

# Upgrade Report

Vladimir Koskin

November 2022

## Contents

<b>1</b>	<b>Introduction giving the context of the work</b>	<b>3</b>
<b>2</b>	<b>Dynamical Systems - Kinetic networks framework</b>	<b>3</b>
2.1	Introduction (Abstract) . . . . .	3
2.2	Introduction (Technical) . . . . .	3
2.2.1	Markov chains . . . . .	3
2.2.2	Eigenvalues and Eigenvectors . . . . .	4
2.2.3	Correlation Functions . . . . .	5
2.2.4	Mean First Passage Times (MFPT) . . . . .	5
2.3	RESULT 1 (Published): Paper 1 - Correlation functions, MFPT, and the Kemeny constant . . . . .	8
2.3.1	Introduction . . . . .	8
2.3.2	Linking MFPTs and Kemeny constants to correlation functions . . . . .	9
2.3.3	Constructing rate matrices from MFPTs . . . . .	10
2.3.4	Construction transition probability matrices from MFPTs . . . . .	11
2.3.5	Conclusion . . . . .	12
2.4	RESULT 2 (Unpublished): MFPT definition for continuous times . . . . .	13
2.4.1	Introduction . . . . .	13
2.4.2	Continuous time MFPT with "Entry" convention . . . . .	13
2.4.3	Continuous time MFPT with "Reach" convention . . . . .	14
2.4.4	Detailed Balance in terms of Mean First Passage Times . . . . .	15
2.4.5	Mean First Observation Time: an intuitive way to describe Detailed Balance in terms of MFPT . . . . .	15
2.4.6	Side note / Future - W matrix and similarity to M matrix . . . . .	18
2.4.7	Conclusion . . . . .	18
<b>3</b>	<b>Enhanced Sampling Problem</b>	<b>19</b>
3.1	Introduction (Abstract) . . . . .	19
3.2	Introduction (Technical) . . . . .	19
3.2.1	Umbrella Sampling . . . . .	19
3.2.2	WHAM . . . . .	19
3.2.3	DHAM . . . . .	20
3.2.4	Estimation error of the PMF: minRMSD . . . . .	20
3.3	RESULT 1 (Unpublished) ESTIMATOR - DHAR, Statistical uncertainty, and Markov Neural Networks . . . . .	20
3.3.1	Introduction . . . . .	20
3.3.2	Single Non-Biased Trajectory . . . . .	21
3.3.3	Multiple Biased Trajectories . . . . .	23
3.3.4	Proof that enforcing DB in a 2nd step keeps $\pi$ unchanged . . . . .	25
3.3.5	Applications . . . . .	27
3.3.6	Conclusion . . . . .	29
3.3.7	Side note: Markov Neural Network . . . . .	30
3.4	RESULT 3 (Unpublished) BIASING - MDP-like Biasing . . . . .	35
3.4.1	Introduction . . . . .	35
3.4.2	Heuristics . . . . .	36
3.4.3	Independence of estimated lines in our MSM with DHAM . . . . .	37
3.4.4	First approach . . . . .	38
3.4.5	Application . . . . .	39

3.4.6	Conclusion . . . . .	42
3.4.7	Side Notes from Journal . . . . .	43
3.5	RESULT 4 (Unpublished) BIASING - Dancing Umbrellas . . . . .	43
3.5.1	The method . . . . .	43
3.5.2	Application and comparison to standard umbrella sampling . . . . .	46
3.5.3	Analytic result for position correction / PMF gradient $\leftrightarrow$ histogram deviation . . . . .	50
3.5.4	Adapting the Umbrella forces . . . . .	51
3.5.5	Application to 2-dimensional potential . . . . .	53
3.5.6	Conclusion . . . . .	55
3.6	RESULT 5 (Unpublished) BIASING - Cyclic Lifting . . . . .	55
3.6.1	Introduction . . . . .	55
3.6.2	Turning a 1-D space into a circular network . . . . .	56
3.6.3	Create a net flux . . . . .	57
3.6.4	First implementation . . . . .	60
3.6.5	First implementation, results . . . . .	60
3.6.6	Application to assymmetric 2-well potential . . . . .	63
3.6.7	Parallelize with another simulation lifted to opposite direction . . . . .	65
3.6.8	Comparing to Umbrella Sampling and Unbiased Simulations . . . . .	67
3.6.9	Conclusion . . . . .	69
3.6.10	Additional Ideas . . . . .	69
<b>4</b>	<b>Clustering Problem</b> . . . . .	<b>69</b>
4.1	Introduction (Abstract) . . . . .	69
4.2	Introduction (Technical) . . . . .	69
4.2.1	Clustering in complex networks . . . . .	69
4.2.2	Modularity . . . . .	70
4.2.3	Kinetic clustering . . . . .	71
4.3	RESULT 1 (Published) THEORY - Timescales as variational parameter for kinetic network clustering . . . . .	74
4.3.1	Introduction . . . . .	75
4.3.2	Coarse Graining of Rate Matrices based on MFPTs . . . . .	75
4.3.3	Variational principle for Kemeny Constant in Hummer-Szabo Coarse Graining . . . . .	78
4.3.4	Coarse Graining of Transition Matrices based on MFPTs . . . . .	78
4.3.5	Analytic maximization of the Kemeny constant . . . . .	79
4.3.6	Conclusion . . . . .	81
4.4	RESULT 3 (Published) ALGO - Parrallel Tempering Variational Clustering . . . . .	82
4.4.1	Introduction . . . . .	82
4.4.2	Original Parallel Tempering . . . . .	83
4.4.3	Parallel Tempering Variational Clustering . . . . .	84
4.4.4	Clustering results . . . . .	86
4.4.5	Conclusion . . . . .	91
4.5	RESULT 4 (Unpublished) ALGO - (Faster algorithm inspired from Louvain Modularity maximization ) . . . . .	92
4.5.1	Introduction . . . . .	92
4.5.2	Original Louvain Method . . . . .	92
4.5.3	Louvain for Kemeny . . . . .	92
4.5.4	Conclusion . . . . .	96
<b>5</b>	<b>X - Plan and timetable for remainder work</b> . . . . .	<b>97</b>
<b>A</b>	<b>APPENDIX - Dynamical Systems / Kinetic Networks</b> . . . . .	<b>98</b>
<b>B</b>	<b>APPENDIX 2 - Enhanced Sampling</b> . . . . .	<b>98</b>
<b>C</b>	<b>APPENDIX 3 - Clustering</b> . . . . .	<b>99</b>
C.1	Condition on the Coarse Grained MFPTs. . . . .	99
C.2	Derivation of Eq. (296) . . . . .	99
C.3	Retrieval of Hummer-Szabo Coarse Graining . . . . .	100
C.4	Diffusion on 1D Symmetric potential: 3-state clustering . . . . .	101
C.5	Diffusion on 1D potential: $m$ -state clustering . . . . .	106
C.6	Discrete random walk on 1D potential: $m$ -state clustering . . . . .	107

C.7 Random walks on complex networks: $m$ -state clustering . . . . .	108
<b>D SI - Clustering</b>	<b>110</b>
<b>E Acknowledgements</b>	<b>110</b>

## 1 Introduction giving the context of the work

We study Dynamic Systems in general, with a focus on Molecular Dynamics.

In order to understand real dynamic systems, we need to sample the dynamics, either via observation or via numerical simulation. In the case of Molecular Dynamics we have to perform simulations by relying the laws of physics to compute the forces and generate trajectories.

In general we do not always know the exact nature of the dynamics, and therefore need to use a model, with the associated statistical estimators. Alternatively, as in Molecular Dynamics, the true dynamics can be known but computationally untractable; therefore, a simpler and relevant model is chosen to approximate the dynamics. In particular, I considered the Markov State Model, which has been used to model Molecular Dynamics extensively.

Sometimes the dynamics contain important events that are rare, and therefore require a large amount of time to be sampled and estimated properly.

It is possible to introduce a bias in simulations to make rare events of interest more frequent and sample them quicker, and use an unbiased estimator to obtain an estimate of the quantities of interest without the bias. This approach is called Enhanced Sampling, and is used to increase the exploration of configuration space, increase computational efficiency and reduce the statistical uncertainty of the obtained estimates.

Once a Markov State Model has been estimated, it is represented in the form of a Network with edges defining kinetic properties, which has been referred to as "Kinetic Network" in other works.

When Markov State Models (or Kinetic Networks) contain a large number of nodes, to visualize and interpret them we use Coarse-Graining (Clustering, which is a form of dimensionality reduction, classification problem). The optimal approach to coarse-grain a Markov State Model based on its kinetic properties is not yet known. In this context, my work is mainly focused on enhanced sampling and coarse-graining methods, that approximate the dynamics as Markov State Models.

In the present report I will first introduce and discuss the concepts, theories, and mathematical objects that are used extensively in the presented context. Then, I will present my work related to the Enhanced Sampling problem, followed by the Clustering problem. Finally, I will discuss potential future research topics and present my plan and timetable for the remainder of the work upon completion of the PhD.

## 2 Dynamical Systems - Kinetic networks framework

### 2.1 Introduction (Abstract)

Recently, kinetic networks have been important driving forces in molecular simulations of diverse applications [1–12]. One of the simplest mathematical and physical descriptions of kinetic networks is provided by Markov State Models (MSM) [13–20]. The broad applicability of MSM is noticeable in a wide variety of fields and has resulted in many aspects of the theory of Markov processes being derived on multiple occasions in diverse ways [21–23]. In this study we aim to present a unified framework that links several results in the literature and provide some novel insights.

### 2.2 Introduction (Technical)

Here we collate and present a variety of results pertaining to dynamical systems, and kinetic network models in particular, in a unified framework. The aim is to lay out explicit links between several important quantities commonly studied in the field, including mean first passage times (MFPTs), correlation functions and the Kemeny constant. We provide new insights on (i) a simple physical interpretation of the Kemeny constant, (ii) a relationship to infer equilibrium distributions and rate matrices from measurements of MFPTs

#### 2.2.1 Markov chains

A kinetic network consists of  $n$  discrete states labelled  $i = \{1, \dots, n\}$ . Each discrete state has a time dependent probability to be occupied  $p_i(t)$ . The evolution of the probabilities  $P(x, t)$ , in continuous time, is governed by

the rate at which the system moves between different states. The rate  $K_{ji}$  of transition from state  $i$  to state  $j$  is given by

$$K_{ji} = \lim_{\tau \rightarrow 0} \frac{p(j, t + \tau | i, t)}{\tau}, \quad (1)$$

where  $P(j, t + \tau | i, t)$  is the probability to transition to state  $j$  from state  $i$  in a small interval of time  $\tau$ . The time-evolution of the probability of state occupation is given by the master equation

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} [k_{ij} p_j(t) - k_{ji} p_i(t)], \quad (2)$$

which can be written in matrix notation

$$\frac{d\mathbf{p}}{dt} = \mathbf{K}\mathbf{p} \quad (3)$$

using the fact that the diagonal elements of the rate matrix  $\mathbf{K}$  are necessarily given by  $k_{ii} = -\sum_j k_{ji}$  for conservation of probability. If  $\mathbf{K}$  has a complete set of eigenvectors, Eq. (260) is solved by

$$\mathbf{p}(t) = e^{\mathbf{K}t} \mathbf{p}(0), \quad (4)$$

where the so-called propagator  $e^{\mathbf{K}t}$  is a matrix which evolves the probability distribution at one time to a new distribution at a time  $t$  later.

In discrete time  $t = \ell\tau$ , where moves between states happen at multiples  $\ell = 1, 2, \dots$  of the lag-time  $\tau$ , one defines the transition probability matrix  $\mathbf{M}(\tau) = e^{\mathbf{K}\tau}$ , whose elements give the transition probability over a single time step, for any pair of states. The probability vector at the  $\ell$ -th time step can then be found as

$$\mathbf{p}(\ell) = [\mathbf{M}(\tau)]^\ell \mathbf{p}(0). \quad (5)$$

### 2.2.2 Eigenvalues and Eigenvectors

The rate matrix  $\mathbf{K}$  can be spectrally decomposed and represented in terms of its eigenvalues  $\{\lambda_\ell\}_{\ell=1}^n$  and left and right eigenvectors,  $\{\phi^{(\ell)}\}_{\ell=1}^n$  and  $\{\psi^{(\ell)}\}_{\ell=1}^n$ , respectively

$$\mathbf{K} = \sum_{\ell=1}^n \lambda_\ell \psi^{(\ell)} \phi^{(\ell)}. \quad (6)$$

We will focus on systems satisfying detailed balance, where eigenvalues are real. The largest eigenvalue of  $\mathbf{K}$  is 0 and so all other eigenvalues are negative. They are usually indexed in descending order

$$0 = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N. \quad (7)$$

The corresponding eigenvectors are indexed in the same manner. The right eigenvector corresponding to the zero eigenvalue  $\psi^{(1)}$  is known as the stationary probability (or, for reversible dynamics, equilibrium probability)  $\pi$  with elements  $\pi_i$ . The corresponding left eigenvector  $\phi^{(1)}$  is the  $n$ -dimensional row vector with all the components equal to 1,  $\mathbf{1}_n^T$ .

