of a Triple Well Potential in to Two States

the wells together and have no preference for which pair it groups. In the figure, we find that HS and MFPT once again find identical boundary positions at the peaks of the energy barriers.

Again, we find that the PCCA+ algorithm obtains a distinct clustering. In this case, it chooses a particularly bad clustering by dividing perfectly in two and splitting the central well. This is problematic as now both of the macrostates will contain free energy barriers within them, leading to highly non-Markovian effects. However, this test is possibly harsh on the PCCA+ algorithm as in reality one is unlikely to obtain a perfectly symmetric potential with equal height barriers. In a more realistic potential, the PCCA+ will immediately switch to placing the barrier at the peak of the higher boundary.

Examining the three cluster situation in figure 3.5, we observe that all three methods (HS, MFPT and PCCA+) match exactly again as we would expect as all methods are able to identify metastable states. In particular, we can see that PCCA+ is effective at obtaining metastable state boundaries when the correct number of states is identified.

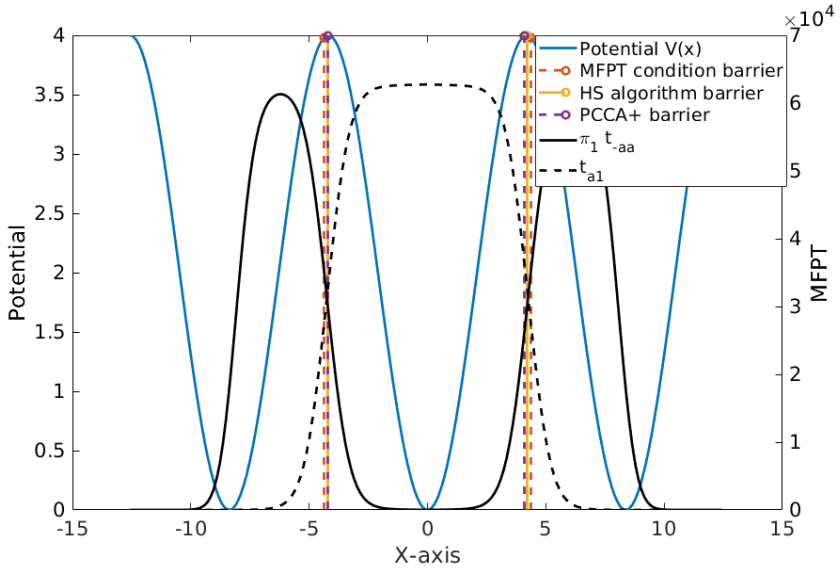


Figure 3.5: Clustering of a Triple Well Potential in to Three States

### 3.5.2 Pentalanine MD simulation

The next test of the derived equations is to demonstrate that they hold true when examining more realistic time series data such as MD simulation data. To do this we generated data by performing simulations of pentalanine in a 20Å box of TIP3P water[2] (figure 3.6). We used the online tool CHARMM-GUI for setting up the system [123].

CHARMM-GUI produces all the files need to perform the simulation but requires the user to make some parameter decisions. The ligand was capped with an acetyl at the N terminus and an methylamine group at the C terminus. Additionaly we solvated the ligand with an explicit water box. The simulations were run using NAMD [124] at a temperature of 300 Kelvin and time step of 2 femtoseconds with a Langevin thermostat. Following an equilibration run, a total of 1 microsecond production run was performed.

To test the derived equations in this paper, the 10 backbone dihedral angles $(\Phi, \Psi)$ of the peptide are extracted from the simulation data. As an example, here we used

[2]This box provides a 20Å buffer of water in each direction around the ligand, resulting in a cube with each dimension approximately 42Å.

$\Psi_1$ and $\Phi_1$ to construct Markov models from which the mean first passage times can be extracted. Similar results to those presented for $\Psi_1$ and $\Phi_1$ were obtained and are left to the appendix.

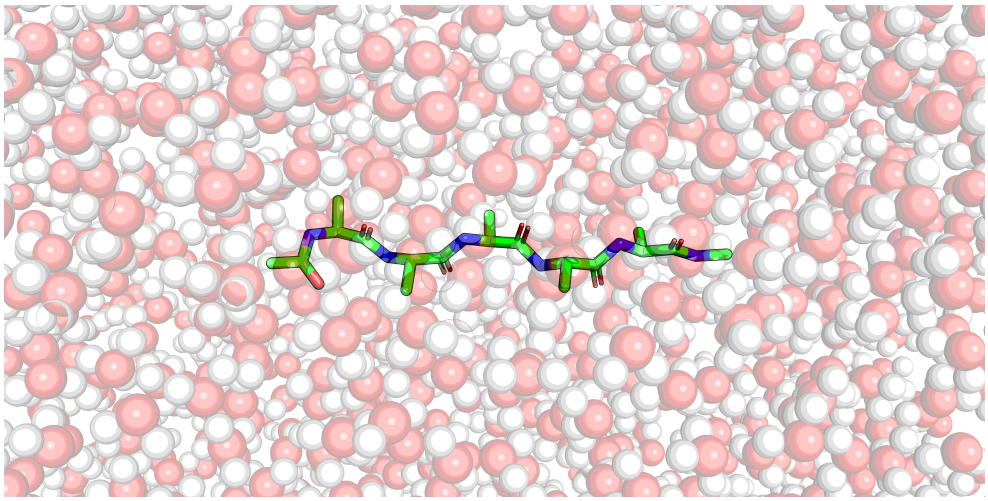

Figure 3.6: Illustration of the $Ala_5$ simulation system.

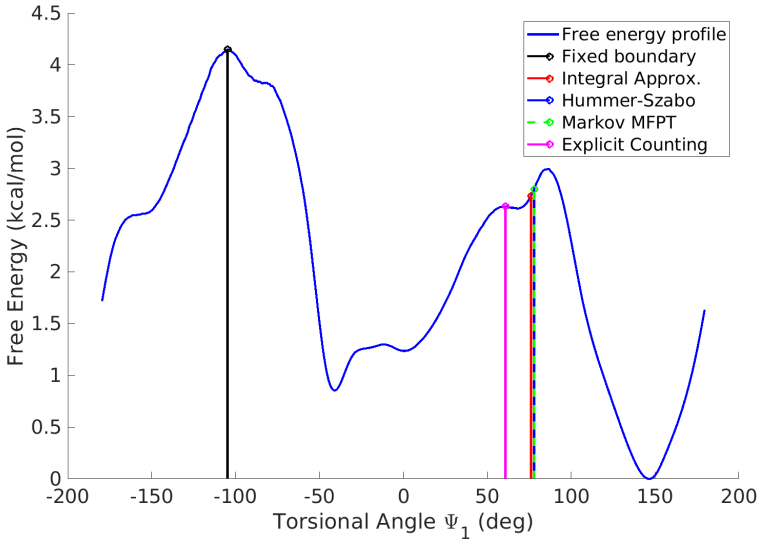


Figure 3.7: Free energy profile and optimal boundaries for first pentalanine Ramachandran angle.

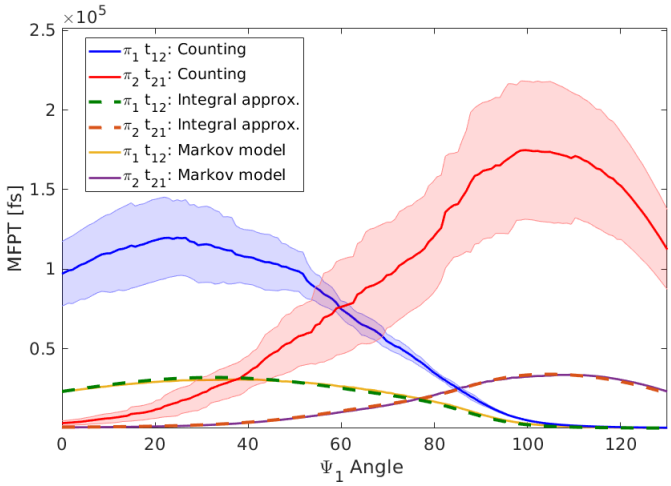


Figure 3.8: Explicit counting of MFPT from discrete data.

Applying the theory for estimating MFPTs described previously requires us to account for the fact that the coordinates are periodic[3]. To address this, we fix one boundary at the free energy maximum and then use our equations to identifying the position of the other boundary.

We implemented both the explicit counting procedure and the integral approximation method to estimate the MFPTs, as well as the Hummer-Szabo method. We find that like the analytic potentials the potentials are almost identical (figure 3.7). The explicit counting is slightly shifted due to the errors from the finite trajectory data.

In figure 3.8, we examine more closely the MFPTs obtained from the explicit counting. The error bars are obtained by splitting the trajectory in to four parts and calculating the variance on the resulting MFPTs. We can see that the functional dependence of the counting method is similar to the other two approaches which agree very closely. We postulate that the discrepency between the methods is due to the explicit counting method being more sensitive to the equilibrium sampling of states and requiring longer simulation data for accurate statistics.

At the same time, the agreement between the integral approximation and the

[3]For a periodic coordinate, two boundaries are required for a two state clustering.

Markov model arises in part from our fitting the curves together to obtain an estimate for the diffusion coefficient $D$. The closeness of the agreement demonstrates that the assumption of a constant diffusion coefficient is a good approximation since the Markov model MFPTs do not require the assumption of constant $D$. The fitted diffusion coefficient is $D \approx 1.82$ deg$^2$/fs.

## 3.6 Conclusions

In this chapter we have examined the fundamental theory behind the timescale based clustering procedure we proposed in Chapter 2. By connecting the clustered relaxation times to the MFPTs calculated on the underlying potential and optimising, we were able to provide some simple kinetic equations which could describe the optimal boundary positions.

We verified these equations by testing on two systems, a simple analytic test potential and an MD simulation of a small peptide. We provided some discussion on methods for the computation of mean first passage times from discrete time series data and in analysing the MD simulation data contrasted the results obtained from using different methods.

So far we have, in the past two chapters, examined a new eigenvalue based clustering protocol and derived a mean first passage time based framework for interpreting and analysing the results. The method has thus far proven effective at the tasks which we derived it to do (robustly identifying transition states). However we have not yet addressed the pressing issue that our method requires searching through all possible clusterings to find the one which optimises the parameter. We come to this in the next chapter by proposing an algorithm for efficient search through the clustering space to optimise the variational parameter.

*"My candle burns at both ends; It will not last the night;*
*But ah, my foes, and oh, my friends— It gives a lovely*
*light!"*

Edna St. Vincent Millay, A Few Figs from Thistles

# 4

# Efficient Clustering of High Dimensional Networks

## 4.1 Introduction

In the two previous chapters we have laid out a new eigenvalue based clustering method for identifying coarse-grained states. The method suggests iteratively searching through possible clusterings to find the variational optimum. In this chapter we extend this method in two directions. Firstly, the suggested protocol of searching all clusterings

clearly becomes inefficient for systems with large numbers of microstates or large numbers of clusters. To address this issue, we develop a parallel tempering inspired approach to improve the efficiency of this protocol. Secondly, we have thus far considered only the clustering of kinetic network models such as MSMs. Here we will seek to extend this to more general geometric networks and demonstrate that the algorithm can yield interesting results for systems of this kind. We begin by introducing the theory of parallel tempering to motivate our new clustering algorithm.

## 4.2 Parallel Tempering

Parallel tempering is a simulation method which was developed for the purposes of finding the minimum energy configuration of some physical systems [125]. One runs $N$ simulations of a system in parallel, each at a different temperature. After each time step, each simulation is proposed to swap its current configuration for some new configuration. If the proposed configuration reduces the energy (i.e. improves the parameter to be optimized) then it is accepted, however if the energy is increased then the configuration is only accepted with a probability that depends on the difference between the current energy $E_{old}$ and the proposed energy $E_{new}$ as well the temperature of the simulation $T_i$ as in equation 4.1.

$$p = \min\left(1, e^{(E_{old}-E_{new})\left(\frac{1}{k_B T_i}\right)}\right) \tag{4.1}$$

$k_B$ is the Boltzmann constant. After some number of simulation steps, the configurations at different temperatures are interchanged with a probability given in equation 4.2 where $i$ and $j$ index the simulations being interchanged.

$$p = \min\left(1, e^{(E_i-E_j)\left(\frac{1}{kT_i}-\frac{1}{kT_j}\right)}\right) \tag{4.2}$$

$E_i$ and $T_i$ are the energy and temperature of simulation $i$ respectively and sim-

ilarly for the $j$ subscripts. This procedure results in the high temperature replicas exploring the configuration space freely while the low temperature replicas find the minimum energy state more delicately. The interchanging of configurations between different temperatures means that if a high temperature simulation finds a good configuration then this will be likely to cascade down to be explored more delicately by lower temperature simulations.

## 4.3 Existing uses of Tempering

Tempering inspired approaches have been used beyond their original domain to find configurations of systems which extremize some variational parameter of interest. Of particular interest to our discussion here are the fields of network and graph theory where tempering approaches have been used for finding communities in networks. In these applications, they typically use modularity [126] as the parameter to optimise. Modularity is a measure for whether the number of links between nodes within clusters is greater than what would be expected if the links were generated uniformly at random. These tempering approaches have been effective in the field at finding communities but are not used in practice as they are computationally slow [127].

## 4.4 Parallel Tempering for Variational Clustering

We propose that a parallel tempering approach can be implemented for finding the choice of clustering which optimises our chosen timescale parameter. We will here consider this parameter to be the slow timescales in the system although this could be adapted depending on the specific application (network modularity, conductance). In this section, we lay out the steps in our new method Parallel Tempering for Variational Clustering (PTVC).

First one calculates the rate matrix elements of the system if these are not already known. Then using the obtained rate matrix $\mathbf{K}$, one can identify the most kinetically

distinct states either by looking for the pair of states which have the largest mutual mean first passage time between them (i.e. $i$ and $j$ which maximise the mean first passage time in both directions as in equation 4.3)[1]. Once these two states are identified, the remaining states can be ordered based on their likelihood to first reach one state over the other.

$$\max_{i,j}(t_{ij} + t_{ji}) \tag{4.3}$$

With this 1-D ordering of the states, a set of simulations can be initialised by placing boundaries randomly along the ordering. This placing of boundaries along the 1-D ordering provides our starting configurations for the simulations[2]. Then at each time step of the simulation, the nodes which are connected to neighbouring clusters are identified and a swap is proposed. The value of the parameter for this new clustering is calculated and accepted/rejected in the same manner as for traditional parallel tempering in equation 4.1. The method described in words above is shown graphically in figure 4.1.

Other methods in the field of network cluster identification have employed the idea of introducing the concept of an artificial temperature to accelerate a variational search through conformations [128], typically presented as 'simulated annealing' methods. These simulated annealing approaches have been shown to find optimized parameter values but are slow. Our method differs from these existing methods in a number of important respects.

- Simulated annealing progressively heats and cools the systems to explore configurations. In contrast, parallel tempering runs parallel simulations at multiple temperatures and interchanges configurations at neighbouring temperatures.

[1] One can also compute this pair of states by examining the second eigenvector of the rate matrix and looking for the most negative and most positive elements. In our studied examples we found that these two approaches often produced similar choices of $i$ and $j$.

[2] To be exact we in fact attempt $N_{bound}$ different placements of boundaries (for some very large value of $N_{bound}$) and take the best performing subset to initialise $N_{sim}$ simulations ($N_{sim} << N_{bound}$). This helps us to avoid wasting time by initialising simulations from very bad configurations.

- Our method employs the kinetic timescales of the system as the variational parameter to identify transition states as opposed to modularity.
- We also introduce a kinetically motivated initial ordering of the states to enhance the quality of the starting clusters.