It can be shown that the elements of the left and right eigenvectors are related by the equilibrium probability

$$\psi_i^{(\ell)} = \phi_i^{(\ell)} \pi_i \quad (8)$$

and  $\sum_i \psi_i^{(\ell)} = 0$  for  $\ell > 1$ . Hence, left and right eigenvectors associated to non-zero eigenvalues will have positive and negative entries. These contain useful kinetic information, as they are related to relaxation processes.

This link can be seen by using the spectral decomposition from Eq. (6) in Eq. (4) and singling out the contribution from  $\ell = 1$

$$p_i(t) - \pi_i = \sum_{\ell \geq 2}^n e^{-|\lambda_\ell|t} \psi_i^{(\ell)} \phi^{(\ell)} \cdot \mathbf{p}(0), \quad (9)$$

where we have used  $\psi_i^{(1)} = \pi_i$ ,  $\phi_j^{(1)} = 1 \forall j$ ,  $\sum_j p_j(0) = 1$  and  $\lambda_\ell < 0 \forall \ell \geq 2$ . For large time, the right hand side (RHS) of Eq. (9) is dominated by the first term in the sum, so the probability distribution will tend towards the equilibrium distribution with a timescale given by  $\tau_2 = 1/|\lambda_2|$  (often called the relaxation time). The other timescales, are each given by the inverse of the magnitude of the corresponding eigenvalue

$$\tau_\ell = 1/|\lambda_\ell| \quad (10)$$

and can be interpreted as the time with which the rate matrix moves probability density between the oppositely signed regions of the corresponding eigenvector. This can be seen by considering the evolution of the scalar product between the time-dependent probability and the different eigenvectors

$$\phi^{(s)} \cdot \mathbf{p}(t) = e^{-|\lambda_s|t} \phi^{(s)} \cdot \mathbf{p}(0). \quad (11)$$

Each scalar product vanishes on a timescale set by the inverse eigenvalue, indicating that the probability mass becomes distributed evenly across positive and negative entries of the eigenvector  $\phi^{(s)}$ , on the timescale  $1/|\lambda_s|$ .

### 2.2.3 Correlation Functions

The correlation function between two observables  $\theta_i$  and  $\theta_j$  at a lagtime  $\tau$  is given by

$$C_{ji}(\tau, t) = \langle \theta_j(t + \tau)\theta_i(t) \rangle - \langle \theta_j(t + \tau) \rangle \langle \theta_i(t) \rangle \quad (12)$$

Defining  $\theta_i(t)$  as the indicator function which takes value 1 when the system is in state  $i$  at time  $t$  and 0 otherwise, the first term of Eq. (12) gives the joint probability that the system is in state  $i$  at time  $t$  and in state  $j$  at a time  $\tau$  later

$$\begin{aligned} C_{ji}(\tau, t) &= P(j, t + \tau; i, t) - p_j(t + \tau)p_i(t) \\ &= [P(j, t + \tau|i, t) - p_j(t + \tau)]p_i(t) \end{aligned} \quad (13)$$

where the conditional probability  $P(j, t + \tau|i, t)$  is given by the  $ji$ 'th entry of the propagator matrix, and depends only on the lagtime  $\tau$ , i.e.  $P(j, t + \tau|i, t) = [e^{\mathbf{K}\tau}]_{ji} = P(j, \tau|i, 0)$ . If the system is in equilibrium, where one-time quantities are time-independent, the correlation function becomes a function of only the lagtime

$$C_{ji}^{\text{eq}}(\tau) = [e^{\mathbf{K}\tau}]_{ji}\pi_i - \pi_j\pi_i. \quad (14)$$

In many practical situations, one averages Eq. (13) over the earlier time  $t$ , with the expectation that if the system is ergodic (i.e. a sufficiently long trajectory will sample all states with equilibrium probability) the resulting time average equates the equilibrium correlator

$$\overline{C_{ji}(\tau, t)} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T dt C_{ji}(\tau, t) \equiv C_{ji}^{\text{eq}}(\tau). \quad (15)$$

Repeating the same steps that led to Eq. (9), the equilibrium correlator (269) can be written as a superposition of exponential functions

$$C_{ji}^{\text{eq}}(\tau) = \sum_{\ell \geq 2} e^{-|\lambda_\ell|\tau} \psi_j^{(\ell)} \phi_i^{(\ell)} \pi_i \quad (16)$$

decaying to zero at large lagtime. The area underneath the correlator, then serves as a measure of how quickly an initial probability distribution will tend to the equilibrium probability, and it can be expressed as a weighted sum of the timescales in the system

$$\int_0^\infty C_{ji}^{\text{eq}}(\tau) d\tau = \sum_{\ell \geq 2} \frac{1}{|\lambda_\ell|} \psi_j^{(\ell)} \phi_i^{(\ell)} \pi_i = \sum_{\ell \geq 2} \tau_\ell \psi_j^{(\ell)} \phi_i^{(\ell)} \quad (17)$$

where we have also used Eq. (8).

### 2.2.4 Mean First Passage Times (MFPT)

Next, we derive an expression for MFPTs, i.e. the expected time it takes to the system to first reach a state  $j$  given its current state is  $i$ ,  $t_{ji}$ , within the fundamental theory of Markov processes. We will consider the discrete and continuous time cases separately to highlight the subtle theoretical difference between the two cases.

#### 2.2.4.1 MFPT - discrete time

First we consider the case where the system can make transitions at discrete intervals, without loss of generality we define our units of time such that this time interval is 1. This system is defined by a transition matrix  $\mathbf{M}$ , such that  $\sum_j M_{ji} = 1 \forall i$ , which has eigenvalues  $1 = \lambda'_1 \geq \lambda'_2 \geq \dots \geq \lambda'_N$  and eigenvectors as for the rate matrix  $\mathbf{K}$ .

We will use a prime index to denote quantities in discrete time dynamics that differ from their analogues in continuous time dynamics, for which we will use the same symbols without the prime. Accordingly, we will denote with  $t'_{ji}$  the mean number of *time steps* that it takes to the system to first reach  $j$  from  $i$ , in discrete time dynamics, whereas the corresponding quantity in continuous time dynamics will be denoted with  $t_{ji}$ , and will measure the mean *time* for the first visit to  $j$ , from  $i$ , to occur.

When the system starts in state  $i$ , it can either move to  $j$  directly (i.e. in one time step), with probability  $M_{ji}$ , or transition to some other state  $k$  with probability  $M_{ki}$  (in one time step) and then move to  $j$  in a time of  $t'_{jk}$ , ( $t'_{jk} + 1$  in total), leading to the recursion

$$t'_{ji} = M_{ji} + \sum_{k \neq j} (t'_{jk} + 1) M_{ki} = 1 + \sum_{k \neq j} t'_{jk} M_{ki}. \quad (18)$$

We can rewrite Eq. (18) as

$$\sum_k t'_{jk}(\delta_{ki} - M_{ki}) = 1 - M_{ji}t'_{jj} \quad (19)$$

where  $\delta_{ki}$  is the Kronecker delta, that leads to the more convenient matrix form

$${\mathbf{t}'_j}^T (\mathbf{I} - \mathbf{M}) = (1 - M_{j1}t'_{jj}, \dots, 1 - M_{jN}t'_{jj}) \quad (20)$$

where we have defined  $\mathbf{t}'_j^T = (t'_{j1}, \dots, t'_{jN})$  as the row vector with the MFPTs to  $j$  as components.

If  $\mathbf{M}$  has a complete set of orthonormal eigenvectors (which is guaranteed if detailed balance is satisfied), one can express  $\mathbf{t}'_j^T$  as a linear combination of the (left) eigenvectors of  $\mathbf{M}$ , for certain coefficients  $a_{nm}$  to be determined a posteriori

$${\mathbf{t}'_j}^T = \sum_\ell a_{j\ell} \phi^{(\ell)}. \quad (21)$$

Inserting in Eq. (20) gives the vector equation

$$\sum_\ell a_{j\ell} (1 - \lambda'_\ell) \phi^{(\ell)} = (1 - M_{j1}t'_{jj}, \dots, 1 - M_{jN}t'_{jj}). \quad (22)$$

Next we consider the equation for the component  $r$

$$\sum_\ell a_{j\ell} (1 - \lambda'_\ell) \phi_r^{(\ell)} = 1 - M_{jr}t'_{jj}. \quad (23)$$

Multiplying left and right hand sides times  $\psi_r^{(s)}$  and summing over  $r$  gives

$$\sum_{\ell>1} a_{j\ell} (1 - \lambda'_\ell) \delta_{\ell s} = \delta_{s1} - \lambda_s \psi_j^{(s)} t'_{jj} \quad (24)$$

where we have used that  $\psi^{(s)}$  is the right eigenvector of  $\mathbf{M}$  associated to eigenvalue  $\lambda'_s$ , and the properties of the eigenvectors of the matrix  $\mathbf{M}$ ,  $\sum_r \psi_r^{(s)} = \delta_{s1}$ , and  $\sum_r \phi_r^{(\ell)} \psi_r^{(\ell)} = \delta_{\ell s}$ . Equation (24) yields for  $s = 1$

$$t'_{jj} = \frac{1}{\pi_j} \quad (25)$$

This quantity is greater than or equal to one, with equality holding for  $\pi_j = 1$ , and it can be interpreted as the expected number of time steps it takes to the system to first visit state  $j$ , *after* its release from state  $j$  itself, also known as the "recurrence time" or Kac's lemma [24]. Equation (25) can also be derived from Eq. (18) without assuming the existence of a complete set of eigenvectors of  $\mathbf{M}$ . Multiplying Eq. (31) times  $\pi_i$ , summing over  $i$

$$\sum_i t'_{ji} \pi_i = 1 + \sum_{ki} t'_{jk} M_{ki} \pi_i - \sum_i t'_{jj} M_{ji} \pi_i \quad (26)$$

and using  $\sum_i M_{ji} \pi_i = \pi_j$  one obtains  $1 = t'_{jj} \pi_j$  which gives Eq. (25). At this point it should be noted that some studies in the literature set this quantity to zero as a 'convention'. The analysis above shows that, in the discrete time formulation of MFPTs, convention Eq. (25) should be used. For  $s > 1$ , using Eq. (25) one gets from Eq. (24)

$$a_{js} = -\frac{1}{\pi_j} \frac{\lambda'_s}{1 - \lambda'_s} \psi_j^{(s)}. \quad (27)$$

Singling out the contribution from  $a_{j1}$  in Eq. (21)

$$\mathbf{t}'_j = a_{j1} \phi^{(1)} + \sum_{\ell>1} a_{j\ell} \phi^{(\ell)} \quad (28)$$

using  $\phi_k^{(1)} = 1 \forall k$  and Eq. (27), we get

$$t'_{jk} = a_{j1} - \frac{1}{\pi_j} \sum_{\ell>1} \frac{\lambda'_\ell}{1 - \lambda'_\ell} \psi_j^{(\ell)} \phi_k^{(\ell)} \quad (29)$$

where  $a_{j1}$  can be determined by setting  $j = k$  in the above and using Eq. (25)

$$a_{j1} = \frac{1}{\pi_j} \left( 1 + \sum_{\ell>1} \frac{\lambda'_\ell}{1 - \lambda'_\ell} \phi_j^{(\ell)} \psi_j^{(\ell)} \right). \quad (30)$$

Substituting in Eq. (29), we finally obtain an explicit relation for the MFPTs in terms of the eigenvalues and eigenvectors of the transition matrix

$$t'_{jk} = \frac{1}{\pi_j} \left[ 1 + \sum_{\ell>1} \frac{\lambda'_\ell}{1 - \lambda'_\ell} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_k^{(\ell)}) \right]. \quad (31)$$

This formula, also derived in [25, 26], will serve as a starting point to derive a number of useful relations in the following sections.

#### 2.2.4.2 MFPT - continuous time

Next we consider how these results differ when our system is described by a continuous time rate matrix  $\mathbf{K}$  instead of a discrete time transition probability matrix. Results for continuous time dynamics can be derived by setting the time step to  $\tau$  in the discrete time dynamics, and taking the limit  $\tau \rightarrow 0$  at the end. For small but finite  $\tau$ , the transition matrix  $\mathbf{M}$  can be written as  $e^{\mathbf{K}\tau}$ , and its eigenvalues are given by  $\lambda'_\ell = e^{\lambda_\ell \tau}$ . Defining  $t_{ji} = t'_{ji}\tau$  as the mean first time from  $i$  to  $j$ , and using the same logic as in Eq. (18), we can write a similar recursion

$$t_{ji} = [e^{\mathbf{K}\tau}]_{ji}\tau + \sum_{k \neq j} [e^{\mathbf{K}\tau}]_{ki}(t_{jk} + \tau) = \tau + \sum_{k \neq j} [e^{\mathbf{K}\tau}]_{ki}t_{jk}, \quad (32)$$

that can be rearranged as in Eq. (33),

$$\sum_k (\delta_{ki} - [e^{\mathbf{K}\tau}]_{ki})t'_{jk} = 1 - [e^{\mathbf{K}\tau}]_{ji}t'_{jj}. \quad (33)$$

Following the same steps that led to Eq. (31) we can arrive at

$$t'_{ji} = \frac{1}{p_j^{\text{eq}}} \left[ 1 + \sum_{\ell>1} \frac{e^{\lambda_\ell \tau}}{1 - e^{\lambda_\ell \tau}} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}) \right] \quad (34)$$

Finally, using  $t_{ji} = t'_{ji}\tau$  and taking the limit  $\tau \rightarrow 0$ , gives a formula for the MFPTs in continuous time dynamics, in terms of eigenvalues and eigenvectors of the rate matrix

$$t_{ji} = \frac{1}{p_j^{\text{eq}}} \sum_{\ell>1} \frac{1}{|\lambda_\ell|} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}). \quad (35)$$

Note that in contrast to the discrete time result Eq. (25), in continuous time dynamics, Eq. (35) implies

$$t_{jj} = 0, \quad (36)$$

which is intuitively understood, as here there is no time step to wait to return to the state.

As an aside, we observe that expanding Eq. (33) for small  $\tau$  as in Eq. (37)

$$-\tau \sum_k t'_{jk} K_{ki} = 1 - (\delta_{ji} + \tau K_{ji})t'_{jj} \quad (37)$$

using Eq. (25),  $t_{ji} = \tau t'_{ji}$  and *then* letting  $\tau \rightarrow 0$ , gives

$$\mathbf{tK} = -\mathbf{1}_n \mathbf{1}_n^T + \mathbf{D}_n^{-1}, \quad (38)$$

where  $\mathbf{D}_n$  is an  $n \times n$  diagonal matrix with  $\boldsymbol{\pi}$  on the diagonal. Note that the order in which these operations are executed matters, as  $t'_{jj}$  and  $t_{jk}$  (with  $j \neq k$ ) should remain finite as  $\tau$  is sent to zero. Taking the limit naively, leads to the expression given in Eq. (39), with  $\mathbf{t}_j^T = (t_{j1}, \dots, t_{jN})$

$$\mathbf{t}_j^T \mathbf{K} = -\mathbf{1}_n^T, \quad (39)$$

which is sometimes reported in the literature. This is equivalent to  $\mathbf{tK} = -\mathbf{1}_n \mathbf{1}_n^T$ , thus it differs from Eq. (38) for the diagonal terms. It is easy to show that Eq. (38) is correct, while Eq. (39) is not, e.g. by multiplying both expressions times  $\boldsymbol{\pi}$  from right and using  $\mathbf{K}\boldsymbol{\pi} = 0$ ,  $\mathbf{1}_n^T \boldsymbol{\pi} = 1$  and  $\mathbf{D}_n^{-1} \boldsymbol{\pi} = \mathbf{1}_n$ .

Finally we note that, although Eq. (38) provides a correct expression for the MFPTs,  $\mathbf{K}$  is not directly invertible due the presence of zero eigenvalues, hence MFPTs are more easily computed from relations that we will derive in the next sections, which directly follow from (35).

#### 2.2.4.3 Kemeny Constant

Starting with Eq. (35) we can examine the quantity  $\sum_j \pi_j t_{ji}$  and make use of  $\sum_j \psi_j^{(\ell)} = \delta_{\ell,1}$  and  $\sum_j \phi_j^{(\ell)} \psi_j^{(\ell)} = 1$  for all  $\ell$ , to get

$$\begin{aligned} \sum_j \pi_j t_{ji} &= \sum_j \sum_{\ell>1} \frac{1}{|\lambda_\ell|} \psi_j^{(\ell)} (\phi_j^{(\ell)} - \phi_i^{(\ell)}) = \sum_{\ell>1} \frac{1}{|\lambda_\ell|} \\ &= \sum_{\ell>1} \tau_\ell \equiv \zeta. \end{aligned} \quad (40)$$

This result is known as the Kemeny constant [27–29] and is remarkable as it relates a weighted sum of MFPTs starting from some state  $i$  to a sum over relaxation timescales (which is independent of the particular choice of  $i$ ).

The corresponding quantity in discrete time dynamics is obtained summing Eq. (31) over  $j$

$$\sum_j \pi_j t'_{ji} - N = \sum_{\ell>1} \frac{\lambda'_\ell}{1 - \lambda'_\ell} (1 - \delta_{\ell,1}) = \sum_{\ell>1} \left( \frac{1}{1 - \lambda'_\ell} - 1 \right)$$

which, simplifies to

$$\sum_j \pi_j t'_{ji} = 1 + \sum_{\ell>1} \frac{1}{1 - \lambda'_\ell} \quad (41)$$

or, using Eq. (25), to

$$\sum_{j(\neq i)} \pi_j t'_{ji} = \sum_{\ell>1} \frac{1}{1 - \lambda'_\ell} \equiv \zeta'. \quad (42)$$

#### 2.2.4.4 A simple proof of Kemeny Constant's constancy

A simple proof for the independence of the quantity  $\sum_j \pi_j t'_{ji}$ , on the state  $i$ , which does not require the eigenvectors of  $\mathbf{M}$  to form a complete set, and hence it holds for systems that violate detailed balance, can be derived as follows. Multiplying Eq. (18) times  $\pi_j$  and summing over  $j$  we get

$$\sum_j \pi_j t'_{ji} = 1 + \sum_{jk} \pi_j t'_{jk} Q_{ki} - \sum_j \pi_j t'_{jj} Q_{ji} \quad (43)$$

Using Eq. (25) and  $\sum_j Q_{ji} = 1$  one has

$$\boldsymbol{\pi} \mathbf{t}' = \boldsymbol{\pi} \mathbf{t}' \mathbf{M} \quad (44)$$

showing that  $\boldsymbol{\pi} \mathbf{t}'$  is a left eigenvector of  $\mathbf{M}$  associated to eigenvalue 1, hence it has to be proportional to  $\mathbf{1}_n^T$  and all its entries must be identical.

For the continuous time dynamics, one can similarly prove the constancy of Kemeny constant starting from Eq. (38), that reads, in scalar form

$$\sum_j t_{ij} K_{j\ell} = -1 + \frac{1}{\pi_i} \delta_{i\ell}. \quad (45)$$

Multiplying times  $\pi_i$  and summing over  $i$  we get  $\sum_{ij} \pi_i t_{ij} K_{j\ell} = 0$  hence

$$\boldsymbol{\pi} \mathbf{t} \mathbf{K} = 0. \quad (46)$$

This shows that  $\boldsymbol{\pi} \mathbf{t}$  is a left eigenvector of the rate matrix associated to eigenvalue zero, hence proportional to  $\mathbf{1}_n^T$ , and must have all its components must be equal.

### 2.3 RESULT 1 (Published): Paper 1 - Correlation functions, MFPT, and the Kemeny constant

#### 2.3.1 Introduction

With the theory laid out, we are now equipped to make some observations about how these quantities relate. In particular we will show how MFPTs and Kemeny constant are related to rate matrices and correlation functions. This will lead to a simple interpretation of the Kemeny constant and to a recipe for reconstructing rate matrices from MFPTs measurements, which may be helpful in milestoning[18, 30–33] and transition path sampling [34–38].

For this work, we will focus on continuous time dynamics, as much of the focus on MFPTs in the literature is for discrete time dynamics.

### 2.3.2 Linking MFPTs and Kemeny constants to correlation functions

In this section, we provide expressions for MFPTs in terms of rate matrices and correlation functions and provide a physical interpretation for Kemeny constants. We start by adding and subtracting  $\psi_j^{(1)}$  from equation (35), using  $\phi_i^{(1)} = 1 \forall i$  and  $|\lambda_\ell| = -\lambda_\ell \forall \ell > 1$

$$t_{ji} = \frac{1}{\pi_j} \left[ \psi_j^{(1)} \phi_j^{(1)} - \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} \phi_j^{(\ell)} - \psi_j^{(1)} \phi_i^{(1)} + \sum_{\ell>1} \frac{1}{\lambda_\ell} \psi_j^{(\ell)} \phi_i^{(\ell)} \right] \quad (47)$$

to reformulate the expression for the MFPTs in terms of matrix elements

$$t_{ji} = \frac{1}{\pi_j} \left[ (\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})_{jj}^{-1} - (\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})_{ji}^{-1} \right] \quad (48)$$

where we have used  $\boldsymbol{\pi} = \boldsymbol{\psi}^{(1)}$  and  $\mathbf{1}_n^T = \boldsymbol{\phi}^{(1)}$ . This gives an explicit formula for MFPTs in continuous time dynamics, in terms of rate matrices, which complements similar results available in the literature for discrete time dynamics [39], formulated in terms of the so-called 'fundamental matrix'  $(\mathbf{P} \mathbf{1}_n^T + \mathbf{I} - \mathbf{M})^{-1}$ . Now using Eq. (??), one can provide yet another expression for MFPTs, in terms of time-integrated correlation functions

$$t_{ji} = \frac{1}{\pi_j} \left[ \frac{\int_0^\infty C_{jj}^{\text{eq}}(\tau) d\tau}{\pi_j} - \frac{\int_0^\infty C_{ji}^{\text{eq}}(\tau) d\tau}{\pi_i} \right] \quad (49)$$

which is appealing as it does not require the inversion of a high dimensional matrix, in the same way as Eq. (48) does. The Kemeny constant follows as

$$\zeta = \sum_j p_j^{\text{eq}} t_{ji} = \sum_j \left[ \frac{\int_0^\infty C_{jj}^{\text{eq}}(\tau) d\tau}{\pi_j} - \frac{\int_0^\infty C_{ji}^{\text{eq}}(\tau) d\tau}{\pi_i} \right]. \quad (50)$$

Since  $C_{ji}^{\text{eq}}(\tau)/\pi_i = P(j, \tau|i, 0) - \pi_j$  and  $\sum_j P(j, \tau|i, 0) = 1 \forall \tau$ , swapping sums with integrals in Eq. (50), which is valid for finite state space, it becomes clear that the second term on the RHS vanishes, giving

$$\zeta = \sum_j \frac{\int_0^\infty C_{jj}^{\text{eq}}(\tau) d\tau}{\pi_j} \equiv \sum_j D_{jj} \quad (51)$$

$$= \sum_j \int_0^\infty [P(j, \tau|i, 0) - p_j] d\tau \quad (52)$$

The first term in the square brackets measures the fraction of trajectories that are in  $j$  at time  $\tau$ , out of those that start in  $j$  at time 0. The second term measures the fraction of trajectories that are in  $j$  at a given time  $\tau$ , out of all the trajectories. Equation (52) reveals that Kemeny constant can be regarded as the time-integrated difference between the conditional and the a priori probability to be in any given state, as similarly pointed out in [40]. Furthermore, Eq. (51) shows that  $\zeta$  can be written as the trace of a matrix, that is known as the 'deviation matrix'  $\mathbf{D}$  [40, 41].

A more convenient writing of Eq. (52), which avoids its formulation in terms of the (finite) difference between two divergent integrals, can be obtained by introducing the decorrelation time of a state  $i$

$$T_i = \int_0^\infty \frac{C_{ii}^{\text{eq}}(\tau)}{C_{ii}^{\text{eq}}(0)} d\tau, \quad (53)$$

as the area underneath the normalised autocorrelation functions  $\hat{C}_{ii}^{\text{eq}}(\tau) = C_{ii}^{\text{eq}}(\tau)/C_{ii}^{\text{eq}}(0)$ . The latter takes values 1 for  $\tau = 0$  and zero for  $\tau \rightarrow \infty$ , and it decays as a multi-exponential, thus yielding a convergent integral. Using  $C_{ii}^{\text{eq}}(0) = \pi_i(1 - \pi_i)$ , one can express the Kemeny constant as in Eq. (54)

$$\zeta = \sum_i T_i (1 - \pi_i). \quad (54)$$

This leads to a simple interpretation of the Kemeny constant, as a weighted sum of the decorrelation times of the individual states. Here,  $1 - \pi_i$  can be thought of as the difference between the maximum value, 1, and the minimum value,  $\pi_i$ , of the conditional probability  $P(i, \tau|i, 0)$ , (attained at  $\tau = 0$  and  $\tau = \infty$  respectively), while  $T_i$  measures how fast  $P(i, \tau|i, 0)$  decays from the former to the latter value.