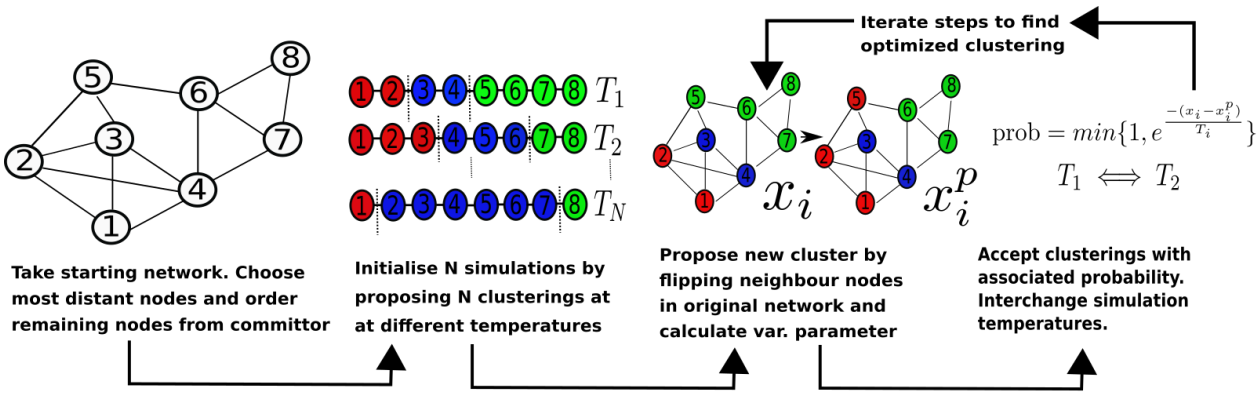


Figure 4.1: Illustration of parallel tempering clustering algorithm.

A practical consideration that requires discussion is the choice of temperature to be used. In this clustering context, the temperature has no physical motivation. It exists only to manage the probability of accepting proposed switches for different simulations. To determine the temperature we use, we propose to initialise the temperatures of all simulations as $T = T_i$ where $i \in 1, ..., N_{sim}$, $T_1 < T_2 < ... < T_{N_{sim}}$ and $T_{N_{sim}} << 1$. This effectively enforces that initially only proposed moves which optimize the parameter are accepted. From here, we can then slowly increase the temperatures (while tracking the acceptance probabilities) until the average acceptance probability over all simulations reaches 50%[3]. This removes the danger of initially over estimating

[3]In order to finish with useful temperatures it is still important to make a good initial choice for the range of magnitudes of the original temperatures. Although they all are initialised at much less than 1, the initial relative magnitude will be preserved under linear scaling. As a rule of thumb, we found three orders of magnitude from coldest to hottest ($T_{N_{sim}} \approx 10^3 T_1$) was effective although this will be somewhat dependent on the system of interest.

the temperature and causing the clusterings in all parallel simulations to deform from their initially well chosen configurations.

We demonstrate here that using this algorithm can yield dramatic speed increases in finding the variationally optimal clustering. We will examine the speed increases obtained for a high dimensional model which is difficult to compute exhaustively.

## 4.5 Application to Geometric networks

Having addressed the first of this chapters two aims by developing a new algorithm for efficiently identifying optimal clusterings, we now discuss how the variational protocol which we have developed might be extended to more general geometric networks (as opposed to the kinetic networks we have examined thus far).

In a geometric network we have a set of nodes and an associated adjacency matrix $A$ that describes which pairs of nodes are connected. For example, $A_{ij}$ equals one if $i$ and $j$ are connected and is zero otherwise. In this model, there is no inherent definition of kinetics so to apply our approach to identify clusters[4] we need to define a rate matrix $K$ from our adjacency matrix.

To artificially construct a rate matrix for the geometric network we define the transition rates from the adjacency matrix via equation 4.4 so that each nodes outward flow is distributed over all the nodes to which it is connected. This will allow less well connected nodes to have fast transition rates to highly connected nodes.

$$k_{ji} = \frac{A_{ji}}{\sum_{i'} A_{i'i}} \tag{4.4}$$

Due to the nature of how we made the transition from the adjacency matrix to the rate matrix, we will be restricted to applying our algorithm to networks which are well approximated by a stochastic block model (SBM) [129, 130]. The adjacency matrix of

[4]What we have thus far referred to as clusters are typically called 'communities' in the networks field.

a SBM is constructed according to equation 4.5.

$$P(A_{ij} = 1) = \frac{cW(x_i, x_j)}{NP(x_i)P(x_j)} \tag{4.5}$$

$x_i \in 1, ..., q$ indicates to which cluster node $i$ belongs (where there are q clusters in total). $P(x_i)$ is the probability of a node being in cluster $x_i$. $W(x_i, x_j)$ is the probability of a link existing being cluster $x_i$ and $x_j$. $N$ is the total number of nodes in the network and $c$ is a chosen constant which gives the average number of edges per node.

Existing methods for identifying clusters in geometric networks typically focus on either using the eigenvectors of the adjacency matrix (similarly to PCCA+) or optimizing some geometric parameter such as modularity (a measure of whether the nodes within the cluster have a greater than random density of linkage). In close parallel to the MD simulation field, these methods are effective at finding highly connected regions of the network but do not typically identify the transition state like regions which we have been interested in in this thesis (for a comprehensive overview of cluster identification see reviews by Fortunato [127, 131]).

## 4.6 Results

In the results section of this chapter we examine the clusterings obtained for some geometric networks and also investigate how the algorithms efficiency scales with the number of possible clusterings.

### 4.6.1 Computational Efficiency

The number of possible arrangements of states for a system with $N_{micro}$ microstates and $N_{macro}$ macrostates will be $\mathcal{O}(N_{micro}^{N_{macro}})$. Here will compute the time to find the optimal state for a high dimensional model.

One drawback with PTVC is that it has no objective measurable criteria for con-

vergence. With an exhaustive search one can know precisely that the optimum has been found but with PTVC, one must define a time interval such that if the optimum has not changed then one declares the algorithm converged. To assess the efficiency we will contrast the computational time of a complete exhaustive search vs the time for PTVC to find the optimum. This is a somewhat unfair comparison as in practice, the PTVC will require longer for the user to know that it has converged. However, we want to avoid distorting our results by adding an arbitrarily chosen convergence check time to our simulation time[5].

To demonstrate the computational efficiency of our PTVC method, we examine a four state clustering of the 625 microstate, 2-D kinetic system (previously examined in Chapter 2). We observed that running the algorithm through Matlab on an Intel Core i3 processor, the optimal configuration is found in on average 623 seconds. This requires computing the variational parameters for 26670 configurations. In comparison, allowing for a one dimensional ordering of the 625 states, one would expect to search through 40,495,000 possible configurations. A complete exhaustive search would require far more configurations and quickly become intractable.

### 4.6.2 Geometric Networks

In this second results section, we apply our protocol to the general geometric networks described previously and demonstrate that this eigenvalue based approach can prove useful even for non-kinetic systems.

#### Stochastic Block Model

The first test system we consider is a randomly generated SBM where each node is assigned to cluster and the likelihood of intracluster linkage is much higher than inter-cluster linkage.

[5]The convergence rule which we chose for our code was that if the code had been running for longer than some short prechosen time and the time since the last improvement is more than 50% of the total simulation time then the system is said to have converged.

Here we generate a random three cluster adjacency matrix with parameter values of $c = 4$, $N = 60$, $W = 0.7$ for same cluster nodes, $W = 0.005$ for nodes in distinct clusters and $P(x_i) = 1/3$ for each cluster (equation 4.5). The generated three cluster system and the associated four macrostate grouping found using three different variational kinetic parameters are shown in figure 4.2.

Similarly to the observations of the kinetic systems in chapter 2, we find that clustering in to four states can find the three clusters as well as the short-lived transition state regions. In figure (a), the slowest timescale identifies a transition state linking the least well connected state to the more highly connected regions. In figure (b), we instead find a transition state region linking the next fastest dynamics. Finally in (c) when we use all the timescales (Kemeny constant) we find that this method takes the most well connected state and divides it. The results observed here can be thought of as completely analogous to the 1-D smooth potential of Chapter 2.

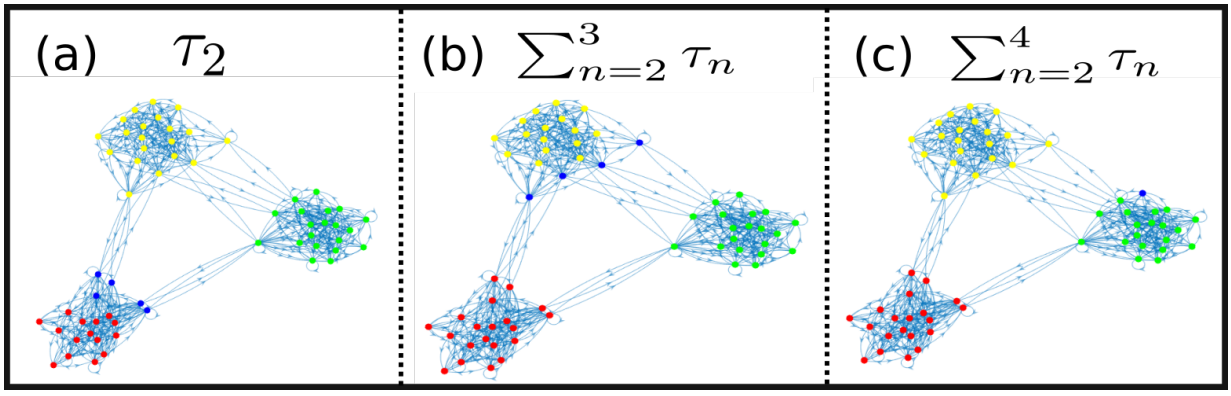


Figure 4.2: Four state clustering of random stochastic block model network.

**Santa Fe Collaboration Network**

For our final example, we consider a real world network, namely the Santa Fe collaboration network, a popular network for testing clustering algorithm. This network is composed of a set of 118 researchers who co-published, there are three main clus-

ters with some interdisciplinary researchers creating links between the major research groups.

The results of our clustering is demonstrated in figure 4.3. We have performed our kinetic clustering to group in to two, three and four states using our various clustering parameters ($\tau_2$, $\sum_{n=2}^{3} \tau_n$). Firstly in (a) we use $\tau_2$ to group in to two clusters. We find that the algorithm splits apart the most well connected cluster (red) from the other two clusters. In extending to a three state clustering in (b) with $\tau_2$, the algorithm now separates apart the two clusters which we previously aggregated in the two state clustering.

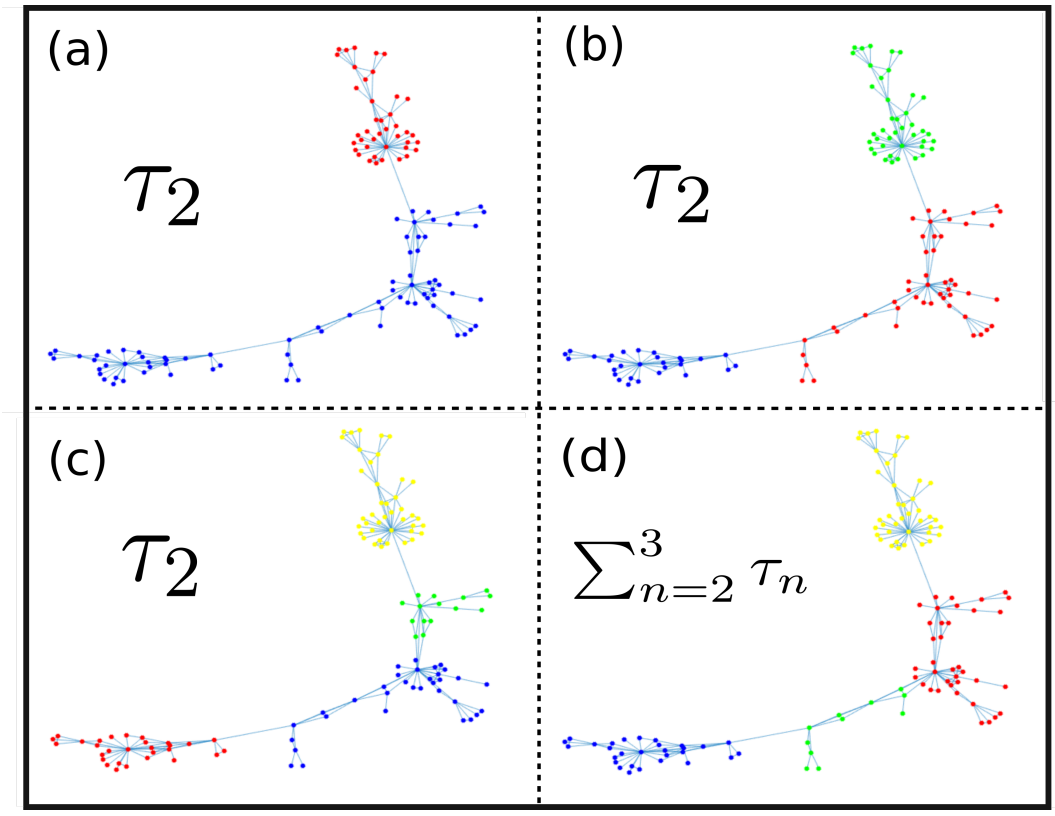


Figure 4.3: Multistate clustering of Santa Fe research network.

Next in figures (c) and (d) we cluster in to four states with $\tau_2$ and $\sum_{n=2}^{3} \tau_n$ re-

spectively. As we expect from the examples we investigated in chapter 2, since we have identified the three metastable clusters, the four state clustering with $\tau_2$ places a transition state between the slow states found in (a) and similarly the clustering with $\sum_{n=2}^{3} \tau_n$ places a transition state between the two states distinguished in (b).

So using our kinetic based clustering, we can conclude that the Santa Fe collaboration network is well described by three metastable states with two transition state clusters.

## 4.7 Conclusions

In this chapter, we have extended our network clustering method to also work for geometric networks and demonstrated that our new approach is effective at identifying both metastable and transition state clusters. Additionally we have shown that using a parallel tempering inspired method for optimizing our variational parameter, we can make our method much more computationally efficient for large, multi-dimensional systems.

At this point in the thesis, it is useful to summarise what has come so far. The last three chapters (2-4) have all been built towards the goal of developing, justifying and refining our eigenvalue based clustering with the culmination of a highly efficient method for carrying this out. There is still significant work that can be done to further accelerate the speed at which the algorithm performs. While the PTVC method will find the optimal state quicker than an exhaustive search, it has a number of arbitrary parameters to be chosen (how many parallel simulations to run, how frequently to perform switches etc.). Similarly, at present the initialisation for the simulations is performed randomly while one could develop a method for making better initial guesses/predictions for the boundary positions (for example at local free energy maxima). It is our hope that this method can continue to evolve in to one that will further improve the state of the art methods in the field.

Having drawn this line in the sand, we move away from our variational clustering and onwards to two new and distinct pieces of work in chapters 5 and 6.

*"Prediction is very difficult, especially if it's about the future."*

Anonymous (Danish Proverb)

# 5

# Estimation of Relaxation Times from Markov Models

The goal of building an MSM is to describe a continuous dynamics by some humanly interpretable model consisting of discrete states with transition probabilities between them. As we've seen so far, this modeling will require an approximation to be made to the dynamics which usually introduces non-Markovian effects. These non-Markovian effects arise in essence due to the fact that, on short timescales, the system remembers how it entered its current state due to the discretization of time and space.

For example, in figure 5.1 we can see an example of how poor spatial-temporal resolution can influence the Markovianity of the model. The black potential represents the actual underlying energy profile governing the dynamics while the red dashed lines are the boundaries defining how the microstates have been discretized. Since there is a free energy barrier within the microstate then the model will retain memory of how it entered the state (i.e. the blue particle entering from the left will be more likely to exit left than to cross the barrier and exit right) if the lagtime of the model is shorter than the time required to cross the barrier inside the microstate. In other words, the combination of space and time discretization must be chosen such that the equilibration time within the states is shorter than the lag at which transitions are considered.