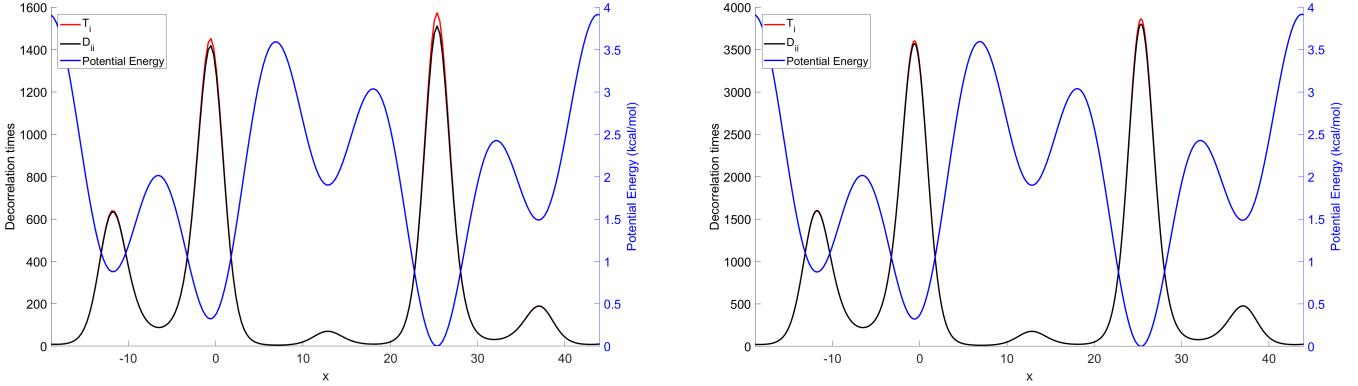


Figure 1: Decorrelation times  $T_i$  (red line) and diagonal entries of the deviation matrix  $D_{ii}$  (black line), as a function of the state  $i$ , with  $i = 1, \dots, n$ , for a system diffusing in the one-dimensional potential energy function  $v(x) = \sin((x - \pi)/2) + \sin((x - \pi)/5)$  (blue line). Transition rates are given by Eq. (315) with  $K_B T = 0.596$ . In the left panel, the number of states is  $n = 200$  and small deviations can be observed between  $D_{ii}$  and  $T_i$  for the states with highest equilibrium populations. In the right panel, where the number of states is 500, spatial resolution is higher and the two quantities are almost indistinguishable. Note the different scaling of the  $y$ -axis for the potential energy.

Note that for systems with a large number of states  $n$  and broad equilibrium distribution, one is normally interested in, individual state probabilities are small, i.e.  $\pi_i \ll 1 \forall j$ , hence

$$\zeta \simeq \sum_{i=1}^n T_i, \quad n \gg 1 \quad (55)$$

To check the validity of the approximation in Eq. (55), we consider dynamics in a potential energy function  $v(x)$ , along the continuous reaction coordinate  $x$  between finite boundaries,  $x_{\min}$  and  $x_{\max}$ . Upon discretizing the continuous problem, we obtain a discrete state Markov processes, where the number of states  $n$  is given by the number of discretization bins and the transition rates between adjacent states are given by the Arrhenius law

$$k_{ji} = A e^{-\frac{v(j)-v(i)}{k_B T}}. \quad (56)$$

In Figure (1) we plot the diagonal elements of the deviation matrix  $D_{ii} = T_i(1 - \pi_i)$  and the decorrelation times  $T_i$ , for different numbers of bins, namely  $n = 50$  and  $n = 200$ . We see that  $D_{ii}$  is almost indistinguishable from  $T_i$  for  $n = 200$ , while they show little deviation when the number of states is decreased to  $n = 50$ , the states with highest probabilities (i.e. with minimum potential energy) exhibiting the largest deviation. This confirms our intuition that, if the number of states is sufficiently high, the Kemeny constant converges to the sum of the decorrelation times of the individual states.

In conclusion, combining Eq. (54) and (40) provides an interesting chain of relations for MFPTs, relaxation times and decorrelation times

$$\sum_{j=1}^n \pi_j t_{ji} = \sum_{\ell=2}^n \tau_\ell = \sum_{j=1}^n T_j(1 - \pi_j) \approx \sum_{j=1}^n T_j \quad (57)$$

where the last approximation holds for large  $n$  and broad  $\pi$ .

### 2.3.3 Constructing rate matrices from MFPTs

With an explicit expression for MFPTs in terms of rate matrices, we can now invert this expression, to obtain a recipe for constructing rate matrices with given MFPTs. Upon defining  $\mathbf{z}$  as the vector with components  $z_j = [(\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})^{-1}]_{jj}$ , we can write Eq. (48) in matrix form

$$\mathbf{D}_n \mathbf{t} = \mathbf{z} \mathbf{1}_n^T - (\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})^{-1} \quad (58)$$

Rearranging, we obtain

$$\mathbf{K} = \boldsymbol{\pi} \mathbf{1}_n^T - (\mathbf{z} \mathbf{1}_n^T - \mathbf{D}_n \mathbf{t})^{-1}, \quad (59)$$

where  $\mathbf{z}$  can be expressed in terms of  $\mathbf{t}$  by demanding  $\mathbf{K}\boldsymbol{\pi} = 0$

$$\mathbf{z} = \boldsymbol{\pi} + \mathbf{D}_n \mathbf{t} \boldsymbol{\pi}. \quad (60)$$

Substituting into Eq. (59) then gives

$$\mathbf{K} = \boldsymbol{\pi} \mathbf{1}_n^T - [\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{D}_n \mathbf{t} (\mathbf{I} - \boldsymbol{\pi} \mathbf{1}_n^T)]^{-1}. \quad (61)$$

It is easy to show that Eq. (61) also satisfies  $\mathbf{1}_n^T \mathbf{K} = 0$ , by noting that  $\mathbf{1}_n^T \mathbf{D}_n = [\boldsymbol{\pi}]^T$  and

$$[\boldsymbol{\pi}]^T \mathbf{t} = \zeta \mathbf{1}_n^T, \quad (62)$$

which is implied by the definition of Kemeny constant in Eq. (40). Equation (62) also shows that the equilibrium distribution can be fully determined from the matrix of MFPTs, as  $[\boldsymbol{\pi}]^T = \zeta \mathbf{1}_n^T \mathbf{t}^{-1}$  where  $\zeta$  follows from the normalization of  $\boldsymbol{\pi}$ , as  $\zeta = 1/(\mathbf{1}_n^T \mathbf{t}^{-1} \cdot \mathbf{1}_n)$ , so

$$[\boldsymbol{\pi}]^T = \frac{\mathbf{1}_n^T \mathbf{t}^{-1}}{\mathbf{1}_n^T \mathbf{t}^{-1} \cdot \mathbf{1}_n}. \quad (63)$$

By using Eq. (62) and the Sherman-Morrison formula, as shown in Appendix (A), Eq. (61) can be simplified to obtain

$$\mathbf{K} = \mathbf{t}^{-1} (\mathbf{D}_n^{-1} - \mathbf{1}_n \mathbf{1}_n^T), \quad (64)$$

which could have also been derived from Eq. (38). Since  $\mathbf{D}_n$  follows directly from  $\boldsymbol{\pi}$ , Eq. (63) and Eq. (64) show that  $\boldsymbol{\pi}$  and  $\mathbf{K}$  can be both computed by inverting a single matrix (i.e.  $\mathbf{t}$ ).

These equations then give a recipe to infer the equilibrium probability and the rate matrix of a system with  $n$  states, from the *sole* observation of MFPTs between pairs of states. This may be useful in practical situations where information about MFPTs is readily available, whereas information about the rate matrix and the equilibrium distribution is not.

We note that in Markov processes with ordered states, reflecting boundary conditions, and transitions only occurring between adjacent states, one has, for any pair of states  $i < j$ ,  $t_{ij} = \sum_{k=i}^{j-1} t_{k,k+1}$ . Hence, the full matrix  $\mathbf{t}$  can be determined from the knowledge of only MFPTs between adjacent states,  $t_{k,k\pm 1}, \forall k$ . Equations (63) and (64) can then be used to reconstruct the full equilibrium distribution and rate matrix, from the observation of MFPTs between adjacent states, which can be computed efficiently, e.g. via the trajectory coloring procedure introduced in [42, 43]. This can be useful in milestoneing procedures, aimed at inferring the full kinetics of a system from the observation of many short trajectories, between adjacent states (milestones).

We note that for milestoneing on one-dimensional potentials, recipes to construct rate matrices have been given in terms of MFPTs *and* committor probabilities for adjacent milestones [18, 31]. Equation (64), equipped with Eq. (63), provides an alternative route which does not require to estimate committor probabilities. The above framework provides an intuitive explanation for the observed accuracy of milestoneing techniques, when applied to one-dimensional Smoluchowski processes, in predicting the full distribution of MFPTs, by using rate matrices constructed from MFPTs between adjacent milestones [18, 31]: for these processes, MFPTs between adjacent states are sufficient to construct the whole MFPTs matrix, which univocally determines the rate matrix and the equilibrium distribution, as shown by Eq. (63) and Eq. (64).

An interesting pathway for future research would be to find optimal recipes to infer the rate matrix  $\mathbf{K}$  and the equilibrium distribution  $\boldsymbol{\pi}$  from *partial* observations of the entries of matrix  $\mathbf{t}$ , for more general kinetic networks, where MFPTs between adjacent states do not encode the full distribution of MFPTs.

### 2.3.4 Construction transition probability matrices from MFPTs

For completeness, in this section we provide an expression for MFPTs in terms of transition matrices in discrete time dynamics, and, conversely, we show how to construct transition matrices and equilibrium distributions from MFPTs. Starting with Eq. (31), rewriting  $\lambda'_\ell / (1 - \lambda'_\ell) = 1 / (1 - \lambda'_\ell) - 1$ , using the spectral representation of the identity matrix element  $I_{jk} = \sum_\ell \psi_j^{(\ell)} \phi_k^{(\ell)}$  and repeating the same reasoning that led to Eq. (48), we obtain

$$t'_{jk} = \frac{1}{\pi_j} \left[ I_{jk} + (\boldsymbol{\pi} \mathbf{1}_n^T + \mathbf{I} - \mathbf{M})_{jj}^{-1} - (\boldsymbol{\pi} \mathbf{1}_n^T + \mathbf{I} - \mathbf{M})_{jk}^{-1} \right] \quad (65)$$

Similarly to Eq. (58), this can be cast in vector notation

$$\mathbf{D}_n \mathbf{t} = \mathbf{I} + \mathbf{z}' \mathbf{1}_n^T - (\boldsymbol{\pi} \mathbf{1}_n^T + \mathbf{I} - \mathbf{M})^{-1} \quad (66)$$

where  $z'_j = [(\boldsymbol{\pi} \mathbf{1}_n^T + \mathbf{I} - \mathbf{M})^{-1}]_{jj}$ . Rearranging for  $\mathbf{M}$  and requiring  $\mathbf{M}\boldsymbol{\pi} = \boldsymbol{\pi}$  gives  $\mathbf{z}' = \mathbf{D}_n \mathbf{t} \boldsymbol{\pi}$  and

$$\mathbf{M} = \mathbf{I} + \boldsymbol{\pi} \mathbf{1}_n^T - (\mathbf{I} - \mathbf{D}_n \mathbf{t}' + \mathbf{D}_n \mathbf{t}' \boldsymbol{\pi} \mathbf{1}_n^T)^{-1}. \quad (67)$$

Equation ((63)) remains true for  $\mathbf{t}'$ , as from ((62)) one has  $[\boldsymbol{\pi}]^T \mathbf{t}' = (1 + \zeta') \mathbf{1}_n^T$ , with  $1/(1 + \zeta') = \mathbf{1}_n^T \mathbf{t}'^{-1}$  following from normalization of  $\boldsymbol{\pi}$ , so

$$[\boldsymbol{\pi}]^T = \frac{\mathbf{1}_n^T \mathbf{t}'^{-1}}{\mathbf{1}_n^T \mathbf{t}'^{-1} \cdot \mathbf{1}_n}. \quad (68)$$

However, in discrete time dynamics, the equilibrium distribution can also be directly read off from the diagonal elements of  $\mathbf{t}'$ , as shown in (25).

An alternative expression for the transition matrix  $\mathbf{M}$ , can be obtained by setting  $\mathbf{M} = e^{\mathbf{K}\tau}$  in Eq. (33). Rewriting this in vector notation

$$\mathbf{t}'(\mathbf{I} - \mathbf{M}) = \mathbf{1}_n \mathbf{1}_n^T - \mathbf{D}_n^{-1} \mathbf{M} \quad (69)$$

and rearranging for  $\mathbf{M}$  gives

$$\mathbf{M} = (\mathbf{I} - \mathbf{D}_n \mathbf{t}')^{-1} (\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{D}_n \mathbf{t}'). \quad (70)$$

It can be easily shown that Eq. (67) and Eq. (70) coincide, by multiplying Eq. (67) times  $(\mathbf{I} - \mathbf{D}_n \mathbf{t}' + \mathbf{D}_n \mathbf{t}' \boldsymbol{\pi} \mathbf{1}_n^T)$  from left, expanding the products and using  $\mathbf{1}_n^T \mathbf{M} = \mathbf{1}_n^T$ . In conclusion, like rate matrices, transition matrices can be computed by inverting a single matrix, involving MFPTs, e.g.  $\mathbf{I} - \mathbf{D}_n \mathbf{t}'$ .

### 2.3.5 Conclusion

In this study we have presented and linked together several results existing in the literature for mean first passage times and the Kemeny constant and have provided new relations in terms of correlation functions. These relations lead to a new writing of the Kemeny constant, and a simple interpretation in terms of decorrelation times.

In addition, we have provided a recipe to infer the equilibrium distribution and the rate matrix of a process, from measurements of MFPTs. This does not require the estimation of committor probabilities and it only requires the inversion of a single matrix (with MFPTs between pairs of states as entries). For systems whose transitions are well approximated by memoryless jumps between adjacent states, as the one dimensional Smoluchowski process, MFPTs between any pair of states can be expressed in terms of MFPTs between adjacent states, hence the rate matrix can be constructed from the sole measurements of MFPTs between adjacent states, using this recipe.

This observation provides an intuitive explanation for the accuracy of milestoneing techniques in inferring the whole MFPTs distribution, from short trajectories between adjacent milestones, which has been pointed out in [18, 31]. An interesting pathway for future work would be to define optimal recipes to infer rate matrices, from partial observations of MFPTs, in more complex kinetic networks, where MFPTs between adjacent states are not sufficient to reconstruct the full MFPTs matrix.

The derived relation between rate matrices and MFPTs, given in equation Eq. (64), may find application in several domains. For example, in transport networks, the mean travelling times of passengers between two stations (a proxy for MFPTs), may be readily available from smart cards, and can be used to infer the rates at which passengers move along the links of the network, which might be more difficult to measure in practice. Often, a simple diffusive process (controlled by the degrees of the nodes) is assumed, but due to the varying importance of different nodes, this assumption may be invalid [44]. Equation (64) may thus be used to model such processes more accurately.

Another application we can mention, is the inference of gene regulatory networks from the time series generated in gene knock-out experiments [45], which provide information on the first time at which the expression of a gene  $j$  is modified, as a result of knocking out a gene  $i$ . This can be regarded as the MFPT to reach node  $j$  from node  $i$  on the relevant gene regulatory network. Using this information, an effective rate matrix can be computed via Eq. (64), which may give information on the rate at which a perturbation of gene  $i$  propagates to gene  $j$ , thus providing insights on the interactions between genes.

## 2.4 RESULT 2 (Unpublished): MFPT definition for continuous times

### 2.4.1 Introduction

In section 2.2.4.1 we have defined a recursive formula for MFPT in discrete time, and to obtain the equivalent for continuous time, in Sec. 2.2.4.2, we took the limit when  $\tau \rightarrow 0$ .

This approach relies on a spectral decomposition of the rates matrix  $\mathbf{K}$ , and requires the existence of a complete set of orthonormal eigenvectors to obtain the provided result.

In the present section we will take an alternative route, by defining the MFPT directly in continuous time, and show that we are able to recover the same expression for  $\mathbf{K}$  in terms of MFPT, without the use of spectral decomposition, and without assuming Detailed Balance.

When considering  $t_{ii}$ , in continuous time, two conventions are possible: either the time must be zero because we are already in  $i$ , alternatively the time must positive because we require to first leave the state  $i$  and then re-enter it. In Sec. 2.2.4.1 we use the former convention. We will call these conventions "Reach" and "Entry" conventions respectively in the following, and will provide the definition for MFPT in continuous time for both cases.

Moreover, to contribute further to the framework involving MFPT in the study of kinetic networks, we will express the condition of Detailed Balance and Global Balance purely in terms of MFPT and equilibrium probabilities, and provide the resulting relations.

### 2.4.2 Continuous time MFPT with "Entry" convention

Using the same idea as in the previous section, we can define a recursive equation for the MFPT in terms of rates:

$$t_{ji} = MFTL(i)p(\rightarrow j|i \rightarrow) + \sum_{\alpha \neq j,i} (MFTL(i) + t_{j\alpha})p(\rightarrow \alpha|i \rightarrow) \quad (71)$$

Here the idea is exactly the same as for the discrete case:

Given that we start in state  $i$  and leave it to reach  $j$ , we will either directly reach  $j$  upon leaving  $i$ , and the elapsed time will be only the Mean First Time to Leave  $i$ , ( $MFTL(i)$ ) (first term on r.h.s.), or we will reach another state  $\alpha$  different from  $i$  and  $j$ , also after waiting the time to leave  $i$ , and then add the time to transition from  $\alpha$  to  $j$  (second term on r.h.s.).

Note that by first-order kinetics approximation of rate dynamics,  $-\frac{k_{ji}}{k_{ii}}$  is defined as the conditional transition probability to transition to state  $j$  given that we just left state  $i$ :

$$p(\rightarrow j|i \rightarrow) \equiv -\frac{k_{ji}}{k_{ii}} \quad (72)$$

Inserting Eq. (72) into Eq. (71) and rearranging, we obtain:

$$t_{ji} = MFTL(i) + \sum_{\alpha \neq j,i} t_{j\alpha} \frac{k_{\alpha i}}{-k_{ii}} \quad (73)$$

Where the interpretation is even simpler: to leave  $i$  we need to first wait  $MFTL(i)$  on average, then we either transition directly to  $j$  and stop measuring time, or to another state  $\alpha$ , different from  $j$  and  $i$ , and add the average time to get from  $\alpha$  to  $j$ , weighted by the probability to reach  $\alpha$  after leaving  $i$ .

#### 2.4.2.1 The matrix $[t\mathbf{K}]$ with "Entry" convention

Applying Eq. (71) to the case  $j \neq i$  and  $j = i$  separately,

$$\begin{cases} t_{ji} = MFTL(i) + \sum_{\alpha \neq j,i} t_{j\alpha} \frac{k_{\alpha i}}{-k_{ii}} \\ t_{ii} = MFTL(i) + \sum_{\alpha \neq i} t_{i\alpha} \frac{k_{\alpha i}}{-k_{ii}} \end{cases} \quad (74)$$

using  $MFTL(i) = -1/k_{ii}$  and multiplying both sides by  $-k_{ii}$ , we obtain:

$$\begin{cases} -t_{ji}k_{ii} = 1 + \sum_{\alpha \neq j,i} t_{j\alpha}k_{\alpha i} \\ -t_{ii}k_{ii} = 1 + \sum_{\alpha \neq i} t_{i\alpha}k_{\alpha i} \end{cases} \quad (75)$$

Now completing the sum on the r.h.s. yields:

$$\begin{cases} -t_{ji}k_{ii} = 1 + \sum_{\alpha} t_{j\alpha}k_{\alpha i} - t_{ji}k_{ii} - t_{jj}k_{ji} \\ -t_{ii}k_{ii} = 1 + \sum_{\alpha} t_{i\alpha}k_{\alpha i} - t_{ii}k_{ii} \end{cases} \quad (76)$$

Which rearranged yields the expression of the element of the matrix  $tK$ :

$$\sum_{\alpha} t_{i\alpha}k_{\alpha i} = (tK)_{ji} = \begin{cases} t_{jj}k_{ji} - 1 & j \neq i \\ -1 & j = i \end{cases} \quad (77)$$

The "Entry" convention of the present section is not our primary interest and therefore we will focus on the "Reach" convention, where  $t_{ii} = 0$ .

#### 2.4.3 Continuous time MFPT with "Reach" convention

We can adapt the definition of MFPT to the "Mean First Reaching Time" convention, which implies  $t_{ii} = 0$ , by including a Kronecker delta ( $\delta_{ij}$ ):

$$t_{ji} = \left[ MFTL(i) + \sum_{\alpha \neq j,i} t_{j\alpha} \frac{k_{\alpha i}}{-k_{ii}} \right] (1 - \delta_{ij}) \quad (78)$$

Inserting  $MFTL(i) = -\frac{1}{k_{ii}}$  into Eq. (78)

$$\Leftrightarrow -k_{ii}t_{ji} = \left[ 1 + \sum_{\alpha \neq j,i} t_{j\alpha}k_{\alpha i} \right] (1 - \delta_{ij}) \quad (79)$$

Completing the sum and expanding the factorisation we get:

$$\Leftrightarrow 0 = \left[ 1 + \sum_{\alpha} t_{j\alpha}k_{\alpha i} - t_{jj}k_{ji} \right] - \delta_{ij} \left[ 1 + \sum_{\alpha} t_{j\alpha}k_{\alpha i} - t_{jj}k_{ji} - t_{ji}k_{ii} \right] \quad (80)$$

Now we can use the fact that  $t_{ii} = 0$  by construction and simplify.

$$\sum_{\alpha} t_{j\alpha}k_{\alpha i} = -1 + \delta_{ij} \left[ 1 + \sum_{\alpha} t_{j\alpha}k_{\alpha i} - t_{ji}k_{ii} \right] \quad (81)$$

From this we directly see that the off-diagonal elements of the matrix  $tK$  are equal to  $-1$ .

To find the value of the diagonal elements we can multiply the above equation by  $\pi_i$  and sum over  $i$  ('multiply from the right'):

$$0 = -1 + \pi_j \left[ 1 + \sum_{\alpha} t_{j\alpha}k_{\alpha j} - t_{jj}k_{jj} \right] \quad (82)$$

Using again  $t_{ii} = 0$  and simplifying, we obtain

$$\sum_{\alpha} t_{j\alpha}k_{\alpha j} = \frac{1}{\pi_j} - 1 \quad (83)$$

Hence we see that the elements of the matrix  $W = tK$  can be defined as

$$(tK)_{ji} = \sum_{\alpha} t_{j\alpha}k_{\alpha i} = \begin{cases} \frac{1}{\pi_j} - 1 & \text{if } i = j \\ -1 & \text{else} \end{cases} \quad (84)$$

In matrix form we clearly recover Eq.(38) equivalent to Eq.(64):

$$tK = -1_n 1_n^T + D_n^{-1} \quad \text{Eq.(38)} \quad (85)$$

$$\Leftrightarrow K = t^{-1} [D_n^{-1} - 1_n 1_n^T] \quad \text{Eq.(64)} \quad (86)$$

Interestingly, we did not have to use the assumption of Detailed Balance to arrive at this equation, because we did not involve the eigendecomposition of the Rates matrix  $\mathbf{K}$ .

Instead we defined the MFPT recursive equation in terms of Rates directly in a meaningful way.