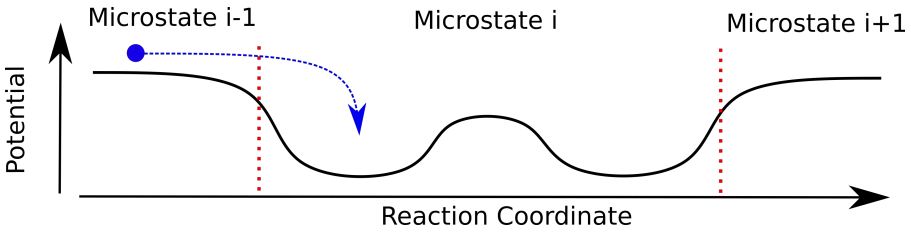


Figure 5.1: Illustration of memory effects in Markov model construction.

Typically one uses the CK test to check that the constructed propagator is insensitive to the choice of lagtime and moreover that the slowest relaxation time is insensitive to this choice.

$$T(\tau)^n P(t) = T(n\tau)P(t) \tag{5.1}$$

$$t_2(n\tau) = \frac{-n\tau}{\lambda_2(n\tau)} = \frac{-n\tau}{n\lambda_2(\tau)} = \frac{-\tau}{\lambda_2(\tau)} = t_2(\tau) \tag{5.2}$$

There are a number of practical and interesting cases where it may be difficult to construct a model which satisfies the CK test. For example, consider situations where the lagtime needed to satisfy the CK condition reduces the transition sampling too drastically to obtain meaningful statistics or even potentially when one is combining

a large number of short simulations to construct a single model the lagtime needed might be comparable to the simulation length[1].

The objective of this chapter is to derive a simple functional dependence between the relaxation time obtained from a constructed MSM at a particular lagtime and the 'true' relaxation time of the underlying continuous Markovian dynamics which are being approximated. This functional dependence will allow us to perform a best fit to the calculated data (relaxation time vs lagtime) and extract the relaxation time in the limit where the lagtime tends towards to infinity.

## 5.1 Lagtime dependence of Relaxation Times

We begin by deriving the functional dependence of the relaxation time on the lagtime. The only theory necessary is that of correlation functions which have been shown to be extremely useful for several different applications so far. In this instance, we will be using the spectral decomposition of the normalized connected correlator. As in chapter 2 (equation 2.3) we can write the correlation of two arbitrary functions $f$ and $g$ at a time difference $\tau$ using the Markov matrix $T$.

$$c(f,g,\tau) = \sum_{i,j} g(j) f(i) [\mathbf{T}(\tau)]_{ji} p_i^{eq} \tag{5.3}$$

Using the spectral decomposition of the propagator, one can see that the correlation function is given by projecting the variables of interest on to the right eigenvectors with the eigenvalues dictating the relative weighting at different times.

$$c(f,g,\tau) = \sum_{n=1}^{N} e^{\lambda_n \tau} (g \cdot \psi_n^R)(f \cdot \psi_n^R) \tag{5.4}$$

Extracting the long time limit ($(g \cdot \psi_1^R)(f \cdot \psi_1^R) = \langle g \rangle \langle f \rangle$) amounts to removing the first term from the sum while normalization requires division by the correlator at

[1]The lagtime cannot be longer than the length of the shortest simulation used to construct the MSM.

$\tau = 0$.

$$\bar{c}(f, g, \tau) = \frac{\sum_{n=2}^{N} e^{\lambda_n \tau} (g \cdot \psi_n^R)(f \cdot \psi_n^R)}{\sum_{n=2}^{N} (g \cdot \psi_n^R)(f \cdot \psi_n^R)} \tag{5.5}$$

The logic of this derivation is as follows:

- Non-Markovian effects arise from attempting to describe a continuous Markovian dynamics by dynamics on a finite set of discrete states.
- The local equilibrium model for coarse-graining links the correlation functions of the full and reduced system at some finite lagtime.
- By assuming the local equilibrium model to link the full and reduced system correlation functions and exploiting orthogonality of eigenvectors then the functional dependence of coarse-grained relaxation time on lagtime can be derived.

We write the correlation function at lagtime $\tau$ of the full continuous description from a propagator $\mathbf{T}(\tau)$ as in equation 5.6.

$$\bar{c}(f, g, \mathbf{T}(\tau)) = \frac{\sum_{n=2}^{\infty} e^{\lambda_n \tau} (g \cdot \psi_n^R)(f \cdot \psi_n^R)}{\sum_{n=2}^{N} (g \cdot \psi_n^R)(f \cdot \psi_n^R)} \tag{5.6}$$

Meanwhile the correlator from a coarse-grained Markov state model $\mathbf{T}^{MSM}$ constructed at lagtime $\tau$ is given by equation 5.7.

$$\bar{c}(f, g, \mathbf{T}^{MSM}(\tau)) = \frac{\sum_{n=2}^{\infty} e^{\lambda_n \tau} (g \cdot \psi_n^{R-MSM})(f \cdot \psi_n^{R-MSM})}{\sum_{n=2}^{N} (g \cdot \psi_n^{R-MSM})(f \cdot \psi_n^{R-MSM})} \tag{5.7}$$

In the previous two equations, we have kept the general expression for a correlator between two arbitrary functions $f$ and $g$. Next, we consider the concrete example where these are chosen to both be equal to the second left eigenvector of the discrete model, $\psi_2^{L-MSM}$. From orthogonality of left and right eigenvectors, one obtains that the correlation function in the discrete case collapses to equation 5.8.

$$\bar{c}(\psi_2^{L-MSM}, \psi_2^{L-MSM}, \mathbf{T}^{MSM}(\tau)) =$$
$$\frac{\sum_{n=2}^{\infty} e^{\lambda_n^{MSM}\tau}(\psi_2^{L-MSM} \cdot \psi_n^{R-MSM})(\psi_2^{L-MSM} \cdot \psi_n^{R-MSM})}{\sum_{n=2}^{N}(\psi_2^{L-MSM} \cdot \psi_n^{R-MSM})(\psi_2^{L-MSM} \cdot \psi_n^{R-MSM})} = e^{\lambda_2^{MSM}\tau} \quad (5.8)$$

Similarly we can examine the correlation function of $\hat{P}(\psi_2^{L-MSM})$, the projection of the second MSM eigenvector back on to the full dimensional space such that the $i \in I$th element of the $\hat{P}(\psi_2^{L-MSM})$ vector is equal with the corresponding coarse-grained element $\psi_2^{L-MSM}(I)$.

$$\bar{c}(\hat{P}(\psi_2^{L-MSM}), \hat{P}(\psi_2^{L-MSM}), \mathbf{T}(\tau)) =$$
$$\frac{\sum_{n=2}^{\infty} e^{\lambda_n\tau}(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^{R})(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^{R})}{\sum_{n=2}^{N}(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^{R})(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^{R})} = \sum_{n=2}^{\infty} A_n e^{\lambda_n\tau} \quad (5.9)$$

Where we have defined $A_i = \frac{(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_i^R)(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_i^R)}{\sum_{n=2}^{\infty}(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^R)(\hat{P}(\psi_2^{L-MSM}) \cdot \psi_n^R)}$. Next we will show that the assumption of local-equilibrium, allows the correlation functions of equations 5.8 and 5.9 to be equated. The local equilibrium condition amounts to equating correlation functions of indicator functions. Defining indicator functions $f_I(J)$ and $g_I(i)$,

$$\mathrm{f}_I(J) = \begin{cases} 1, \text{ if } J = \mathrm{I} \\ 0, \text{ if } J \neq \mathrm{I} \end{cases} \quad (5.10)$$

$$\mathrm{g}_I(i) = \begin{cases} 1, & \text{if } i \in I \\ 0, & \text{if } i \notin I \end{cases} \quad (5.11)$$

One can then express the local equilibrium condition in terms of spectral decom-

positions with these indicator functions as in equation 5.12.

$$\begin{aligned} T^{MSM}(I,J)P(J) &= \sum_{n=1}^{N} e^{\lambda_n^{MSM}\tau}\left(f_I(I)\cdot\psi_n^{R-MSM}\right)\left(\mathrm{f}_J(J)\cdot\psi_n^{R-MSM}\right) \\ &= \sum_{i\in I}\sum_{j\in J}\sum_{n=1}^{\infty} e^{\lambda_n\tau}\left(g_I(i)\cdot\psi_n^{R}\right)\left(g_J(j)\cdot\psi_n^{R}\right) \\ &= \sum_{i\in I}\sum_{j\in J} T^{full}(i,j)P(j) \end{aligned} \tag{5.12}$$

Using equation 5.12 and writing $\psi_2^{L-MSM}$ as a linear combination of weighted sum of $f_I$-s as basis vectors ($\sum_I \psi_2^{L-MSM}(I)g_I(i) = \hat{P}(\psi_2^{L-MSM})$), we can link equations 5.8 and 5.9.

$$\sum_{n=2}^{\infty} A_n e^{\lambda_n\tau} = e^{\lambda_2^{MSM}\tau} \tag{5.13}$$

If we assume that $\tau$ is sufficiently large that the $\lambda_2$ term dominates in the summation then we obtain equation 5.14[2].

$$e^{\lambda_2^{MSM}\tau} = \sum_{n=2}^{\infty} A_n e^{\lambda_n\tau} \approx A_2 e^{\lambda_2\tau} \tag{5.14}$$

$$\lambda_2^{MSM} = \lambda_2 + \frac{\varepsilon}{\tau} \tag{5.15}$$

Where $\epsilon = log(A_2)$. The relaxation time scales of the system are the inverse of the eigenvalues ($\mu_n^{relax} = \frac{1}{\lambda_n}$). This leads to the following equation, which describes the relaxation time, $\mu_2^{relax-MSM}$ as a function of lagtime $\tau$[3].

$$\mu_2^{relax-MSM} = \frac{\tau \times \mu_2^{relax}}{\tau + \varepsilon\mu_2^{relax}} \tag{5.16}$$

[2]Alternately, one could argue that we would expect that $A_2 >> A_{i>2}$ as $\hat{P}(\psi_2^{L-MSM})$ should be close to orthogonal to $\psi_{n>2}^R$ (and close to orthonormal to $\psi_2^R$).

[3]In this text so far, we have used a subscripted $\tau_n$ to describe the n-th lowest relaxation time and an unsubscripted $\tau$ to express the lagtime. To avoid any confusion in this chapter due to the frequency of use, we have adopted $\mu_n$ for the relaxation time.

Before we move on, let's break down exactly what this equation is saying. Equation 5.16 describes that if we construct a Markov state model at lagtime $\tau$ where the underlying dynamics is approximately two state and has a true relaxation time of $\mu_2^{relax}$ then the relaxation time of the Markov model will be $\mu_2^{relax-MSM}$. This is useful because now if we are unable to reach lagtimes where our relaxation time converges to satisfy the CK test then instead we can calculate $\mu_2^{relax-MSM}$ at different values of $\tau$ and perform a least squares fit to obtain the two free parameters $\epsilon$ and $\mu_2^{relax}$.

In the remainder of this chapter, we will test this method for extracting the long lagtime limit of relaxation times and compare it to the hidden Markov model approach.

## 5.2 Hidden Markov Models

For comparison, the approach derived above is contrasted with the results of using a hidden Markov model (HMM) formalism [132, 133]. This method has been applied in varied contexts [134–136] as it helps to describe a system where the observed distribution is non-stationary in time (i.e. the observed equilibrium probabilities change depending on the occupied hidden state).

The application of HMMs to modelling of molecular dynamics simulation data was first outlined in a publication by Noe et al [137]. The main idea of HMMs is that there exist some set of unobserved (hidden) states on which the dynamics of the system are Markovian. Then from these underlying hidden states $h_i$, at each observation time, the system will project onto one of our observed states $x_j$ with a given probability $E_{ij}$ as shown in figure 5.2.

Then given a set of observation data amongst the observed states, one can then construct a HMM that describes the dynamics amongst the hidden states and proceed to analyze the resulting hidden transition probability matrix as one would with a regular MSM.

However, it has been observed that the RTs obtained by implementing the HMM

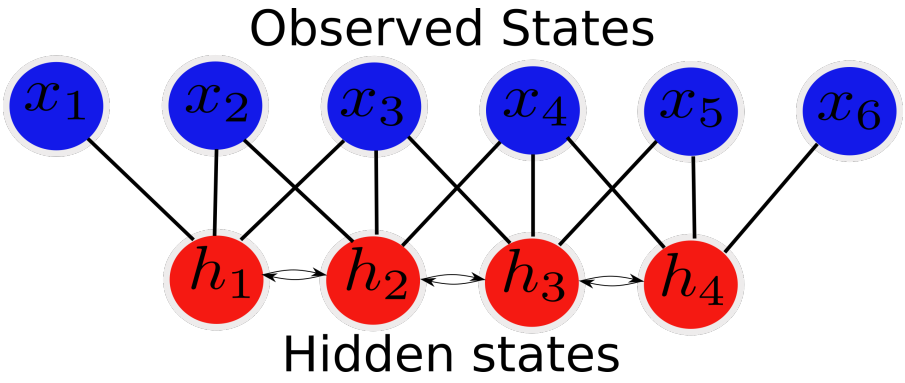


Figure 5.2: Illustration of Hidden Markov model architecture.

method do not generally follow any variational principle and may result in longer RTs than the true value. More importantly, while the RTs of HMMs tend to rapidly converge (i.e. at shorter lagtimes) for sufficiently large datasets, they do not follow the functional dependence on the lagtimes as MSMs do, as there is no corresponding theoretical description. Put another way, our derived fitting procedure is not applicable to HMM data as HMM do not display the same functional dependence as our fitting equation. As such, HMMs provide the best alternative approach with which to compare the results of fitting procedure.

## 5.3 Application to Test Systems

The derived equation for the slowest RT is tested on three different systems: (i) a series of MC trajectories generated in an unbiased analytic potential, (ii) unbiased MD simulations of pentalanine [118], and (iii) umbrella sampling simulations of an ion passing through a pentameric Gloeobacter ligand-gated ion channel (GLIC) [138, 139]. The results are compared to the RTs predicted by the HMM approach implemented in PyEMMA [140] for the unbiased cases. A series of Markov models are constructed at different lagtimes, and the values for the fitting parameters that minimize the error are calculated. The fitting parameters are obtained by doing least squares fitting over

the range of lagtimes shown in the figures.

### 5.3.1 Analytic Potential

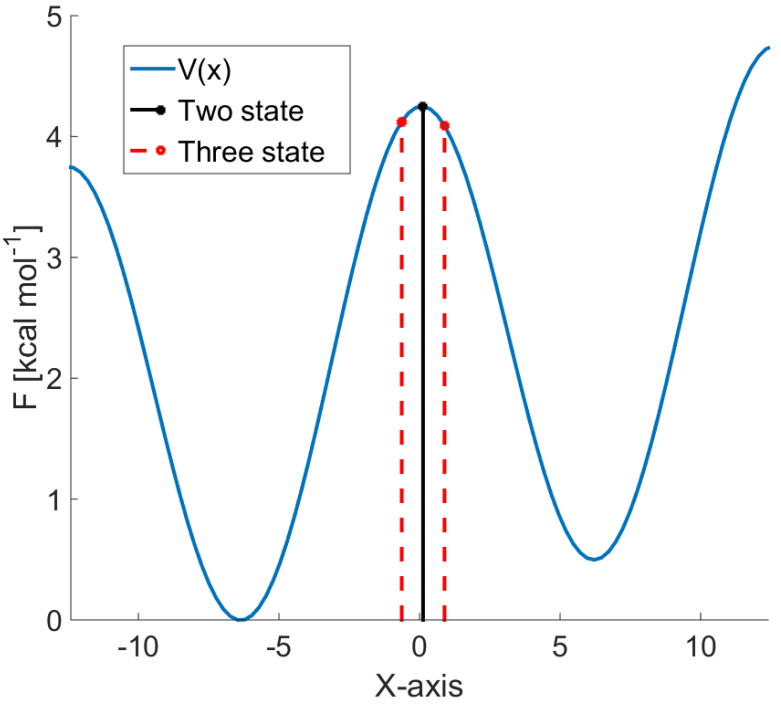


Figure 5.3: Analytic potential with cluster boundaries.