Moreover, we can obtain another interesting relation from Detailed Balance

#### 2.4.4 Detailed Balance in terms of Mean First Passage Times

(Using "Reach" convention)

The Detailed Balance relation  $K_{ji}\pi_i = K_{ij}\pi_j$  can be written in matrix form:

$$\mathbf{K}\mathbf{D}_N = (\mathbf{K}\mathbf{D}_N)^T = \mathbf{D}_N\mathbf{K}^T \quad (87)$$

We can rewrite this relation in terms of MFPT, using the expression of  $\mathbf{K}$  in Eq. (86):

$$\Leftrightarrow \mathbf{t}^{-1}(\mathbf{D}_N^{-1} - \mathbf{1}_N\mathbf{1}_N^T)\mathbf{D}_N = \mathbf{D}_N [\mathbf{t}^{-1}(\mathbf{D}_N^{-1} - \mathbf{1}_N\mathbf{1}_N^T)]^T \quad (88)$$

Which simplifies to

$$\mathbf{t}^{-1}(I - \mathbf{1}_N\pi^T) = (I - \boldsymbol{\pi}\mathbf{1}_N^T)(\mathbf{t}^T)^{-1} \quad (89)$$

Multiplying both sides by  $\mathbf{t}$  from the left and  $\mathbf{t}^T$  from the right, and rearranging, we obtain an expression involving  $\mathbf{t}$  instead of its inverse:

$$\mathbf{t} + \mathbf{1}_N\pi^T\mathbf{t}^T = \mathbf{t}^T + \mathbf{t}\pi\mathbf{1}_N^T \quad (90)$$

Defining the new quantity  $\mathbf{O}$  as follows,

$$\mathbf{O} \equiv \mathbf{t} + \mathbf{1}_N\pi^T\mathbf{t}^T \quad (91)$$

And inserting it in Eq. (90) we highlight how the RHS is equal to the transpose of the LHS, as the equation above becomes:

$$\mathbf{O} = \mathbf{O}^T \quad (92)$$

Finally, using  $T_{ii} = 0$  from the "Reach" convention, for an arbitrary couple of disjoint states (i,j) we have the following relation:

$$O_{ij} = O_{ji} \Leftrightarrow t_{ij} + \sum_{\alpha} t_{j\alpha}\pi_{\alpha} = t_{ji} + \sum_{\alpha} t_{i\alpha}\pi_{\alpha} \quad (93)$$

Which is equivalent to the following relation for an arbitrary number of states:

$$\Leftrightarrow t_{ji}[1 - \pi_i] - t_{ij}[1 - \pi_j] = \sum_{\alpha \neq i,j} [t_{j\alpha} - t_{i\alpha}]\pi_{\alpha} \quad (94)$$

##### 2.4.4.1 2-State case or cluster all except one

For the 2-States case, the term on the RHS of Eq. (94) vanishes and the relation reduces to:

$$\frac{t_{ji}}{t_{ij}} = \frac{1 - \pi_j}{1 - \pi_i} = \frac{\pi_i}{1 - \pi_i} \quad (95)$$

If we define  $C \equiv \{\Omega \setminus i\}$  as the complement of  $i$  (cluster of all states except  $i$ ), the relation in Eq. (94) gives:

$$\frac{T_{Ci}}{T_{iC}} = \frac{\pi_i}{1 - \pi_i} \quad (96)$$

Now, we can use the well known expression for the mean time to leave the state  $i$ :  $T_{Ci} = -\frac{1}{k_{ii}}$

$$T_{iC} = \frac{1 - \pi_i}{-k_{ii}\pi_i} \quad (97)$$

Which is the Mean First Time time to reach state  $i$  given that the system is not in  $i$ .

#### 2.4.5 Mean First Observation Time: an intuitive way to describe Detailed Balance in terms of MFPT

##### 2.4.5.1 Motivation - The usual MFPT is implicitly conditional on a starting state

The usual Mean First Passage Time ( $t_{ji}$ ) is defined as the first time on average to reach the state  $j$ , conditional on starting in  $i$ . For clarity, we denote it as  $MFPT(i \rightarrow j|i) = t_{ji}$ .

We can consider a more general quantity, conditional on starting in an arbitrary state  $\alpha$ , which is expressed as follows:

$$MFPT(i \rightarrow j|\alpha) = MFPT(\alpha \rightarrow i|\alpha) + MFPT(i \rightarrow j|i) = t_{i\alpha} + t_{ji} \quad (98)$$

This expression can be understood as follows: the system starts in state  $\alpha$ , and needs to first reach state  $i$ , and then reach state  $j$ .

Note that here we naturally use the "Reach" convention  $t_{ii} = 0$ , and when  $\alpha = i$  the expression reduces to  $t_{ji}$ .

#### 2.4.5.2 MFOT: definition and mathematical expression

Another quantity of interest is the first time on average to observe the system (or a particule in the system) transition from state  $i$  to  $j$ , not conditional on starting in  $i$ .

Another quantity of interest is the first time on average to observe the system (or a particule in the system) transition from state  $i$  to  $j$ , not conditional on starting in  $i$ . We will call this quantity "Mean First Observation Time" (MFOT), and will show that an intuitive definition of MFOT leads to the mathematical expression of  $O$  in Eq. (91)

Formally, we define the Mean First Observation Time as the average time to wait to observe a transition from  $i$  to  $j$  of a system sampled following the equilibrium (stationary) distribution  $\pi$ :

$$MFOT(i \rightarrow j) = \sum_{\alpha} MFPT(i \rightarrow j|\alpha)\pi_{\alpha} \quad (99)$$

Using the expression for  $MFPT(i \rightarrow j|\alpha)$  defined above, we obtain

$$MFOT(i \rightarrow j) = t_{ji} + \sum_{\alpha} t_{i\alpha}\pi_{\alpha} \quad (100)$$

Which, in matrix form, is written as

$$MFOT(i \rightarrow j) = \mathbf{t} + \mathbb{1}_n \boldsymbol{\pi}^T \mathbf{t}^T = O \quad (101)$$

We now recognize that  $MFOT(i \rightarrow j)$  is indeed equal to the matrix  $O$  defined in Eq. (91).

#### 2.4.5.3 Easing the inversion of matrix $\mathbf{t}$ ?

Starting from Eq. (90), and rearranging we have

$$\mathbf{t}(I_n - \boldsymbol{\pi} \mathbb{1}_n^T) = (I_n - \mathbb{1} \boldsymbol{\pi}_T) \mathbf{t}^T \quad (102)$$

Where the Left Hand Side, that we will denote  $\mathbf{A}$ , is equal to the transpose of the RHS. Hence  $\mathbf{A}$  is a symmetric matrix.

Left-multiplying by  $t^{-1}$ , and right-multiplying by  $\mathbf{A}^{-1}$ , we obtain:

$$\mathbf{t}^{-1} = (I_n - \boldsymbol{\pi} \mathbb{1}_n^T) \mathbf{A}^{-1} \quad (103)$$

If computing the inverse of a symmetric matrix is easier than a non-symmetric one, then this formula can be used to compute the inverse of  $\mathbf{t}$  more efficiently, when needed..

#### 2.4.5.4 Detailed Balance in terms of MFOT (intuitive)

Detailed Balance means that the flux between two states is equal at equilibrium.

The flux being quantified in terms of transition counts per unit of time, its inverse is the average time per transition count, and if one quantity is equal (between two states), the other is too.

The average time per transition count is exactly what MFOT represents, therefore Detailed Balance is equivalent to the equality of MFOT between any pair of disjoint states:

$$DB \Leftrightarrow MFOT(i \rightarrow j) = MFOT(j \rightarrow i) \quad (104)$$

Inserting the expression of MFOT:

$$t_{ji} + \sum_{\alpha} \pi_{\alpha} t_{i\alpha} = t_{ij} + \sum_{\alpha} \pi_{\alpha} t_{j\alpha} \quad (105)$$

Rearranging the relation:

$$\Leftrightarrow t_{ji} - t_{ij} = \sum_{\alpha} \pi_{\alpha} [t_{j\alpha} - t_{i\alpha}] \quad (106)$$

Using the convention  $t_{ii} = 0$

$$t_{ji} [1 - \pi_i] - t_{ij} [1 - \pi_j] = \sum_{\alpha \neq i,j} \pi_{\alpha} [t_{j\alpha} - t_{i\alpha}] \quad (107)$$

We retrieve the result in Eq. (94) from the previous section.

#### 2.4.5.5 Global Balance in terms of MFOT

Can we prove that Kemeny is constant using only the Global Balance condition?

From the perspective of Mean First Passage Times, GB shall correspond to the equality between the average time to observe an Entering event in  $j$  and the average time to observe a Leaving event from  $j$ .

We can write the Mean First Time (MFT) of both events:

$$MFT(\{\rightarrow j\}) = MFT(\{j \rightarrow\}) \quad (108)$$

From the perspective of a random initial position of the system when we start recording time, we have that:

The average time to observe the system leaving  $j$  is the average time to see the system reach  $j$  + the mean time to leave  $j$ .

$$MFT(\{j \rightarrow\}) = \sum_{\alpha \neq j} MFPT(\{\alpha \rightarrow j | \alpha\}) \pi_{\alpha} + MFLT(j) \quad (109)$$

The average time to observe the system entering  $j$  is the average time to see the system reach  $j$  when not in  $j$  + the Mean First Time to leave and re-Enter  $j$  weighted by the probability to be in  $j$ :

$$MFT(\{\rightarrow j\}) = \sum_{\alpha \neq j} MFPT(\{\alpha \rightarrow j | \alpha\}) \pi_{\alpha} + \pi_j MFET(j|j) \quad (110)$$

Inserting both expression into the equation of Global Balance leads to:

$$MFT(\{\rightarrow j\}) = MFT(\{j \rightarrow\}) \Leftrightarrow \pi_j MFET(j|j) = MFLT(j) \quad (111)$$

Which is equivalent to:

$$MFET(j|j) = \frac{MFLT(j|j)}{\pi_j} = -\frac{1}{k_{jj}\pi_j} \quad \text{or} \quad \frac{1}{\pi_j(1 - M_{jj})} \quad (112)$$

Where  $MFET(j|j)$  is the Mean First Entry Time into node  $j$  given that we are in node  $j$ .

Alternatively, if from Eq. (111) we expand the term  $MFET(j|j)$  using its definition:

$$MFET(j|j) = MFLT(j) + \sum_{\alpha \neq j} t_{j\alpha} \frac{k_{\alpha j}}{-k_{jj}} \quad (113)$$

And using  $T_j = -1/k_{jj}$  we obtain the following equation:

$$\frac{1}{-k_{jj}} = \pi_j \left( \frac{1}{-k_{jj}} + \sum_{\alpha \neq j} t_{j\alpha} \frac{k_{\alpha j}}{-k_{jj}} \right) \quad (114)$$

Which after multiplying by  $-k_{kk}$  and rearranging becomes

$$\frac{1}{\pi_j} = 1 + \sum_{\alpha \neq j} t_{j\alpha} k_{\alpha j} \quad (115)$$

$$\Leftrightarrow \frac{1}{\pi_j} - \sum_{\alpha \neq j} t_{j\alpha} k_{\alpha j} = 1 \quad (116)$$

Which is equivalent to the expression for the diagonal elements of the matrix  $\mathbf{tK}$  in Eq. (38).

We can see that the LHS should be dependent on  $j$ , but is in fact equal to one for any  $j$ . This might indicate that there is a way to prove that Kemeny constant is constant from here, but requires further work.

#### 2.4.6 Side note / Future - W matrix and similarity to M matrix

Consider the Kemeny constant

$$\zeta = \sum_j T_{ij} \pi_j \quad (117)$$

This relation can be translated into matrix notation:

$$\boldsymbol{\pi}^T \mathbf{t} = \zeta [\mathbb{1}]^T \quad (118)$$

Which can be rewritten an and eigenvalue equation:

$$W[\mathbb{1}] = K[\mathbb{1}] \quad (119)$$

Upon definition of the matrix  $\mathbf{M}$  as

$$W \equiv D_\pi T \quad \text{And} \quad (W)_{ji} = \pi_j T_{ji} \quad (120)$$

where  $D$  is a diagonal matrix with  $\pi$  as entries, we directly find that  $\mathbf{M}$  have  $\zeta$  as an eigenvalue associated to the left-eigenvector  $[\mathbb{1}]$ .

What is the Right eigenvector associated to  $K$ ?

Considering the right eigenvector  $\mathbf{v}$  associated to  $\zeta$ :

$$\mathbf{Wv} = \zeta \mathbf{v} \quad (121)$$

Element-wise, we have

$$(Wv)_j = \zeta v_j \quad (122)$$

Replacing  $\zeta$  by the sum:

$$\sum_z \pi_j T_{jz} v_z = (\sum_z \pi_z T_{z\alpha}) v_j \quad \forall \alpha \quad (123)$$

The equation holds for any index  $\alpha$ , and summing over  $j$  leads to  $1 = 1$ .

We have observed experimentally that the matrix  $W$  have only  $\zeta$  as positive eigenvalue, and the other eigenvalues  $\lambda$  are equal to  $-\tau_i$ . Hence this matrix contains all the timescales of the system, and its Trace is equal to zero.

It is interesting to see how this matrix  $\mathbf{M}$  have similar properties to the rates matrix  $\mathbf{K}$ , and this matrix might be the object of future work.

#### 2.4.7 Conclusion

We have introduced an alternative definition of MFPT from Rates matrix for both possible convention relative to self-transitions. We have shown that it result in the same relation between transition rates and the MFPT, without the use of the Detailed Balance assumption nor spectral decomposition of the rates matrix. We have shown that the Detailed Balance condition usually defined in terms of transition rates is equivalent to the symmetry of the Mean First Observation Times (MFOT), which is intuitively constructed from MFPT. Moreover, we have found a simple expression for the self-transition MFPT in the "Entry" convention of MFPT (average time to leave and re-enter).

### 3 Enhanced Sampling Problem

#### 3.1 Introduction (Abstract)

Consider a rigged coin to outcome tail most of the time, and need to figure out the actual probability to get head. Is it 1 over 100 flips? over 1 million? 1 billion?

The straightforward approach would be to toss the coin many times and count the number of head and tails, until the statistical error is sufficiently small. This approach is called sampling.

However, if the probability is 1 over 1 million, it would require approximately 116 days of tossing a coin every 10 seconds to observe the first head, on average. To tackle this problem, enhanced sampling have been invented.

Consider now that we are able to change the value of getting head by multiplying it by any positive number  $b$  of our choice. Denoting the original probability of getting head to be  $p_h$ , we would have the new head probability  $p_h^b$  written as

$$p_h^b \sim p_h * b \quad (124)$$

$$p_h^b = \frac{p_h * b}{p_h * b + (1 - p_h)} \quad (125)$$

Now the task at hand is, having the value of  $p_h$  unknown, to design an optimal way to set the value of  $b$  in order to find the value of  $p_h$ , with sufficiently small uncertainty, as quickly as possible.

This quantity  $b$  is called "bias", as it biases the results, but if we do it properly and keep track of the value  $b$ , we can then "unbias" the sampled data to obtain the true value of getting head,  $p_h$ .

Multiple methods have been developed that address the sampling problem, such as umbrella sampling, metadynamics and parallel tempering.[46–51] Depending on the biasing methods, a relevant statistical estimator must be used.

#### 3.2 Introduction (Technical)

Here we collate and describe a variety of results and pertaining to the Enhanced Sampling framework.

##### 3.2.1 Umbrella Sampling

Umbrella Sampling (US) is among the most commonly used methods for enhanced sampling of molecular dynamics simulations. In this method, multiple relatively short simulations are generated. The potential of mean force is biased via adding a quadratic function, which forces the system to fluctuate around a certain position.

Formally, we note the biased potential as follows

$$V^{(u)}(x) = V(x) + u(x) \quad \text{and} \quad u(x) = f_u(x_u - x)^2 \quad (126)$$

Where  $u(x)$  is the bias function, which is called "umbrella" because of its shape resembling an umbrella turned upside down,  $x_u$  and  $k_u$  are the position and force parameters of the umbrella, respectively.

This bias is useful to optimize the sampling along a certain reaction coordinate, or path, as it forces each simulation to sample its own region of the configuration space.

##### 3.2.2 WHAM

One commonly used approach to estimate the PMF from biased Molecular Dynamics simulation is the Weighted Histogram Analysis Method (WHAM)[52]. WHAM assumes that each simulation have sampled the system at equilibrium, and that adjacent umbrella simulations have sampled positions with enough overlap (??). However these assumption do not always hold, which can lead to significant estimation errors. Moreover, WHAM only estimates the equilibrium probability, and not the transition rates or transition probabilities, and therefore do not allow to fully grasp the processes of the dynamic system.

### 3.2.3 DHAM

The Dynamic Histogram Analysis Methods (DHAM)(??) have been developed to overcome these limitations. Based on the Markov assumption, DHAM provides a formula to estimate the unbiased transition probabilities matrix  $\mathbf{M}$  from a set of biased simulations.

The DHAM result is obtained by maximizing the log-Likelihood of observing the sampled transitions over all simulations (with their own bias), identified with the index  $s$

$$\log - L = \ln \prod_{s=1}^{n_s} \prod_{j=1}^m \prod_{i=1}^m (M_{ji}^{(s)})^{T_{ji}^{(s)}} \quad (127)$$

Where  $n_s$  and  $m$  are the number of simulations and bins (states) respectively, and  $T_{ji}^{(s)}$  is the number of sampled transitions from  $i$  to  $j$  in simulation  $s$ .

$M_{ji}^{(s)}$  denotes the biased probability, and is assumed to be of the following form:

$$M_{ji}^{(s)} = M_{ji} c_{ji}^{(s)} f_i^{(s)} \quad (128)$$

Where  $c_{ji}^{(s)}$  is a multiplicative factor applied to the transition probability, resulting from the bias  $s$ , and  $f_i^{(s)}$  acts as the normalization of the biased probabilities in simulation  $s$  as it is found to be equal to  $(\sum_j M_{ji} c_{ji}^{(s)})^{-1}$ .

Deriving the log-Likelihood in Eq. (??) with respect to the parameters, and using the Detailed Balance condition, the authors find the following solution for the unbiased transition probabilities:

$$M_{ji} = \frac{\sum_s T_{ji}^{(s)}}{N_i^{(s)} \exp(-(u_j^{(s)} - u_j^{(s)} i)/(2k_B T)} \quad (129)$$

We will provide a similar result for transition rates.

### 3.2.4 Estimation error of the PMF: minRMSD

To compare the estimated PMF to the ground truth, denoted  $V^{est}$  and  $V^{true}$  respectively, one can use the KL-divergence on the equilibrium probabilities (estimated against ground truth). However this measure is not symmetric, and is not always numerically stable, for this reason in this work we use the RMSD of the PMFs instead.

To account for the fact that we might have empty (non-sampled) entries in the vector of estimated PMF  $V^{est}$ , and effectively compare two vectors of same length, we apply the RMSD formula only to the non-empty indeces of  $V^{est}$ :

$$RMSD(V^{est}, V^{true}) = \sqrt{\sum_{i=1}^{N^{est}} \frac{(V^{est}(i) - V^{true}(i))^2}{N^{est}}} \quad (130)$$

Where  $N^{est}$  is the number of non-empty entries of  $V^{est}$ . We can see that adding a constant to all elements of one of the vectors will change the RMSD, but we would like to measure the distance in *shape* between x and y, hence we will use a variant of RMSD which is invariant to translations of either x or y, *minRMSD*:

$$minRMSD(V^{est}, V^{true}) = min_c RMSD(V^{est} + c, V^{true}) \quad (131)$$

Expanding and taking the derivative with respect to  $c$ , we have a formula for the optimal value of  $c$  to use:

$$c = -\frac{\sum_i^{N^{est}} (V^{est}(i) - V^{true}(i))}{N^{est}} \quad (132)$$

## 3.3 RESULT 1 (Unpublished) ESTIMATOR - DHAR, Statistical uncertainty, and Markov Neural Networks

### 3.3.1 Introduction

Previously proposed approaches to estimating the kinetics of a dynamical system from biased trajectories have failed to deliver an explicit expression for the optimal estimation of transition probability matrices that would satisfy the Detailed Balance (DB) condition. Here, we provide a solution to this problem, by estimating the optimal transition rates matrix satisfying the DB constraint from a set of biased trajectories, from which

the transition probability matrix at any arbitrary lagtime can be obtained. We demonstrate that this method allows to successfully recover the non-biased kinetic rates from a set of short trajectories, for various simulations algorithms such as Gillespie, Monte Carlo, Gibbs sampling and Langevin dynamics.

Inspired by E.Rosta and G. Hummer's approach to the likelihood maximization of kinetic rates [53, 54], we are providing a derivation of the max-Likelihood estimator for the entries of the Rates matrix  $\mathbf{K}$ , under the constraint of Detailed Balance.

The core idea is to change our perspective about what we call a 'transition': before we were measuring the position of the system following an arbitrary lag-time and considered everything as a transition.

An alternative approach that have been mentioned and used in the paper cited above, is to only consider "leaving the current bin/micro-state for another" as a transition (staying in the same state is not a transition).

Hence we want to separately measure the time that the system spent in a state before leaving it, for each transition, and the destination of the transition.

This can allow us to estimate the micro-state lifetimes (Mean First Time to Leave) and transition probabilities conditioned on leaving a particular state, respectively.

(Therefore  $p(-j|i-) = 0$  if  $(j=i)$  in this approach, and the conditional term will be dropped for simplicity below but it does exist).

Using a first order approximation of rate dynamics allows to cancel some terms out and obtain an optimal expression for the rates as will be shown in detail below.

However, there is an implicit assumption that we are able to measure ALL transitions (our measuring lag-time is smaller than the smallest lifetime of the system).

It can be an important problem, in multiple ways: in addition to overestimating the timescale of the fastest events, we estimate some rates as non-zero while they are zero in reality. ('jumping over states' in a linear chain is a good example that can be easily studied).

In practice it can be dealt with by increasing the size of the micro-states a posteriori, if the measuring lag-time cannot be reduced.

In what follows, we will first derive the optimal estimator in the case of a single non-biased trajectory, in order to show key elements of the derivation. Then we will go through the derivation of the solution for biased trajectories, with and without a constraint for Detailed Balance, i.e. reversibility.

### 3.3.2 Single Non-Biased Trajectory

At first, consider the simplest case of a Non-biased trajectory.

#### 3.3.2.1 Construction of the likelihood function

The likelihood is constructed as the product of two key parts: the probability of the transition ending in state  $(j_\alpha)$ , conditional on leaving the current state  $(i_\alpha)$  ( $p(\rightarrow j_\alpha | i_\alpha \rightarrow)$ ), and the probability to stay  $t_\alpha$  time in state  $i_\alpha$  before leaving it (lifetime probability) ( $p(t_\alpha | i_\alpha)$ ), over all transitions  $\alpha$ :

$$L = \prod_{\alpha} p(\rightarrow j_\alpha | i_\alpha \rightarrow) p(t_\alpha | i_\alpha) \quad (133)$$

Invoking First-order kinetics approximation of rate dynamics, we can rewrite this conditional transition probability:

$$p(\rightarrow j_\alpha | i_\alpha \rightarrow) \equiv -\frac{k_{ji}}{k_{ii}} \quad (134)$$

On another hand, considering that the Lifetime(or Survival) probability is given by:

$$p(\tau_\alpha | i) \equiv -k_{ii} \exp(k_{ii} \tau_\alpha) \quad (135)$$

Where  $k_{ii} \equiv -\sum_{z \neq i} k_{iz}$  We can now write down the Likelihood in terms of Rates  $k_{ij}$ :

$$L = \prod_{\alpha} k_{j_\alpha i_\alpha} \exp(k_{i_\alpha i_\alpha} \tau_\alpha) \quad (136)$$

Where the  $k_{ii}$  terms cancelled out, and we can notice that the expression can be split in two products, one concerning the transitions and the other the lifetimes.

Note that  $i_\alpha$  and  $j_\alpha$  are the 'starting state' and 'destination state' of transition  $\alpha$  respectively.

The first part can be rewritten as a product over  $i$  and  $j$  ( $\neq$ ), however the second one requires more attention.

For the lifetimes part, we temporarily define  $T(z)$  as the set of all observed survival times  $\tau$  for a given initial state  $z$ , and combine the product by initial state  $z$ :

$$L = \left[ \prod_i \prod_{j \neq i} k_{ji}^{n_{ji}} \right] \left[ \prod_z \prod_{\tau \in T(z)} \exp(k_{zz}\tau) \right] \quad (137)$$

Where  $n_{ij} \equiv \sum_\alpha \delta(i_\alpha - i)\delta(j_\alpha - j)$

Bringing the products of the second term into the exponential transforms it into a sum:

$$L = \left[ \prod_i \prod_{j \neq i} k_{ji}^{n_{ji}} \right] \left[ \exp\left(\sum_z k_{zz} \sum_{\tau \in T(z)} \tau\right) \right] \quad (138)$$

Where now the  $\tau$  can be summed over:

$$t_z \equiv \sum_{\tau \in T(z)} \tau \quad (139)$$

To yield the final and simplest form of this likelihood function:

$$L = \left[ \prod_i \prod_{j \neq i} k_{ji}^{n_{ji}} \right] \left[ \exp\left(\sum_z k_{zz} t_z\right) \right] \quad (140)$$

Taking the natural logarithm of the Likelihood function gives:

$$\log L = \sum_i \sum_{j \neq i} n_{ji} \log(k_{ji}) + \sum_z t_z \left( -\sum_{j \neq z} k_{zj} \right) \quad (141)$$

Where we have explicited  $k_{zz} = -\sum_{j \neq z} k_{zj}$  (See Appendix (B)).