The first system we tested is an analytic potential given by equation 5.17 where $C_0$ is a number such that the minimum of the function in the domain $-4\pi \leq x \leq 4\pi$ is 0 (figure 5.3).

$$V(x) = -2\sin[(x-\pi)/2] + x/8\pi + C_0 \tag{5.17}$$

The systems dynamics are constrained within this domain. We identified the elements of the associated rate matrix K by discretizing the x-axis into 100 bins and using an Arrhenius-like expression of $K_{ij} = Ae^{-\beta(V(j)-V(i))/2}$ to calculate the transition rates (with $A = 2.5s^{-1}$), where our analytic potential is given by the function

$V(x)$. By using an artificial potential we can compare our calculated values to the exact relaxation time of the system.

After generating the rate matrix for this system, we created and analysed trajectories in two distinct manners, i) randomly initialised simulations and ii) downhill trajectories initialised within the transition state.

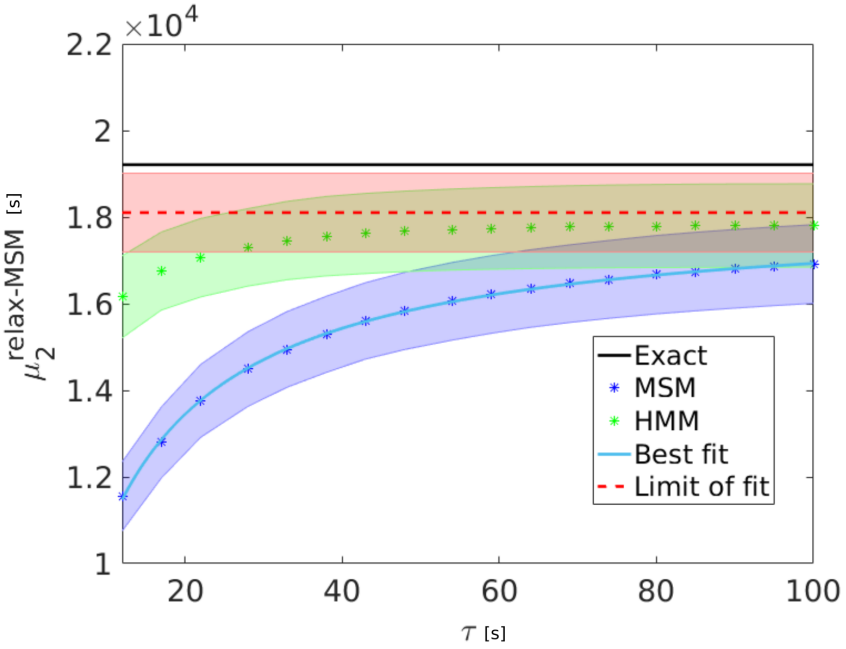


Figure 5.4: Fit to analytic potential with half relaxation time length.

For the first approach we used a Markov chain propagation method[4] to generate 100 simulations of length 40,000 (roughly twice the length of the system relaxation time). We repeated this process 10 times to provide error bars. With this set of simulations we clustered the trajectory frames in to a two state description[5] and built both our MSM at different lagtimes and a two state HMM. We applied our fitting method to the calculated MSM relaxation times and extracted the long lagtime limit. The results of

[4]For comparison we also used the Gillespie algorithm [141] to check that both methods gave effectively the same results which they did. As such we have omitted the Gillespie algorithm results.

[5]This clustering step is necessary to ensure that the trajectory used to generate the MSM is coarser than the underlying Markovian model used to generate the full dimensional (100 state) trajectory. If we just analysed the original trajectory we should just get back the exact relaxation time, independent of lagtime.

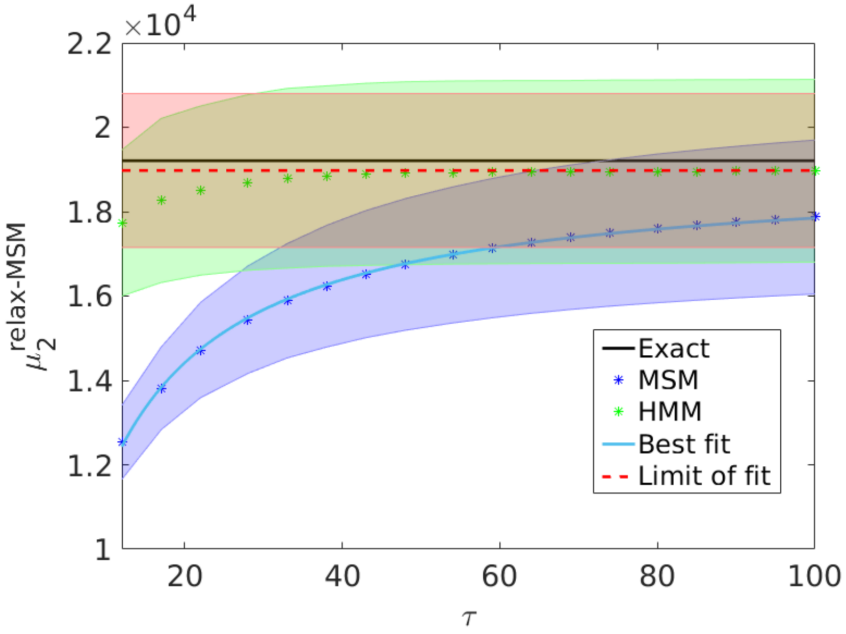


Figure 5.5: Fit to analytic potential with double relaxation time length.

this method are shown in figures 5.4 and 5.5 where we use simulations lengths of half and double the relaxation time respectively.

It can be seen in both of these figures that the HMM calculated relaxation times converge much faster than those of the MSM, however by using the relaxation time fitting procedure one can obtain a long lagtime relaxation time which reaches as close (or even closer) to the true value as the HMM. As expected, increasing the lengths of the simulations used pushes the quality of the estimations towards the true value.

In the second case, we instead generate downhill trajectories from within the transition state identified using the HS variational protocol (and shown by the dotted red lines in figure 5.3). For comparison we consider a three state MSM, and two and three state HMMs. The results of this are presented in figure 5.6. Interestingly, we observe that in this case the HMM rapidly converges (instantly in the three state case) but also overestimates the true relaxation time of the system. By contrast, the fitting procedure returns almost exactly the true value.

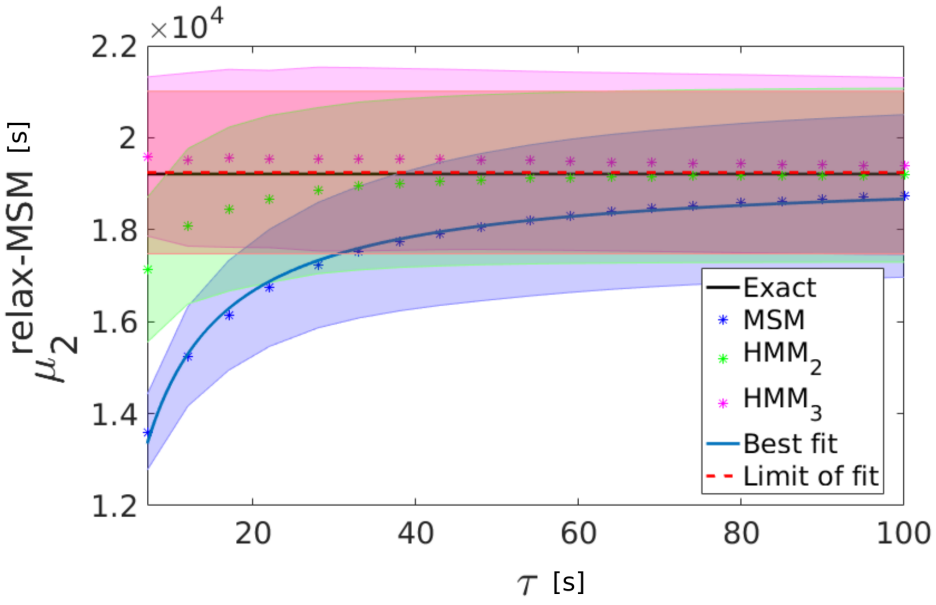


Figure 5.6: Relaxation time plot for analytic downhill trajectories.

### 5.3.2 Pentalanine MD Simulation

In this section, we apply our approach to an MD simulation of pentalanine (Ala5)[6]. Pentalanine is one of the most popular test systems for MSMs within the MD community. This is in part due to it being small enough to be simulated exhaustively while also exhibiting interesting dynamical behaviour in its helix-coil transition. The molecule is typically described in terms of its ten backbone dihedral angles (Ramachandran angles [142]).

The simulation trajectories analysed consist of four 250ns long independent unbiased MD simulations at different initial conditions with frames saved every 1ps. The errors on our calculations were obtained by performing the analysis for each of the four simulations individually. For conciseness, we present the lagtime fitting figure for $\psi_3$ in figure and summarise the results for the other angles in table 5.1.

[6]This simulation data was provided by our collaborators, further details can be found in the associated publication [118]

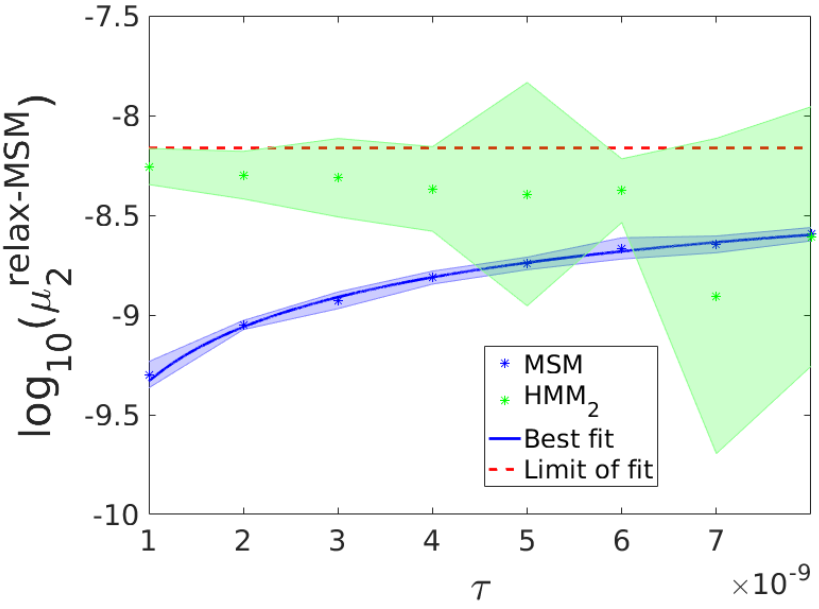


Figure 5.7: $\psi_3$ relaxation time plot

After analysing each angle we found that the relaxation times had not converged even at the longest accessible lagtimes. By performing our best fit we find that the long lagtime limit finds a relaxation time that is significantly closer to the 6–7ns value obtained by an analysis that simultaneously considers all angles and uses a transition-based state assignment [118].

Interestingly the HMM approach finds values which match closely to the 6-7ns value even at the smallest lagtimes considered. However it appears that the HMM description breaks down at shorter lagtimes than the MSM. Examining the results for all the angles in table 5.1 we see that all ten of the angles exhibit similar limiting relaxation times but there are two distinct groupings of $\epsilon$ values for the $\psi$ and $\phi$ angles.

From examining our derived equation, it is possible to show that the $\epsilon$ parameter corresponds to the initial slope of the data (i.e. the derivative of relaxation time with respect to lagtime, evaluated at a lagtime of zero). The epsilon parameter then has the interpretation of the speed with which the curve converges (with smaller values indicat-

ing quicker convergence). This provides a rough measure for determining the quality of a reaction coordinate because for a perfect reaction coordinate one would expect that $\epsilon$ would vanish to zero and the coordinate would exhibit no lagtime dependence.

| Coordinate | lagtime=1 | lagtime=1000 | $\epsilon$ | Limiting RT |
|---|---|---|---|---|
| 1 $\phi_1$ | 6.5 | 516.1 | 1.81 | 6976.3 |
| 2 $\psi_1$ | 952.2 | 2700.7 | 0.23 | 4711.3 |
| 3 $\phi_2$ | 25.5 | 567.7 | 1.75 | 6042.0 |
| 4 $\psi_2$ | 687.2 | 3353.6 | 0.17 | 6571.1 |
| 5 $\phi_3$ | 33.9 | 515.8 | 2.01 | 6875.1 |
| 6 $\psi_3$ | 653.2 | 2813.0 | 0.22 | 5101.8 |
| 7 $\phi_4$ | 65.8 | 424.7 | 2.47 | 9421.1 |
| 8 $\psi_4$ | 490.0 | 1929.3 | 0.47 | 5325.4 |
| 9 $\phi_5$ | 27.1 | 302.9 | 3.43 | 11303.5 |
| 10 $\psi_5$ | 189.5 | 740.5 | 1.06 | 5594.0 |

Table 5.1: Relaxation times (in ps) and $\epsilon$ parameters for pentalanine angles.

Examining the data in Table 5.1, we can see that $\phi_5$ has both the largest $\epsilon$ value and the most distinct value for its limiting relaxation time. The slow convergence indicated by the large $\epsilon$ value might help to explain why our value is so different.

### 5.3.3 Biased GLIC MD Simulation

The final example we consider is a biased simulation of an ion passing through a GLIC channel (figure 5.8). This is a particularly interesting case for us as it is the situation where we believe our derived algorithm will potentially be of greatest value. In this case, we cannot reach lagtimes for our CK plot to converge and also we cannot construct an HMM since the trajectory is biased.

The trajectory data was generated by a series of umbrella sampling Hamiltonian replica exchange simulations [143–145] where the exchange steps were attempted every 200fs [7]. Since the parallel trajectories are exchanged every 200fs, this is the longest lagtime at which one can construct an MSM. However, constructing at the length of

[7] These simulations were performed by collaborators and further simulation details can be found in the associated publications [138, 139]

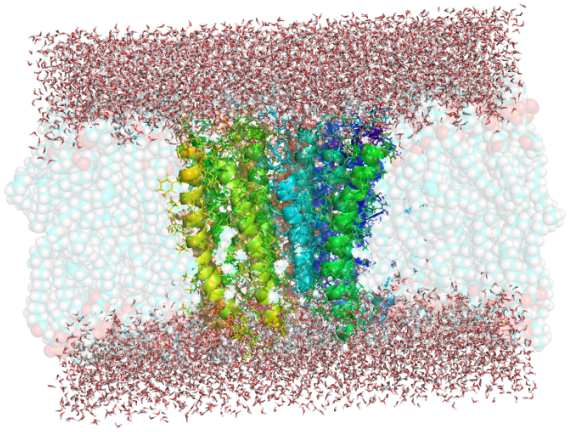

Figure 5.8: GLIC Ion channel

the individual simulations will hugely reduce the number of transitions and comprimise the statistical sampling. As such we constructed our MSM at lagtimes ranging from 1 to 100fs.

By constructing the MSM at the longest accessible lagtime of 100fs, we found the relaxation time to be $4.08 \times 10^8$fs. This is more than an order of magnitude less that the experimentally observed value of $6.25 \times 10^9$fs. But if we use our derived fitting procedure then we can find a value of $4.09 \times 10^9$fs, which is much closer to the true value (and is likely within the margins of error for such techniques).

## 5.4 Conclusions

In this chapter we have derived a method for improving the estimation of relaxation times in the long lagtime limit and compared its performance to the HMM method. Our method proved effective but it comes with a few caveats which we consider here.

Firstly, there is significant ambiguity regarding the choice of lagtimes to fit to in terms of both range and density. We typically found that it was best to avoid fitting to

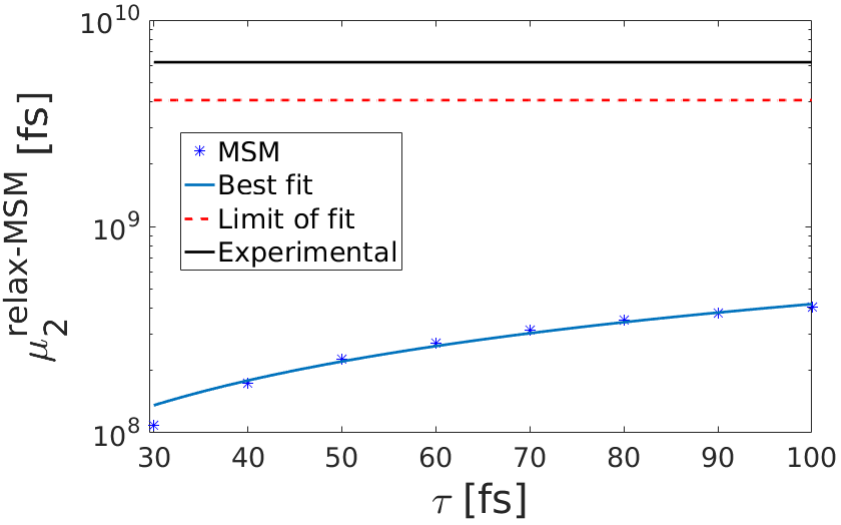


Figure 5.9: Relaxation time plot for replica exchange ion channel simulation.

small lagtimes where the two state description was likely invalid and also avoid large lagtimes where the noise due to reduced sampling was greater.

Regarding the number of points to use for fitting, we found that using a large number of points (typically above 100) caused the relaxation time to fluctuate due to the fitting data not exactly matching the derived functional dependency. To this end, we generally searched for a combination of range and density such that our fitting parameters were not sensitive to variations in these choices.

By examining the analytic potential and pentalanine examples we can see that the fitting procedure performs well but not significantly better than the HMM approach. However for the case where we have biased simulation data and cannot reach useful timescales due to replica exchange then our fitting procedure offers a useful tool for obtaining a more accurate comparison for validation to experiment. Though this procedure requires some decision making on the part of the user, it has a clear domain of problems where it proves useful for making quantitative predictions from limited data sets.

# 6

# Kinetic Analysis of Membranes with Markov Modelling

## 6.1 Introduction

In this penultimate chapter, we examine a further application of the Markov modelling theory described previously. In particular, we focus on the use of MSM theory to improve existing methods for calculating the permeability of cell membranes to different drug molecules. The primary benefit of MD simulation is the ability to give atomistic

level insight in to kinetic processes. The experimentally challenging procedure of determining cell membrane crossing rates is then well suited to MD study. Calculating the membrane permeability is of particular interest to pharmaceutical companies who are keen to quickly and accurately quantify the properties of potential drug candidates.

In an earlier study by Dickson and others at Novartis [1] , it was shown that long MD simulations can produce accurate estimates for the membrane permeabilities which match well with experimental values [146]. However these results required extensive computational resources and a complex analytic procedure. In this chapter we show that using biased simulations to construct a MSM, one can obtain equivalently accurate membrane permeabilities by a much simpler methodology[1].

In particular, we analyse the same seven structurally distinct drugs from the original studies by Eyer and Dickson [1,146] and show how to conveniently calculate the kinetic rates to transition in to, out of and across the cell membrane. As shown in figure 6.1 (reproduced from original computational publication by Dickson et al.) we consider a segment of a cell membrane with water on both sides. We demonstrate that drug permeabilities can be accurately and conveniently computed from MSM relaxation times. Furthermore, we show that generally a variety of kinetic properties linked to the crossing of the membrane (free energy barrier, crossing rates) are accurate indicators for the relative drug permeability.

## 6.2 Methodology

### 6.2.1 Existing Approach

The methodology implemented by Dickson et al. [1] used the following steps.

- Set up and run an MD simulation containing a drug molecule and a portion of a cell membrane (figure 6.1).

[1]For clarity of contribution, the biased simulations were performed by collaborators, the author was responsible for the processing and unbiasing of the data, construction of the Markov model and the development of some new simple equations for computing membrane permeability from an MSM.

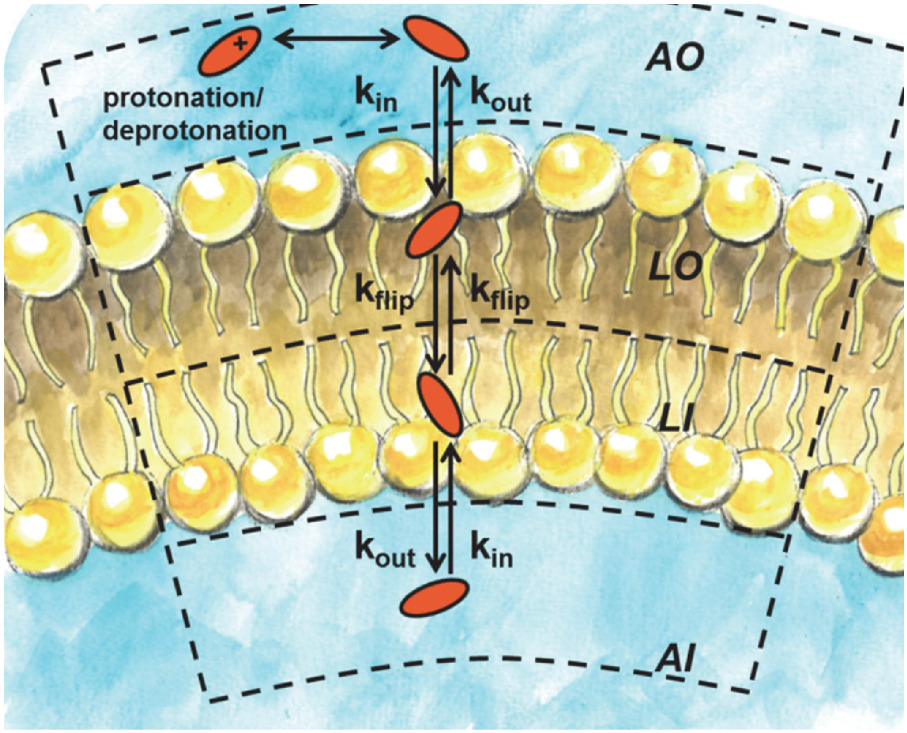


Figure 6.1: Image of drug-membrane system [1]

- Construct an MSM from the data, using the distance from the center of the membrane as a reaction coordinate.
- Use PCCA+ to cluster in to four regions (outer-water, outer-membrane, inner-membrane, inner-water).
- Calculate the new rate constants between the four clustered regions.
- Perform a Monte Carlo simulation of the constructed rates and fit the decay of an initial probability vector (all probability in state 1) to a biexponential function to get the equilibration rates.
- Compute the membrane permeability using the fastest equilibration rate via equation 6.1.

The equation used to calculate the permeability is given below in equation 6.1 where

$P$ is the permeability, $k_a$ is the rate constant of the fastest phase and $r$ is the radius of the experimental cell (assuming a spherical cell) [1].

$$P = \frac{k_a r}{3} \tag{6.1}$$

This approach uses a complicated approach to achieve a dynamic quantity which describes the time with which the drug permeates the cell membrane. In our alternate approach we provide a computationally and conceptually simpler method for obtaining a quantity which can be used to replace $k_a$.

### 6.2.2 Markov Model Inspired Approaches

In our alternative approach [147], we have two main points to make, firstly that biased simulations can greatly reduce the amount of sampling required for accurate kinetics and secondly that there are a number of different kinetic parameters obtainable directly from the MSM which serve as good proxies for the membrane crossing rate $k_a$ used in the existing studies.

- Perform an initial pulling MD simulation where a force is applied to the drug molecule to drag it cross the drug membrane.
- Use the frames of this initial trajectory to initialise and run a series of umbrella sampling potentials at different positions along the reaction coordinate.
- Using the dynamic histogram analysis method (DHAM) [148], construct a MSM from the biased simulation data.
- Use the slowest relaxation time $\tau_2$ directly from the MSM to obtain the permeability $P$ by taking $\tau_2 \approx 1/k_a$.

## 6.3 Theory

Most of the theory necessary for this chapter has been covered in previous chapters but we do need to introduce the topics of biased simulation and how to unbias them.

### 6.3.1 Biased Simulation

The details of MD simulation were laid out in section 1.3 and drew attention to the fact that a simulation generates dynamics by assuming some potential energy model for the interactions between the atoms of the system. One hopes that using such a model will generate accurate dynamics however for many realistic systems the 'interesting' processes happen on the order of milliseconds (or even seconds) whilst the numerical integrator (central differences etc.) requires a timestep $\delta t$ on the order of femtoseconds to be accurate.

The magnitude of these time differences means that it often computationally unfeasible to simulate systems for long enough to obtain sufficient sampling of the rare events. To circumvent this timescale issue, a variety of enhanced sampling techniques have been developed [149–153]. Most of these techniques work by applying an additional term to the potential energy equation used to generate the dynamics, these methods either add a term to move the system away from regions of configuration space that it has already sampled or add a term to constrain the system and ensure a particular region of space is better sampled.

The biasing method employed here will be umbrella sampling (US) [154, 155]. US requires a reaction coordinate of interest to be chosen. One then generates a series of configurations along the reaction coordinate from which to initialise simulations. Next a series of simulations are started where the potential energy function is given an additional harmonic constraint $U_i$ as in equation 6.2 where $k_i$ is the restraining force and $x_i$ is the value of the reaction coordinate which the umbrella simulation is

constrained to be remain close to.

$$U_i(x) = k_i(x - x_i)^2 \tag{6.2}$$

By spacing these simulations appropriately, one obtains an overlapping series of simulations such that the reaction coordinate is (approximately) uniformly sampled. From this sampling of the reaction coordinate one then needs to retrospectively account for this biasing when constructing the Markov model for the underlying system. The most well known approaches for this are the weighted histogram analysis method (WHAM) and the dynamic histogram analysis method (DHAM) which we introduce now.

### 6.3.2 Unbiasing Methods

WHAM is a method for recovering free energies from umbrella sampling simulations [156–158] and has allowed umbrella sampling to grow in to one of the most widely used enhanced sampling methods [159–161]. The logic follows that the probability of observing a particular state during a simulation will be perturbed by a predictable amount during an US simulation since we know the term which has been added to the potential energy[2].

By examining the number of observations of each state in each simulation, one can use knowledge of the biasing term to obtain the unbiased probabilities via the iterative equations 6.3 and 6.4.

$$p_j = \frac{\sum_{k=1}^{M_{sim}} n_j^{(k)}}{\sum_{k=1}^{M_{sim}} N^{(k)} f^{(k)} e^{-u_j^{(k)}/k_B T}} \tag{6.3}$$

$$f^{(k)} = \frac{1}{\sum_{l=1}^{N_{bin}} e^{-u_l^{(k)}/k_B T} p_l} \tag{6.4}$$

[2]WHAM relies on the assumption that the observed states of the system have all been sampled with equilibrium probability.

While WHAM has been extremely popular for reproducing free energy profiles, it has the drawback that it does not make use of the kinetic information from the simulation.

In this sense, one could take the observations of an MD simulation trajectory, shuffle them (like a deck of cards) and WHAM will reproduce the same free energy profile as it cares only about the relative frequency of observations and not the ordering in time. To address this issue the transition reweighted analysis method (TRAM) [162] was developed to use the transition count information to improve the free energy estimation.

TRAM however only returned free energies and not any kinetic information. DHAM [148] was developed as an alternative method to incorporate the kinetic information and produce not a free energy profile but a full MSM from the US data. By using a maximum likelihood approach one can derive the most likely MSM given the biased observations. This results in equation 6.5 where $C_{ji}^{(k)}$ is the number of transition counts between states $i$ and $j$ in simulation $k$, $n_i^{(k)}$ is the total number of counts in state $i$ in simulation $k$ and $V_j^k$ is the potential applied to state $j$ in simulation $k$.

$$M_{ji} = \frac{\sum_{k=1}^{M_{sim}} C_{ji}^{(k)}}{\sum_{k=1}^{M_{sim}} n_i^{(k)} e^{-(V_j^k - V_i^k)/k_B T}} \tag{6.5}$$

Further refinements of these methods have been developed, namely dTRAM [163] which extends TRAM to estimate a MSM and DHAMed [164] which extends the DHAM method to enforce detailed balance. We will use DHAM for our analysis as it allows us to construct an MSM automatically from the umbrella sampling data from which we can extract interesting kinetic information (relaxation time and mean first passage times).

## 6.4 MSM Analysis of Membrane Simulations

In this section we present the following results of applying the aforementioned procedure to the umbrella sampling simulations.

- We perform the CK test to obtain the lagtime at which to construct our MSM and also use our limiting method (from chapter 5) to extract the long lagtime relaxation time.
- We compare the free energy profiles obtained from the umbrella sampling to those originally obtained by Dickson et al. with unbiased simulation.
- We compute the permeability via a number of kinetic parameters and compare and contrast the effectiveness with the original method.

During the analysis, it was observed that the profiles obtained were slightly asymmetric. This arose due to the umbrella window at the very peak of the free energy barrier (center of the membrane) sampling one side of the barrier more than the other. To compensate for this we assumed that since the cell membrane was prepared to be symmetric that we can reflect the observations in the central umbrella window to achieve symmetry at the barrier peak. This highlights one of the issues with traditional US, it often operates by using equidistant umbrella positions and equal constraining forces. In reality one likely requires the umbrella parameters to vary depending on the local curvature of the free energy potential.

### 6.4.1 Chapman-Kolmogorow Test

The relaxation time, $\tau_2$, for each drug molecule was determined by constructing MSMs at a range of lagtimes up to 300ps with 1000 bins, as shown in Figure 6.2. Using our method for calculating the limiting relaxation time of an MSM, we determined the long lagtime limit of the relaxation time for each drug, as shown by the dashed lines in Figure Y. The relaxation times can be seen to level off in the region of lagtimes greater

than 100 ps. In the analysis that follows, we chose to use a lagtime of 200 ps for MSM construction, as it is sufficiently large for $\tau_2$ to be insensitive to the precise choice of the lagtime. Following this initial choice of parameters, seven Markov models (one for each drug molecule) were constructed with 1000 bins and a lagtime of 200ps (100,000 simulation steps).

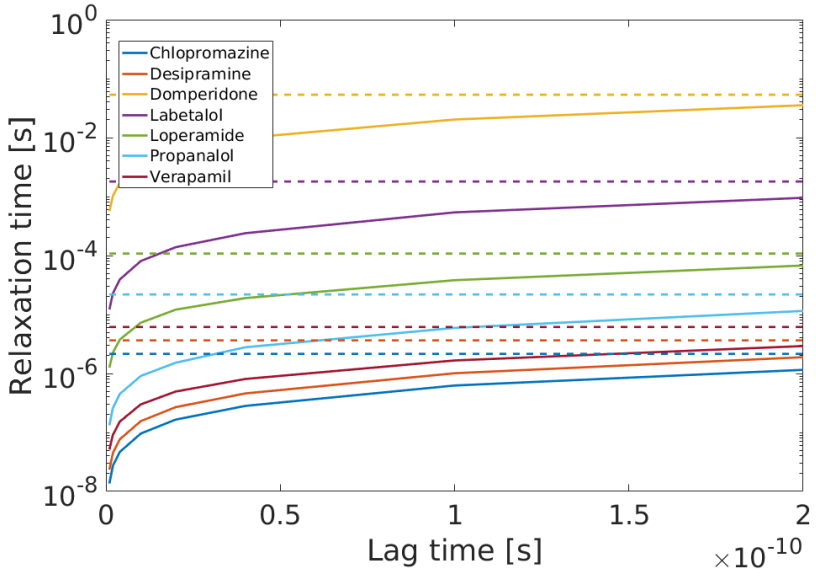


Figure 6.2: Fitted relaxation times for drug molecules

### 6.4.2 Free Energy Profiles

Following MSM construction (using DHAM) we can compute the free energy profiles for each drug and draw comparison with the profiles obtained by Dickson et al. in the unbiased simulations, using WHAM (figures 6.3 and 6.4).

Error bars were determined by dividing the data into two equal sections, determining the profiles independently, and calculating the variance. All of our free energy profiles show the same trend as the one calculated by Dickson et al. (dotted lines) for the combined unbiased and biased MD data, and indeed, all of the WHAM predictions are within the error of the DHAM free energies.

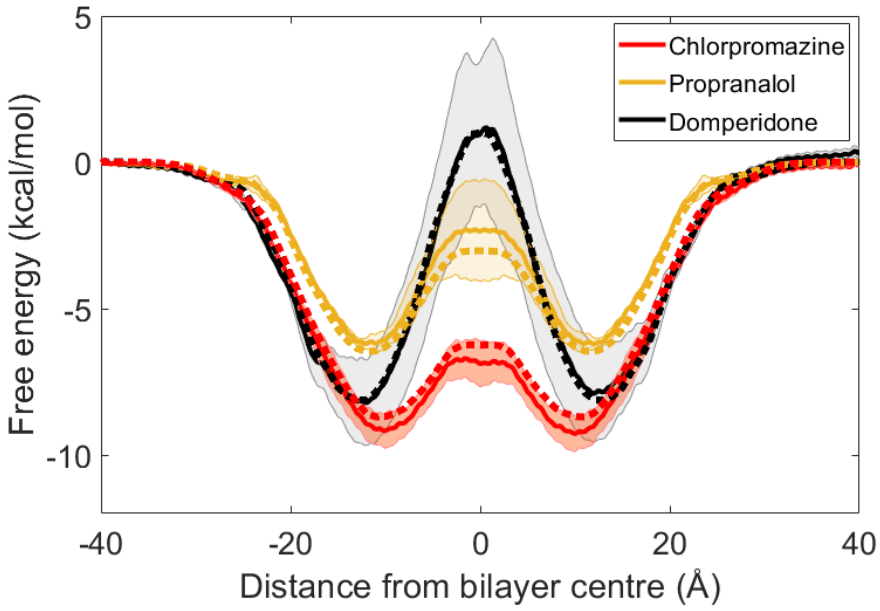


Figure 6.3: DHAM free energies

We found that in some of the simulation data, the central US window at z = 0Å is not sampled uniformly and so the resulting potentials become highly asymmetric. To address this, we reflect the observed trajectory in the central window to symmetrize. This is reasonable if we assume that the cell membrane should behave symmetrically.

While the PMF changes depending on whether the US window at z = 0Å was reflected or not, the permeability data are essentially unchanged. The asymmetry observed in the not fully reflected PMF profiles also suggests that greater sampling of this region may be necessary to accurately estimate crossing times (or that the strength of the umbrella potential might be varied to be more restrictive).

The free energy profiles obtained from the umbrella sampling data accurately reproduce the results of Dickson et al. while using a much smaller amount of computational resources. We used 3.2$\mu$s of simulation for each drug molecule while the work by Dickson et al. used 12.5$\mu$s. This represents a reduction of approximately 75% in the computational intensity of the study.

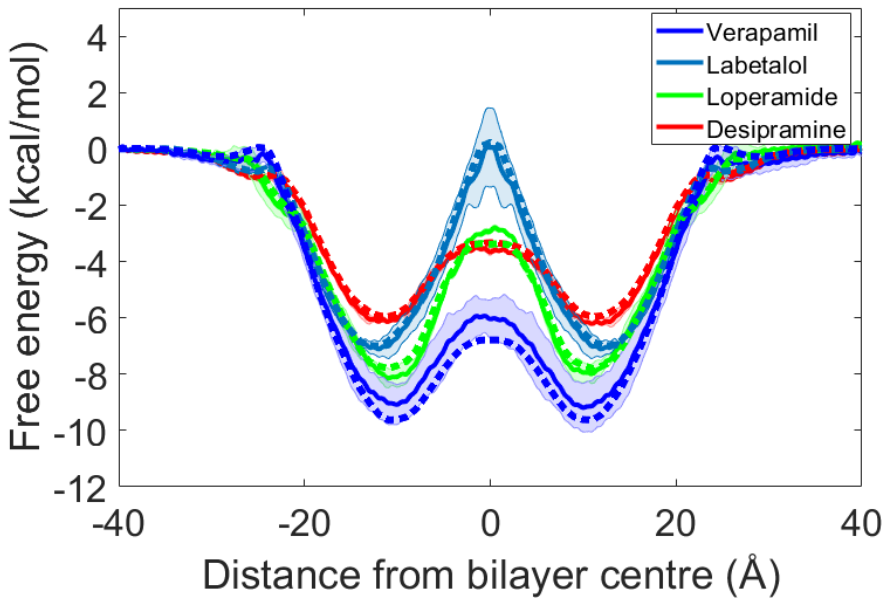


Figure 6.4: DHAM free energies

## 6.5 Permeability Ordering

The ultimate goal of this project was to show that we could compute the relative permeability quickly and conveniently from a biased simulation. We've shown already that biased simulations can be used to get accurate free energy profiles. We now want to show how a variety of kinetic properties can be used to order drug compounds by relative permeability.

As described previously, to obtain the relative permeability, one must compute the timescale of the process corresponding to the crossing of the free energy barrier at the center of the membrane among the different drug molecules. Here, we considered several ways to estimate the relative ordering directly using the kinetics from a MSM (Table 6.1).

We will compare seven log P quantities in total, one experimental value and three values calculated each for the unbiased and biased simulation.

The three kinetic values we use are i) the slowest relaxation time of the full dimen-

sional model $logP(\tau_2)$ ii) using PCCA+ to recover the four regions used by Dickson et al. and calculating the MFPT between the regions on either side of the barrier $logP(k_f)$ and iii) using knowledge of the crossing rate and barrier height to estimate the Arrhenius rate constant $logP(A)$. These approaches are simple to implement compared with simulating a kinetic system from the calculated rates and performing a bi-exponential fit to the resultant time-dependent probabilities.

Computing these quantities, we obtain a distinct range of values. The original $logP$ values obtained by Dickson et al. most closely resemble the experimentally determined values. There are a number of caveats to this though. While the values from our proposed MSM based method less closely resemble the experimental, they provide higher $R^2$ values. The $R^2$ values describe the extent to which two sets of numbers can be mapped to each other via a linear transformation.

In many practical applications, it is this high $R^2$ that we are more interested in. In other words, we want to predict not necessarily the absolute permeability but the relative magnitude to a high degree of certainty. Additionally there is the factor that the MSMs for the unbiased case were constructed at a relatively short lagtime whereas the relaxation time used for the biased $logP(\tau_2)$ was the value extracted from the long lagtime limit fitting procedure. Varying the lagtime will fluctuate the absolute value of the relaxation time but not the relative ordering of the drugs.

| | | Unbiased | | | Biased | | |
|---|---|---|---|---|---|---|---|
| Drug Name | $logP_{exp}$ | $logP$ | $logP(k_f)$ | $logP_A$ | $logP(\tau_2)$ | $logP(k_f)$ | $logP_A$ |
| Domperidone | -2.6 | $-2.65 \pm 0.11$ | -2.93 | 0.663 | $-4.04 \pm 0.95$ | -4.15 | 0.532 |
| Labetalol | -2.1 | $-1.2 \pm 0.36$ | -1.36 | 2.051 | $-2.46 \pm 0.82$ | -2.57 | 2.107 |
| Loperamide | -0.42 | $0.11 \pm 0.09$ | 0.35 | 4.105 | $-1.31 \pm 0.17$ | -1.38 | 3.43 |
| Verapamil | 0.01 | $0.09 \pm 0.05$ | 1.20 | 5.246 | $0.05 \pm 0.01$ | -0.58 | 5.05 |
| Propanalol | 0.19 | $0.51 \pm 0.06$ | 1.39 | 4.814 | $-0.54 \pm 0.88$ | 0.04 | 4.52 |
| Chlopromazine | 0.59 | $0.85 \pm 0.13$ | 1.76 | 5.531 | $0.46 \pm 0.07$ | 0.42 | 5.52 |
| Desipramine | 0.65 | $0.7 \pm 0.01$ | 1.56 | 5.402 | $0.24 \pm 0.01$ | 0.21 | 5.42 |
| $R^2$ | 1 | 0.924 | 0.965 | 0.984 | 0.928 | 0.966 | 0.943 |

Table 6.1: Comparison of Log P values obtained by different methods.

Comparing directly the unbiased $logP$ and the biased $logP(\tau_2)$ in figure 6.5, we can observe that the two sets of numbers correlate well with the experimental value and provide strong measures for ordering drug molecules according to permeability.

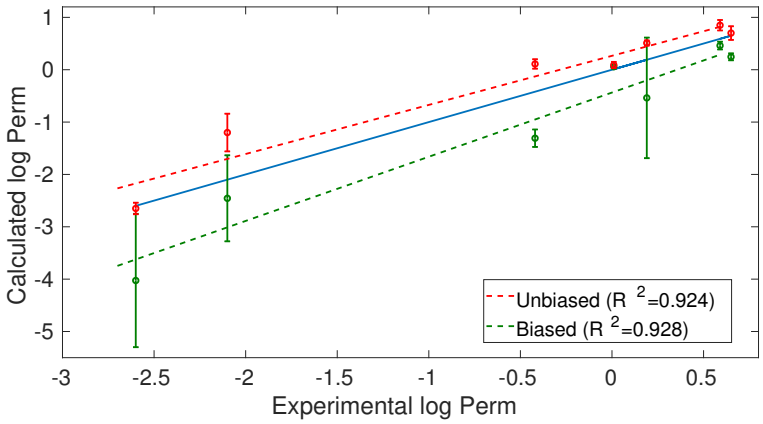


Figure 6.5: Comparison to experimental Log P values

## 6.6 Conclusions

In this chapter, we applied our lagtime fitting procedure to a membrane crossing system, derived a simpler protocol for estimating permeabilities and more generally by examining the kinetic quantities obtained from the full MSM, we managed to provide highly accurate orderings of the seven drug molecules according to their membrane permeability. In general, we found that using any sensible kinetic property provided a good measure for this ordering.

Given the accuracy of the newly proposed protocol, this means that the challenge of efficiently calculating membrane permeabilities from molecular simulation becomes a question of obtaining accurate MSMs. This allows for the use of US to accelerate the sampling. However there is the potential for even greater acceleration as the data studied here was generated by a series of uniformly spaced, equally shaped umbrella potentials. It is likely that an adaptive US procedure could obtain accurate values even

quicker.

*"Well, I've narrowed it down to two possibilities: yes and no."*

Chidi Anagonye, The Good Place

*"Everything I've ever let go of has claw marks on it."*

David Foster Wallace, Infinite Jest

# 7

# Conclusions and Outlook

The goal with the research undertaken was to examine the state of the art techniques being used for Markov modeling throughout science, with a particular focus on the applications to biophysical molecular dynamics. The ultimate prize being a protocol which identifies the key regions clusters of interest in the model while removing the need for arbitrary and uncertain decision making on the part of the user. To this end we conclude this thesis with some comments and opinions on the latest progress in the field and what the future may hold for Markov state modeling and the wider molecular dynamics community.

## 7.1 Automated Construction of Markov state models

The question driving much MSM research over the past few years has been 'how can we remove the user, who may well not be an MSM expert, from needing to make decisions in the construction of the MSM?'. As a means to achieving this goal, many research groups have leveraged machine learning techniques to automate the key problematic procedures such as feature selection [73,165–169] or microstate clustering [96,170] and create single step end-to-end pipelines [171,172].

It is this authors opinion that the concentrated goal of removing the user from the construction process misses the point to some degree. While automating certain tricky decisions for non-experts in such a way that minimizes errors is clearly desirable, this should never come at the expense of the usefulness of the final result. Interpreting the result of any mathematical pipeline (whether the pipeline is automated or not) requires the user to have at the forefront of their mind the precise question the pipeline is designed to answer, as well as any assumptions which are built in its approach.

It is upon this philosophy which we have tried to build the protocols outlined in this thesis. With biophysical simulations, the behaviour we are interested in is always what is the most 'interesting'. This may be the slow dynamics, it may not. More useful than a fully automatic pipeline is a flexible tool which i) makes very clear the principles upon which it is built (and the assumptions it requires) and ii) allows the user the opportunity to avoid non-intuitive parameter selection in favour of instead applying the intuition and expertise they do have for the wider problem at hand.

Therefore the methods developed within this thesis have aimed to be maximally clear and frank about any assumptions made and about the remaining parameter choices. In particular, in chapters 2-4 of this thesis we lay out our variational coarse-graining protocol which variationally searches for the clustering of states which maximises the relaxation timescales obtained from Hummer-Szabo coarse-graining. This

protocol relies on the assumptions that i) Hummer-Szabo is an appropriate coarse-graining method for the system at hand and ii) that timescale optimisation will distinguish the 'interesting' regions of the potential energy landscape. We have sought to provide strong reasoning why we would expect these assumptions to be valid in many systems but they still remain assumptions nonetheless.

In our variational method, we still retain a user input regarding which sum of timescales to use but have also attempted to both make it as straightforward as possible for an experienced scientist to incorporate their own knowledge and experience (by choosing the timescale describing the behaviour of interest) and at the same time recommend an automatic protocol for the more general situation.

## 7.2 The Outlook for Markov Models

We conclude by taking a step back from the research presented in this thesis and examining the current state of the wider field in general as well as speculating on the future of Markov state models as an analysis tool.

The final aim is to produce a toolkit and accompanying mathematical framework such that the end users, MD simulation experts, can provide actionable insight in to their systems and develop new pharmaceutical drugs and materials. To this end there are three main question marks over the use of MSMs which future research will need to address.

Firstly, so much of the current research in the field (our own included) has relied upon the assumption that the slowest dynamics are the most interesting to the user. It is easy to construct simple examples of systems where this assumption is violated and in fact the fast dynamics may be critical. Consider the case of a small drug molecule interacting with a large protein. Access to the binding pocket of the protein may be restricted by whether some fast moving side-chain is blocking the pocket or not. Now while the binding to the pocket is the slow process, it is actually highly influenced by

a fast dynamics. The question remains whether one can automatically build an MSM describing the interesting dynamics of a system or whether this will always require ad-hoc user knowledge. Or perhaps more importantly, can one create a framework which makes it easy for a non-expert user to integrate their prior domain-specific knowledge in a way which will return useful insights?

Secondly, MSMs have long be a source of optimism for developing enhanced sampling methods which are not only effective at accelerating sampling but are quantitatively optimizable according to some reasonable metric. A huge range of sampling techniques have been developed over the last twenty years, including some MSM inspired methods. Qualitative reasoning has usually been employed to argue for one method over another by claiming that for some example system the proposed method can identify the conformations of interest faster than some benchmark method. While this is clearly of value in helping practitioners understand which methods may be advantageous for their particular system, it doesn't allow for any clear, direct comparison between different methods.

This shortcoming arises in part due to the twin objectives of enhanced sampling techniques, the identification of new interesting states and the accelerated sampling of the observed configurations. In some systems, one is most interested in observing new states which were previously unknown (such as metastable protein misfoldings) while in other cases, one is more interested in quickly obtaining an accurate equilibrium sampling of known states (such as in estimating free energy barriers where accurate estimation requires extensive sampling of the lowest probability states). This trade-off between crossing barriers and obtaining accurate free energies is largely what prevents the emergence of a 'best' enhanced sampling algorithm as it does not allow for a single optimisable criterion by which to measure an algorithms performance.

Lastly, the primary advantage of the MSM framework is that it allows independent simulations of the same system to be combined in a statistically significant manner to construct a single model. However there are two natural extensions of this which are

yet to be produced, i) can we combine independent simulations of subdivisions of a system in to a single kinetic model or ii) if we have obtained an MSM for a particular system can we extract information about mutated versions of that system (e.g. the same protein with a single residue altered).

The questions raised above remain unanswered and of huge practical application. One hopes that progress can made on these in the years to come. Although variational optimization and automation has shown promise in recent years, progress on these particular frontiers may require a fundamental shift in approach and mind-set by replacing the goal of complete automation with the development of flexible frameworks which allow for the straight-forward combination of domain specific knowledge/objectives. In this authors opinion, this should be achievable through the mindful application of machine learning techniques to Markovian models.

With all these research possibilities, it is an ever exciting time to be working in kinetic research. The scope of potential systems to study is growing rapidly, as are the resources with which to study them and the corresponding theoretical frameworks need to adapt and evolve to facilitate this growth. The diversity of fields of research from which inspiration has been taken is far reaching, making this one of the most truly cross-disciplinary corners of science.

# Bibliography


[1] Callum J Dickson, Viktor Hornak, Robert A Pearlstein, and Jose S Duca. Structure–kinetic relationships of passive membrane permeation from multiscale modeling. *Journal of the American Chemical Society*, 139(1):442–452, 2016.

[2] Andrei Andreevich Markov. An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains. *Science in Context*, 19(4):591–600, 2006.

[3] Brian Hayes. First links in the markov chain, 2013. www.americanscientist.org.

[4] Ioannis Karatzas and Steven E Shreve. Brownian motion. In *Brownian Motion and Stochastic Calculus*, pages 47–127. Springer, 1998.

[5] Christopher M Turner, Richard Startz, and Charles R Nelson. A markov model of heteroskedasticity, risk, and learning in the stock market. *Journal of Financial Economics*, 25(1):3–22, 1989.

[6] Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using hidden markov model: a new approach. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 192–196. IEEE, 2005.

[7] Aditya Gupta and Bhuwan Dhingra. Stock market prediction using hidden markov models. In *2012 Students Conference on Engineering and Systems*, pages 1–4. IEEE, 2012.

[8] Xinyi Liu, Dimitris Margaritis, and Peiming Wang. Stock market volatility and equity returns: Evidence from a two-state markov-switching model with regressors. *Journal of Empirical Finance*, 19(4):483–496, 2012.

[9] Andrea Kölzsch, Erik Kleyheeg, Helmut Kruckenberg, Michael Kaatz, and Bernd Blasius. A periodic markov model to formalize animal migration on a network. *Royal Society Open Science*, 5(6):180438, 2018.

[10] Ben Dean, Robin Freeman, Holly Kirk, Kerry Leonard, Richard A Phillips, Chris M Perrins, and Tim Guilford. Behavioural mapping of a pelagic seabird: combining multiple sensors and a hidden markov model reveals the distribution of at-sea behaviour. *Journal of the Royal Society Interface*, 10(78):20120570, 2013.

[11] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *COMPUTER NETWORKS AND ISDN SYSTEMS*, pages 107–117, 1998.

[12] Amy N Langville and Carl D Meyer. Updating markov chains with an eye on google's pagerank. *SIAM journal on matrix analysis and applications*, 27(4):968–987, 2006.

[13] Robert Zwanzig. From classical dynamics to continuous time random walks. *Journal of Statistical Physics*, 30(2):255–262, 1983.

[14] B. Widom. Molecular transitions and chemical reaction rates. *Science*, 148(3677):1555–1560, 1965.

[15] B Widom. Reaction kinetics in stochastic models. *The Journal of Chemical Physics*, 55(1):44–52, 1971.

[16] Irwin Oppenheim, Kurt Egon Shuler, George Herbert Weiss, et al. *Stochastic processes in chemical physics*. 1977.

[17] Benjamin Fain. Theory of rate constants: Master equation approach. *Journal of Statistical Physics*, 25(3):475–489, 1981.

[18] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1. Elsevier, 1992.

[19] Hue Sun Chan and Ken A. Dill. Energy landscapes and the collapse dynamics of homopolymers. *The Journal of Chemical Physics*, 99(3):2116–2127, 1993.

[20] P E Leopold, M Montal, and J N Onuchic. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proceedings of the National Academy of Sciences*, 89(18):8721–8725, 1992.

[21] S Banu Ozkan, Ken A Dill, and Ivet Bahar. Computing the transition state populations in simple protein models. *Biopolymers: Original Research on Biomolecules*, 68(1):35–46, 2003.

[22] Ken A Dill and Hue Sun Chan. From levinthal to pathways to funnels. *Nature structural biology*, 4(1):10, 1997.

[23] Ken A Dill, S Banu Ozkan, M Scott Shell, and Thomas R Weikl. The protein folding problem. *Annu. Rev. Biophys.*, 37:289–316, 2008.

[24] Michael Shirts and Vijay S Pande. Screen savers of the world unite! *Science*, 290(5498):1903–1904, 2000.

[25] Vijay S Pande, Ian Baker, Jarrod Chapman, Sidney P Elmer, Siraj Khaliq, Stefan M Larson, Young Min Rhee, Michael R Shirts, Christopher D Snow, Eric J Sorin, et al. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers: Original Research on Biomolecules*, 68(1):91–109, 2003.

[26] Juekuan Yang, Yujuan Wang, and Yunfei Chen. Gpu accelerated molecular dynamics simulation of thermal conductivities. *Journal of Computational Physics*, 221(2):799–804, 2007.

[27] David E Shaw, Martin M Deneroff, Ron O Dror, Jeffrey S Kuskin, Richard H Larson, John K Salmon, Cliff Young, Brannon Batson, Kevin J Bowers, Jack C Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. *ACM SIGARCH Computer Architecture News*, 35(2):1–12, 2007.

[28] Richard H Larson, John K Salmon, Ron O Dror, Martin M Deneroff, Cliff Young, JP Grossman, Yibing Shan, John L Klepeis, and David E Shaw. High-throughput pairwise point interactions in anton, a specialized machine for molecular dynamics simulation. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, pages 331–342. IEEE, 2008.

[29] David E Shaw, Ron O Dror, John K Salmon, JP Grossman, Kenneth M Mackenzie, Joseph A Bank, Cliff Young, Martin M Deneroff, Brannon Batson, Kevin J Bowers, et al. Millisecond-scale molecular dynamics simulations on anton. In *Proceedings of the conference on high performance computing networking, storage and analysis*, page 39. ACM, 2009.

[30] David E Shaw, JP Grossman, Joseph A Bank, Brannon Batson, J Adam Butts, Jack C Chao, Martin M Deneroff, Ron O Dror, Amos Even, Christopher H Fenton, et al. Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 41–53. IEEE Press, 2014.

[31] William Humphrey, Andrew Dalke, and Klaus Schulten. Vmd: visual molecular dynamics. *Journal of molecular graphics*, 14(1):33–38, 1996.

[32] Kevin J Bowers, David E Chow, Huafeng Xu, Ron O Dror, Michael P Eastwood, Brent A Gregersen, John L Klepeis, Istvan Kolossvary, Mark A Moraes, Federico D Sacerdoti, et al. Scalable algorithms for molecular dynamics simulations on commodity clusters. In *SC'06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pages 43–43. IEEE, 2006.

[33] Robert T McGibbon, Kyle A Beauchamp, Matthew P Harrigan, Christoph Klein, Jason M Swails, Carlos X Hernández, Christian R Schwantes, Lee-Ping Wang, Thomas J Lane, and Vijay S Pande. Mdtraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophysical journal*, 109(8):1528–1532, 2015.

[34] Peter Eastman and Vijay Pande. Openmm: a hardware-independent framework for molecular simulations. *Computing in Science & Engineering*, 12(4):34–39, 2010.

[35] Nina Singhal, Christopher D Snow, and Vijay S Pande. Using path sampling to build better markovian state models: predicting the folding rate and mechanism of a tryptophan zipper beta hairpin. *The Journal of chemical physics*, 121(1):415–425, 2004.

[36] Michael Andrec, Anthony K Felts, Emilio Gallicchio, and Ronald M Levy. Protein folding pathways from replica exchange simulations and a kinetic network model. *Proceedings of the National Academy of Sciences*, 102(19):6801–6806, 2005.

[37] Sidney P Elmer, Sanghyun Park, and Vijay S Pande. Foldamer dynamics expressed via markov state models. i. explicit solvent molecular-dynamics simulations in acetonitrile, chloroform, methanol, and water. *The Journal of chemical physics*, 123(11):114902, 2005.

[38] William C Swope, Jed W Pitera, and Frank Suits. Describing protein folding kinetics by molecular dynamics simulations. 1. theory. *The Journal of Physical Chemistry B*, 108(21):6571–6581, 2004.

[39] William C Swope, Jed W Pitera, Frank Suits, Mike Pitman, Maria Eleftheriou, Blake G Fitch, Robert S Germain, Aleksandr Rayshubski, TJ Christopher Ward, Yuriy Zhestkov, et al. Describing protein folding kinetics by molecular dynamics simulations. 2. example applications to alanine dipeptide and a $\beta$-hairpin peptide. *The Journal of Physical Chemistry B*, 108(21):6582–6594, 2004.

[40] Guha Jayachandran, V Vishal, and Vijay S Pande. Using massively parallel simulation and markovian models to study protein folding: examining the dynamics of the villin headpiece. *The Journal of chemical physics*, 124(16):164902, 2006.

[41] Daniel L Ensign, Peter M Kasson, and Vijay S Pande. Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece. *Journal of molecular biology*, 374(3):806–816, 2007.

[42] Nicholas W Kelley, V Vishal, Grant A Krafft, and Vijay S Pande. Simulating oligomerization at experimental concentrations and long timescales: A markov state model approach. *The Journal of chemical physics*, 129(21):214707, 2008.

[43] Gregory R Bowman, Vijay S Pande, and Frank Noé. *An introduction to Markov state models and their application to long timescale molecular simulation*, volume 797. Springer Science & Business Media, 2013.

[44] Brooke E Husic and Vijay S Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, 2018.

[45] Frank Noé and Edina Rosta. Markov models of molecular kinetics, 2019.

[46] David Chandler. Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. *The Journal of Chemical Physics*, 68(6):2959–2970, 1978.

[47] Hugo Touchette. Introduction to dynamical large deviations of markov processes. *Physica A: Statistical Mechanics and its Applications*, 504:5–19, 2018.

[48] Thierry Dauxois. Fermi, pasta, ulam and a mysterious lady. *arXiv preprint arXiv:0801.1590*, 2008.

[49] Enrico Fermi, P Pasta, S Ulam, and M Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos Scientific Lab., N. Mex., 1955.

[50] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965.

[51] Michael Levitt and Arieh Warshel. Computer simulation of protein folding. *Nature*, 253(5494):694, 1975.

[52] Michael Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of molecular biology*, 104(1):59–107, 1976.

[53] Cyrus Chothia, Arthur M Lesk, Anna Tramontano, Michael Levitt, Sandra J Smith-Gill, Gillian Air, Steven Sheriff, Eduardo A Padlan, David Davies, William R Tulip, et al. Conformations of immunoglobulin hypervariable regions. *Nature*, 342(6252):877, 1989.

[54] Michael Levitt, Miriam Hirshberg, Ruth Sharon, and Valerie Daggett. Potential energy function and parameters for simulations of the molecular dynamics of proteins and nucleic acids in solution. *Computer physics communications*, 91(1-3):215–231, 1995.

[55] Jay W Ponder and David A Case. Force fields for protein simulations. In *Advances in protein chemistry*, volume 66, pages 27–85. Elsevier, 2003.

[56] Chris Oostenbrink, Alessandra Villa, Alan E Mark, and Wilfred F Van Gunsteren. A biomolecular force field based on the free enthalpy of hydration and solvation: the gromos force-field parameter sets 53a5 and 53a6. *Journal of computational chemistry*, 25(13):1656–1676, 2004.

[57] Alexander D MacKerell Jr. Empirical force fields for biological macromolecules: overview and issues. *Journal of computational chemistry*, 25(13):1584–1604, 2004.

[58] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

[59] Burton Wendroff. Difference methods for initial-value problems (robert d. richtmyer and kw morton). *SIAM Review*, 10(3):381–383, 1968.

[60] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.

[61] RE Skyner, JL McDonagh, CR Groom, T Van Mourik, and JBO Mitchell. A review of methods for the calculation of solution free energies and the modelling of systems in solution. *Physical Chemistry Chemical Physics*, 17(9):6174–6191, 2015.

[62] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[63] Vincent A Voelz, Gregory R Bowman, Kyle Beauchamp, and Vijay S Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9 (1-39). *Journal of the American Chemical Society*, 132(5):1526–1528, 2010.

[64] Yuqing Deng and Benoît Roux. Calculation of standard binding free energies: Aromatic molecules in the t4 lysozyme l99a mutant. *Journal of Chemical Theory and Computation*, 2(5):1255–1273, 2006.

[65] Lingle Wang, Yujie Wu, Yuqing Deng, Byungchan Kim, Levi Pierce, Goran Krilov, Dmitry Lupyan, Shaughnessy Robinson, Markus K Dahlgren, Jeremy Greenwood, et al. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *Journal of the American Chemical Society*, 137(7):2695–2703, 2015.

[66] Albert C Pan, Huafeng Xu, Timothy Palpant, and David E Shaw. Quantitative characterization of the binding and unbinding of millimolar drug fragments with molecular dynamics simulations. *Journal of chemical theory and computation*, 13(7):3372–3377, 2017.

[67] P Van der Ploeg and HJC Berendsen. Molecular dynamics simulation of a bilayer membrane. *The Journal of Chemical Physics*, 76(6):3271–3276, 1982.

[68] D Peter Tieleman, Siewert-Jan Marrink, and Herman JC Berendsen. A computer perspective of membranes: molecular dynamics studies of lipid bilayer systems. *Biochimica et Biophysica Acta (BBA)-Reviews on Biomembranes*, 1331(3):235–270, 1997.

[69] Fatemeh Khalili-Araghi, James Gumbart, Po-Chao Wen, Marcos Sotomayor, Emad Tajkhorshid, and Klaus Schulten. Molecular dynamics simulations of membrane channels and transporters. *Current opinion in structural biology*, 19(2):128–137, 2009.

[70] Y. Shen, N. Quirke, and D. Zerulla. Polarisation dependence of the squash mode in the extreme low frequency vibrational region of single walled carbon nanotubes. *Applied Physics Letters*, 106(20):201902, 2015.

[71] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[72] Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for markov model construction. *The Journal of chemical physics*, 139(1):07B604_1, 2013.

[73] Brooke E Husic and Frank Noé. Deflation reveals dynamical structure in non-dominant reaction coordinates. *arXiv preprint arXiv:1907.04101*, 2019.

[74] David De Sancho, Adam Kubas, Po-Hung Wang, Jochen Blumberger, and Robert B. Best. Identification of mutational hot spots for substrate diffusion: Application to myoglobin. *Journal of Chemical Theory and Computation*, 11(4):1919–1927, 2015. PMID: 26574395.

[75] Kelly M. Thayer, Bharat Lakhani, and David L. Beveridge. Molecular dynamics–markov state model of protein ligand binding and allostery in crib-pdz: Conformational selection and induced fit. *The Journal of Physical Chemistry B*, 121(22):5509–5514, 2017. PMID: 28489401.

[76] James MacQueen et al. Some methods for classification and analysis of multivariate observations. 1967.

[77] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[78] Stefan Doerr, Igor Ariz-Extreme, Matthew J Harvey, and Gianni De Fabritiis. Dimensionality reduction methods for molecular simulations. *arXiv preprint arXiv:1710.10629*, 2017.

[79] Robert B Best and Gerhard Hummer. Reaction coordinates and rates from transition paths. *Proceedings of the National Academy of Sciences*, 102(19):6732–6737, 2005.

[80] Hendrik Jung, Kei-ichi Okazaki, and Gerhard Hummer. Transition path sampling of rare events by shooting from the top. *The Journal of Chemical Physics*, 147(15):152716, 2017.

[81] Ch Schütte, Alexander Fischer, Wilhelm Huisinga, and Peter Deuflhard. A direct approach to conformational dynamics based on hybrid monte carlo. *Journal of Computational Physics*, 151(1):146–168, 1999.

[82] Peter Deuflhard, Wilhelm Huisinga, Alexander Fischer, and Ch Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled markov chains. *Linear Algebra and its Applications*, 315(1-3):39–59, 2000.

[83] Gregory R Bowman, Vijay S Pande, and Frank Noé. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Springer Netherlands, 2014.

[84] Marcus Weber. Clustering by using a simplex structure. Technical Report 04-03, ZIB, Takustr. 7, 14195 Berlin, 2003.

[85] Marcus Weber. Improved perron cluster analysis. Technical Report 03-04, ZIB, Takustr. 7, 14195 Berlin, 2003.

[86] Marcus Weber, Wasinee Rungsarityotin, and Alexander Schliep. Perron cluster analysis and its connection to graph partitioning for noisy data. Technical Report 04-39, ZIB, Takustr. 7, 14195 Berlin, 2004.

[87] Susanna Kube and Marcus Weber. Identification of metastabilities in monomolecular conformation kinetics. Technical Report 06-01, ZIB, Takustr. 7, 14195 Berlin, 2005.

[88] Susanna Kube and Marcus Weber. Conformation kinetics as a reduced model for transition pathways. Technical Report 05-43, ZIB, Takustr. 7, 14195 Berlin, 2005.

[89] Peter Deuflhard and Marcus Weber. Robust perron cluster analysis in conformation dynamics. *Linear algebra and its applications*, 398:161–184, 2005.

[90] Susanna Röblitz and Marcus Weber. Fuzzy spectral clustering by pcca+: application to markov state models and data classification. *Advances in Data Analysis and Classification*, 7(2):147–179, 2013.

[91] Marcus Weber and Konstantin Fackeldey. G-pcca: Spectral clustering for non-reversible markov chains. Technical Report 15-35, ZIB, Takustr. 7, 14195 Berlin, 2015.

[92] Yuan Yao, Raymond Z Cui, Gregory R Bowman, Daniel-Adriano Silva, Jian Sun, and Xuhui Huang. Hierarchical nyström methods for constructing markov state models for conformational dynamics. *The Journal of Chemical Physics*, 138(17):05B602_1, 2013.

[93] I. Horenko, E. Dittmer, A. Fischer, and C. Schütte. Automated model reduction for complex systems exhibiting metastability. *Multiscale Modeling & Simulation*, 5(3):802–827, 2006.

[94] Verena Schultheis, Thomas Hirschberger, Heiko Carstens, and Paul Tavan. Extracting markov models of peptide conformational dynamics from simulation data. *Journal of Chemical Theory and Computation*, 1(4):515–526, 2005. PMID: 26641671.

[95] Gregory R Bowman. Improved coarse-graining of markov state models via explicit consideration of statistical uncertainty. *The Journal of Chemical Physics*, 137(13):134111, 2012.

[96] Brooke E Husic and Vijay S Pande. Ward clustering improves cross-validated markov state models of protein folding. *Journal of chemical theory and computation*, 13(3):963–967, 2017.

[97] L. Boltzmann. *Vorlesungen über Gastheorie: Th. Theorie van der Waals'; Gase mit zusammengesetzten Molekülen; Gasdissociation; Schlussbemerkungen.* Vorlesungen über Gastheorie. J. A. Barth, 1898.

[98] Robert Zwanzig. Time-correlation functions and transport coefficients in statistical mechanics. *Annual Review of Physical Chemistry*, 16(1):67–102, 1965.

[99] David Chandler. Statistical mechanics of isomerization dynamics in liquids and the transition state approximation. *The Journal of Chemical Physics*, 68(6):2959–2970, 1978.

[100] John A Montgomery Jr, David Chandler, and Bruce J Berne. Trajectory analysis of a kinetic theory for isomerization dynamics in condensed phases. *The Journal of Chemical Physics*, 70(9):4056–4066, 1979.

[101] DJ Bicout and Attila Szabo. Electron transfer reaction dynamics in non-debye solvents. *The Journal of chemical physics*, 109(6):2325–2338, 1998.

[102] Christoph Dellago, Peter G. Bolhuis, Félix S. Csajka, and David Chandler. Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics*, 108(5):1964–1977, 1998.

[103] Gerhard Hummer. From transition paths to transition states and rate coefficients. *The Journal of chemical physics*, 120(2):516–523, 2004.

[104] Richard C. Tolman. *The principles of statistical mechanics, Chapter XII.* Dover Publications, 1938.

[105] Gerhard Hummer and Attila Szabo. Optimal dimensionality reduction of multistate kinetic and markov-state models. *The Journal of Physical Chemistry B*, 119(29):9029–9037, 2014.

[106] J.G. Kemény and J.L. Snell. *Finite markov chains*. University series in undergraduate mathematics. Van Nostrand, 1960.

[107] Adam Kells, Edina Rosta, and Alessia Annibale. Correlation functions, mean first passage times and the kemeny constant, 2019.

[108] Peter G Doyle. The kemeny constant of a markov chain. *arXiv preprint arXiv:0909.2636*, 2009.

[109] Jeffrey J Hunter. The role of kemeny's constant in properties of markov chains. *Communications in Statistics-Theory and Methods*, 43(7):1309–1321, 2014.

[110] Dario Bini, Jeffrey J Hunter, Guy Latouche, Beatrice Meini, and Peter Taylor. Why is kemeny's constant a constant? *Journal of Applied Probability*, 55(4):1025–1036, 2018.

[111] Karl Gustafson and Jeffrey J Hunter. Why the kemeny time is a constant. *Special Matrices*, 4(1).

[112] Jiri Brummer, Elenna Dugundji, and Daphne van Leeuwen. Optimizing community detection using the kemeny constant. 2018.

[113] Angelo Perico, Roberto Pratolongo, Karl F Freed, Richard W Pastor, and Attila Szabo. Positional time correlation function for one-dimensional systems with barrier crossing: Memory function corrections to the optimized rouse–zimm approximation. *The Journal of Chemical Physics*, 98(1):564–573, 1993.

[114] Gerhard Hummer. Position-dependent diffusion coefficients and free energies from bayesian analysis of equilibrium and replica molecular dynamics simulations. *New Journal of Physics*, 7(1):34, 2005.

[115] Robert B Best and Gerhard Hummer. Coordinate-dependent diffusion in protein folding. *Proceedings of the National Academy of Sciences*, 107(3):1088–1093, 2010.

[116] D Holcman and Z Schuss. 100 years after smoluchowski: stochastic processes in cell biology. *Journal of Physics A: Mathematical and Theoretical*, 50(9):093002, jan 2017.

[117] Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: fifty years after kramers. *Reviews of modern physics*, 62(2):251, 1990.

[118] Nicolae-Viorel Buchete and Gerhard Hummer. Coarse master equations for peptide folding dynamics. *The Journal of Physical Chemistry B*, 112(19):6057–6069, 2008.

[119] Attila Szabo, Klaus Schulten, and Zan Schulten. First passage time approach to diffusion controlled reactions. *The Journal of Chemical Physics*, 72(8):4350–4357, 1980.

[120] M.H. Protter and C.B.J. Morrey. *Intermediate Calculus*. Undergraduate Texts in Mathematics. Springer New York, 2012.

[121] Christian H Jensen, Dmitry Nerukh, and Robert C Glen. Calculating mean first passage times from markov models of proteins. In *AIP Conference Proceedings*, volume 940, pages 150–157. AIP, 2007.

[122] Carl D Meyer Jr. An alternative expression for the mean first passage matrix. *Linear algebra and its applications*, 22:41–47, 1978.

[123] Sunhwan Jo, Taehoon Kim, Vidyashankara G Iyer, and Wonpil Im. Charmm-gui: a web-based graphical user interface for charmm. *Journal of computational chemistry*, 29(11):1859–1865, 2008.

[124] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of computational chemistry*, 26(16):1781–1802, 2005.

[125] David J Earl and Michael W Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

[126] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[127] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[128] David Gfeller and Paolo De Los Rios. Spectral coarse graining of complex networks. *Physical review letters*, 99(3):038701, 2007.

[129] Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40(4):2266–2292, 2012.

[130] Jing Lei, Alessandro Rinaldo, et al. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

[131] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.

[132] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.

[133] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[134] Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, 1994.

[135] Anders Krogh, BjoÈrn Larsson, Gunnar Von Heijne, and Erik LL Sonnhammer. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. *Journal of molecular biology*, 305(3):567–580, 2001.

[136] Mark Gales, Steve Young, et al. The application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304, 2008.

[137] Frank Noé, Hao Wu, Jan-Hendrik Prinz, and Nuria Plattner. Projected and hidden markov models for calculating kinetics and metastable states of complex molecules. *The Journal of chemical physics*, 139(18):11B609_1, 2013.

[138] Fangqiang Zhu and Gerhard Hummer. Pore opening and closing of a pentameric ligand-gated ion channel. *Proceedings of the National Academy of Sciences*, 107(46):19814–19819, 2010.

[139] Fangqiang Zhu and Gerhard Hummer. Theory and simulation of ion conduction in the pentameric glic channel. *Journal of chemical theory and computation*, 8(10):3759–3768, 2012.

[140] Martin K. Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Pérez-Hernández, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank Noé. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *Journal of Chemical Theory and Computation*, 11:5525–5542, October 2015.

[141] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[142] Gopalasamudram Narayana Ramachandran. Stereochemistry of polypeptide chain configurations. *J. Mol. Biol.*, 7:95–99, 1963.

[143] Hiroaki Fukunishi, Osamu Watanabe, and Shoji Takada. On the hamiltonian replica exchange method for efficient sampling of biomolecular systems: Application to protein structure prediction. *The Journal of chemical physics*, 116(20):9058–9067, 2002.

[144] Nicolae-Viorel Buchete and Gerhard Hummer. Peptide folding kinetics from replica exchange molecular dynamics. *Physical Review E*, 77(3):030902, 2008.

[145] Cathal T Leahy, Adam Kells, Gerhard Hummer, Nicolae-Viorel Buchete, and Edina Rosta. Peptide dimerization-dissociation rates from replica exchange molecular dynamics. *The Journal of chemical physics*, 147(15):152725, 2017.

[146] Klaus Eyer, Franziska Paech, Friedrich Schuler, Phillip Kuhn, Reinhard Kissner, Sara Belli, Petra S Dittrich, and Stefanie D Krämer. A liposomal fluorescence assay to study permeation kinetics of drug-like weak bases across the lipid bilayer. *Journal of controlled release*, 173:102–109, 2014.

[147] Magd Badaoui, Adam Kells, Carla Molteni, Callum J Dickson, Viktor Hornak, and Edina Rosta. Calculating kinetic rates and membrane permeability from biased simulation. *The Journal of Physical Chemistry B*, 2018.

[148] Edina Rosta and Gerhard Hummer. Free energies from dynamic weighted histogram analysis using unbiased markov state model. *Journal of Chemical Theory and Computation*, 11(1):276–285, 2014.

[149] Rafael C Bernardi, Marcelo CR Melo, and Klaus Schulten. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1850(5):872–877, 2015.

[150] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, 314(1-2):141–151, 1999.

[151] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical review letters*, 100(2):020603, 2008.

[152] Giovanni Bussi, Alessandro Laio, and Michele Parrinello. Equilibrium free energies from nonequilibrium metadynamics. *Physical review letters*, 96(9):090601, 2006.

[153] Massimiliano Bonomi, Davide Branduardi, Giovanni Bussi, Carlo Camilloni, Davide Provasi, Paolo Raiteri, Davide Donadio, Fabrizio Marinelli, Fabio Pietrucci, Ricardo A Broglia, et al. Plumed: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications*, 180(10):1961–1972, 2009.

[154] SS Antman JE Marsden, L Sirovich S Wiggins, L Glass, RV Kohn, and SS Sastry. *Interdisciplinary Applied Mathematics*, volume 3. Springer, 1993.

[155] Shankar Kumar, John M. Rosenberg, Djamal Bouzida, Robert H. Swendsen, and Peter A. Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[156] Alan M Ferrenberg and Robert H Swendsen. Optimized monte carlo data analysis. *Computers in Physics*, 3(5):101–104, 1989.

[157] Shankar Kumar, Djamal Bouzida, Robert H. Swendsen, Peter A. Kollman, and John M. Rosenberg. The weighted histogram analysis method for free-energy calculations on biomolecules. i: The method. 1992.

[158] Johannes Kästner. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(6):932–942, 2011.

[159] Zoran Kurtović, Massimo Marchi, and David Chandler. Umbrella sampling molecular dynamics study of the dielectric constant of water. *Molecular Physics*, 78(5):1155–1165, 1993.

[160] Benoît Roux. The calculation of the potential of mean force using computer simulations. *Computer physics communications*, 91(1-3):275–282, 1995.

[161] Shankar Kumar, John M Rosenberg, Djamal Bouzida, Robert H Swendsen, and Peter A Kollman. Multidimensional free-energy calculations using the weighted histogram analysis method. *Journal of Computational Chemistry*, 16(11):1339–1350, 1995.

[162] Antonia S. J. S. Mey, Hao Wu, and Frank Noé. xtram: Estimating equilibrium expectations from time-correlated simulation data at multiple thermodynamic states. *Phys. Rev. X*, 4:041018, Oct 2014.

[163] Hao Wu, Antonia SJS Mey, Edina Rosta, and Frank Noé. Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states. *The Journal of chemical physics*, 141(21):12B629_1, 2014.

[164] Lukas S. Stelzl, Adam Kells, Edina Rosta, and Gerhard Hummer. Dynamic histogram analysis to determine free energies and rates from biased simulations. *Journal of Chemical Theory and Computation*, 13(12):6328–6342, 2017. PMID: 29059525.

[165] Mohammad M Sultan, Gert Kiss, Diwakar Shukla, and Vijay S Pande. Automatic selection of order parameters in the analysis of large scale molecular dynam-

ics simulations. *Journal of chemical theory and computation*, 10(12):5217–5223, 2014.

[166] Christian R Schwantes and Vijay S Pande. Modeling molecular kinetics with tica and the kernel trick. *Journal of chemical theory and computation*, 11(2):600–608, 2015.

[167] Pratyush Tiwary and BJ Berne. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proceedings of the National Academy of Sciences*, 113(11):2839–2844, 2016.

[168] Mohammad M Sultan and Vijay S Pande. Automated design of collective variables using supervised machine learning. *The Journal of chemical physics*, 149(9):094106, 2018.

[169] Hendrik Jung, Roberto Covino, and Gerhard Hummer. Artificial intelligence assists discovery of reaction coordinates and mechanisms from molecular dynamics simulations. *arXiv preprint arXiv:1901.04595*, 2019.

[170] Brooke E. Husic and Vijay S. Pande. Unsupervised learning of dynamical and molecular similarity using variance minimization, 2017.

[171] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. Vampnets for deep learning of molecular kinetics. *Nature communications*, 9(1):5, 2018.

[172] Andreas Mardt, Luca Pasquali, Frank Noé, and Hao Wu. Deep learning markov and koopman models with physical constraints, 2019.