### 3.3.2.2 Solution with No constraints

Without further constraints, we can derive  $\log L$  with respect to  $k_{i''j''}$  ( $i'' \neq j''$ ):

$$\frac{\partial}{\partial k_{i''j''}} \log L = \sum_i \sum_{j \neq i} \frac{n_{ji}}{k_{ji}} \frac{\partial}{\partial k_{j''i''}} (k_{ji}) - \sum_z t_z \sum_{j \neq z} \frac{\partial}{\partial k_{j''i''}} (k_{zj}) \quad (142)$$

Which simplifies to:

$$\frac{\partial}{\partial k_{j''i''}} \log L = \frac{n_{j''i''}}{k_{j''i''}} - t_{i''} \quad (143)$$

Which equated to zero yields

$$k_{j''i''} = \frac{n_{j''i''}}{t_{i''}} \quad (144)$$

### 3.3.2.3 Constraint for Detailed Balance

The constraint for Detailed Balance defined as  $C_\lambda$

$$C_\lambda = \sum_i \sum_{j \neq i} \lambda_{ji} (\pi_i k_{ji} - \pi_j k_{ij}) \quad (145)$$

Have the following derivative:

$$\frac{\partial}{\partial k_{i''j''}} C_\lambda = \pi_{i''} (\lambda_{j''i''} - \lambda_{i''j''}) \quad (146)$$

Consider now maximizing the Lagrangian:

$$Lagr = \log L - C_\lambda \quad (147)$$

A key step is to solve for  $k_{ij}$  and  $k_{ji}$  simultaneously:

$$\begin{cases} \frac{\partial \text{Lagr}}{\partial k_{j''i''}} = 0 \\ \frac{\partial \text{Lagr}}{\partial k_{i''j''}} = 0 \end{cases} \Rightarrow \frac{\partial \text{Lagr}}{\partial k_{j''i''}} + \frac{\partial \text{Lagr}}{\partial k_{i''j''}} = 0 \quad (148)$$

This relation can be written explicitly in terms of the rates:

$$\pi_{j''} k_{i''j''} [n_{j''i''} - t_{i''} k_{j''i''}] = -\pi_{i''} k_{j''i''} [n_{i''j''} - k_{i''j''} t_{j''}] \quad (149)$$

Using the condition for detailed balance, we can simplify further:

$$[n_{j''i''} - t_{i''} k_{j''i''}] = -[n_{i''j''} - k_{i''j''} t_{j''}] \quad (150)$$

Which by rearranging finally yields:

$$\begin{aligned} k_{j''i''}^{(*)} &= \frac{n_{j''i''} + n_{i''j''}}{t_{i''}} - k_{i''j''} \frac{t_{j''}}{t_{i''}} \\ &= \frac{n_{j''i''}}{t_{i''}} + \frac{n_{i''j''} - t_{j''} k_{i''j''}}{t_{i''}} \end{aligned} \quad (151)$$

Where on the right hand side, we recover the solution of the problem without Detailed Balance constraint with the first term, and have the contribution of Detailed Balance in the second term.

The optimal rate  $k_{j''i''}$  is dependent on the 'opposite rate'  $k_{i''j''}$ . We can use Detailed Balance again to get out  $k_{ij}$ , at the cost of re-introducing  $\frac{\pi_i}{\pi_j}$ :

$$k_{j''i''}^{(2)} = \frac{(n_{j''i''} + n_{i''j''})}{t_{i''}} \frac{1}{[1 + \frac{t_{j''}}{t_{i''}} \frac{\pi_i}{\pi_j}]} \quad (152)$$

$$k_{i''j''}^{(2)} = \frac{(n_{j''i''} + n_{i''j''})}{t_{i''} + t_{j''} \frac{\pi_i}{\pi_j}} \quad (153)$$

Where we now have the  $\pi_i$ s listed explicitly, but the equation is not in a closed form if we know  $\frac{\pi_i}{\pi_j}$ .

This result is consistent with the fact that when the sampling is perfectly converged in terms of  $t_i$ , we recover

$$k_{j''i''} = \frac{\frac{(n_{j''i''} + n_{i''j''})}{2}}{t_{i''}} \quad (154)$$

We will see in a later Section that it is possible to first estimate the Rates without the Detailed Balance constraint, and only then use the obtained Equilibrium distribution  $\pi^{(1)}$  for the solution above. Moreover, this procedure does not perturb the equilibrium distribution and only re-scales the rates to account for Detailed Balance.

Finally, the same applies to Biased Trajectories.

### 3.3.3 Multiple Biased Trajectories

#### 3.3.3.1 Construction of the Likelihood

For the case of biased trajectories, the log-Likelihood becomes:

$$\begin{aligned} \log L &= \\ &= \sum_{\beta} \left[ \sum_i \sum_{j \neq i} n_{ji}^{\beta} \log(k_{ji} c_{ji}^{\beta}) - \sum_z (\sum_{j \neq z} k_{jz} c_{jz}^{\beta}) t_z^{\beta} \right] \end{aligned} \quad (155)$$

Where  $\beta$  is the index of each separate trajectory, whether it is biased or not.

Also,  $n_{ji}^{\beta}$  and  $t_i^{\beta}$  are Observables, and  $k_{ji}^{\beta}$  are the biased rates, assuming a general form:

$$k_{ji}^{\beta} = k_{ji} c_{ji}^{\beta} \quad (156)$$

Where  $c_{ji}^{\beta} = 1$  in the absence of bias in the simulation  $\beta$ .

### 3.3.3.2 Solution without the DB constraint

Deriving the logLikelihood with respect to  $k_{j''i''}$ :

$$\frac{\partial}{\partial k_{j''i''}} \log L = \sum_{\beta} \left[ \frac{n_{ji}^{\beta}}{k_{ji}} - t_{i''}^{\beta} c_{j''i''}^{\beta} \right] \quad (157)$$

Which can be set to zero to yield:

$$k_{j''i''} = \frac{\sum_{\beta} n_{ji}^{\beta}}{\sum_{\beta} t_{i''}^{\beta} c_{j''i''}^{\beta}} \quad (158)$$

### 3.3.3.3 Solution with the Detailed Balance constraint

The constraint for Detailed Balance only deals with the un-biased rates and is therefore unchanged by the multiplicity of trajectories or the presence of a bias, and is therefore defined as  $C_{\lambda}$  in the previous section

Now the derivative of the Lagrangian becomes:

$$\frac{\partial}{\partial k_{j''i''}} \text{Lagr} = \sum_{\beta} \left[ \frac{n_{ji}^{\beta}}{k_{ji}} - t_{i''}^{\beta} c_{j''i''}^{\beta} \right] - \pi_{i''} (\lambda_{j''i''} - \lambda_{i''j''}) \quad (159)$$

As in the the non-biased cased, considering the system of equations:

$$\begin{cases} \frac{\partial}{\partial k_{j''i''}} \text{Lagr} \equiv 0 \\ \frac{\partial}{\partial k_{i''j''}} \text{Lagr} \equiv 0 \\ \frac{\partial}{\partial \lambda_{j''i''}} \text{Lagr} \equiv 0 \end{cases} \quad (160)$$

Gives

$$\begin{cases} \frac{1}{\pi_{i''}} \sum_{\beta} \left[ \frac{n_{ji}^{\beta}}{k_{ji}} - t_{i''}^{\beta} c_{j''i''}^{\beta} \right] - (\lambda_{j''i''} - \lambda_{i''j''}) = 0 \\ \frac{1}{\pi_{j''}} \sum_{\beta} \left[ \frac{n_{ij}^{\beta}}{k_{ij}} - t_{j''}^{\beta} c_{i''j''}^{\beta} \right] - (\lambda_{i''j''} - \lambda_{j''i''}) = 0 \\ \pi_{i''} k_{j''i''} = \pi_{j''} k_{i''j''} \end{cases} \quad (161)$$

Summing and rearranging as in the previous section yields:

$$\sum_{\beta} [n_{ji}^{\beta} - k_{ji} t_{i''}^{\beta} c_{j''i''}^{\beta}] = - \sum_{\beta} [n_{ij}^{\beta} - k_{ij} t_{j''}^{\beta} c_{i''j''}^{\beta}] \quad (162)$$

Now, lets use the following shortcuts:

$$\begin{cases} N_{ji} \equiv \sum_{\beta} n_{ji}^{\beta} \\ T_{ji} \equiv \sum_{\beta} t_{i''}^{\beta} c_{j''i''}^{\beta} \\ T_{ij} \equiv \sum_{\beta} t_{j''}^{\beta} c_{i''j''}^{\beta} \end{cases} \quad (163)$$

The equation above can be rewritten as:

$$(N_{ji} + N_{ij}) = k_{ji} T_{ji} + k_{ij} T_{ij} \quad (164)$$

Where the relation between the optimal  $k_{ji}$  and  $k_{ij}$  satisfying DB is explicated:

$$k_{ji} = \frac{(N_{ji} + N_{ij})}{T_{ji}} - k_{ij} \frac{T_{ij}}{T_{ji}} = \frac{N_{ji}}{T_{ji}} + \frac{N_{ij} - k_{ij} T_{ij}}{T_{ji}} \quad (165)$$

As in the non-biased case, we can use the Detailed Balance relation once more to replace  $k_{ij}$ , which finally gives an equivalent of DHAMed:

$$k_{ji} = \frac{(N_{ji} + N_{ij})}{\frac{\pi_i}{\pi_j} T_{ij} + T_{ji}} \Leftrightarrow \pi_i k_{ji} = \frac{(N_{ji} + N_{ij})}{\frac{T_{ij}}{\pi_j} + \frac{T_{ji}}{\pi_i}} \quad (166)$$

### 3.3.3.4 Discussion about $k^{(1)}$ and $k^{(2)}$

Considering  $k_{ji}^{(1)}$  as the maximum-Likelihood Rates without DB, and  $k_{ji}^{(2)}$  the max-Likelihood with the DB constraint.

With the following inter-changeability between the terms involved in the solutions of Biased and non-Biased trajectories respectively:

$$\begin{cases} N_{ji} \sim n_{ij} \\ T_{ji} \sim t_i \\ T_{ij} \sim t_j \end{cases} \quad (167)$$

The solutions derived in the present manuscript can be expressed in the following compact form:

$$\begin{cases} k_{ji}^{(1)} = \frac{N_{ji}}{T_{ji}} \\ k_{ji}^{(2)} = \frac{N_{ji} + N_{ij}}{T_{ji}} - \frac{T_{ij}}{T_{ji}} k_{ij}^{(2)} = k_{ji}^{(1)} + \frac{N_{ij} - T_{ji} k_{ij}^{(2)}}{T_{ji}} \end{cases} \quad \text{and} \quad k_{ji}^{(2)+reDB} = \frac{(N_{ji} + N_{ij})}{\frac{\pi_i}{\pi_j} T_{ij} + T_{ji}} \quad (168)$$

### 3.3.4 Proof that enforcing DB in a 2nd step keeps $\pi$ unchanged

We will consider the situation where we proceed in 2 steps:

1-Estimate the Rates  $k_{ji}^{(1)}$  without enforcing Detailed Balance, and obtain the consequent  $\pi$  that we will write  $\pi^{(1)}$ .

2-Estimate the Rates  $k_{ji}^{(2)}$  constrained to satisfy Detailed Balance, using  $\pi^{(1)}$  and the same trajectory data.

Now, we write  $\pi^{(2)}$  the Equilibrium Probability following  $k^{(2)}$ , and we will show below that  $\pi^{(1)} = \pi^{(2)}$  for both Biased and non-Biased trajectories.

#### 3.3.4.1 Non-Biased

Since  $k^{(2)}$  satisfies DB, we have:

$$\frac{\pi_i^{(2)}}{\pi_j^{(2)}} = \frac{k_{ij}^{(2)}}{k_{ji}^{(2)}} \quad (169)$$

Where from Eq. (168) we have

$$k_{ji}^{(2)} = (n_{ji} + n_{ij}) \frac{\pi_j^{(1)}}{t_j \pi_i^{(1)} + t_i \pi_j^{(1)}} \quad (170)$$

Replacing  $k_{ij}$  and  $k_{ji}$  in the DB equation allows to cancel most terms and yields:

$$\frac{\pi_i^{(2)}}{\pi_j^{(2)}} = \frac{k_{ij}^{(2)}}{k_{ji}^{(2)}} = \frac{\pi_i^{(1)}}{\pi_j^{(1)}} \quad (171)$$

Since  $\pi^{(1)}$  and  $\pi^{(2)}$  are normalized, it follows that they are equal.

#### 3.3.4.2 Biased

Similarly to the proof for the non-Biased case, we have the DB equation and the formula for  $k_{ij}^{(2)}$ :

$$k_{ji}^{(2)} = (N_{ji} + N_{ij}) \frac{\pi_j^{(1)}}{\pi_i^{(1)} T_{ij} + T_{ji} \pi_j^{(1)}} \quad (172)$$

Where we have rewritten some terms for simplicity:

$$\begin{cases} N_{ji} \equiv \sum_{\beta} n_{ij}^{\beta} \\ T_{ji} \equiv \sum_{\beta} t_{i^{\beta}}^{\beta} c_{j^{\beta}}^{\beta} \\ T_{ij} \equiv \sum_{\beta} t_{j^{\beta}}^{\beta} c_{i^{\beta}}^{\beta} \end{cases} \quad (173)$$

Taking the ratio  $\frac{k_{ij}^{(2)}}{k_{ji}^{(2)}}$  again simplifies everything and gives:

$$\frac{\pi_i^{(2)}}{\pi_j^{(2)}} = \frac{k_{ij}^{(2)}}{k_{ji}^{(2)}} = \frac{\pi_i^{(1)}}{\pi_j^{(1)}} \quad (174)$$

Again, since they are also both normalized they are equal.

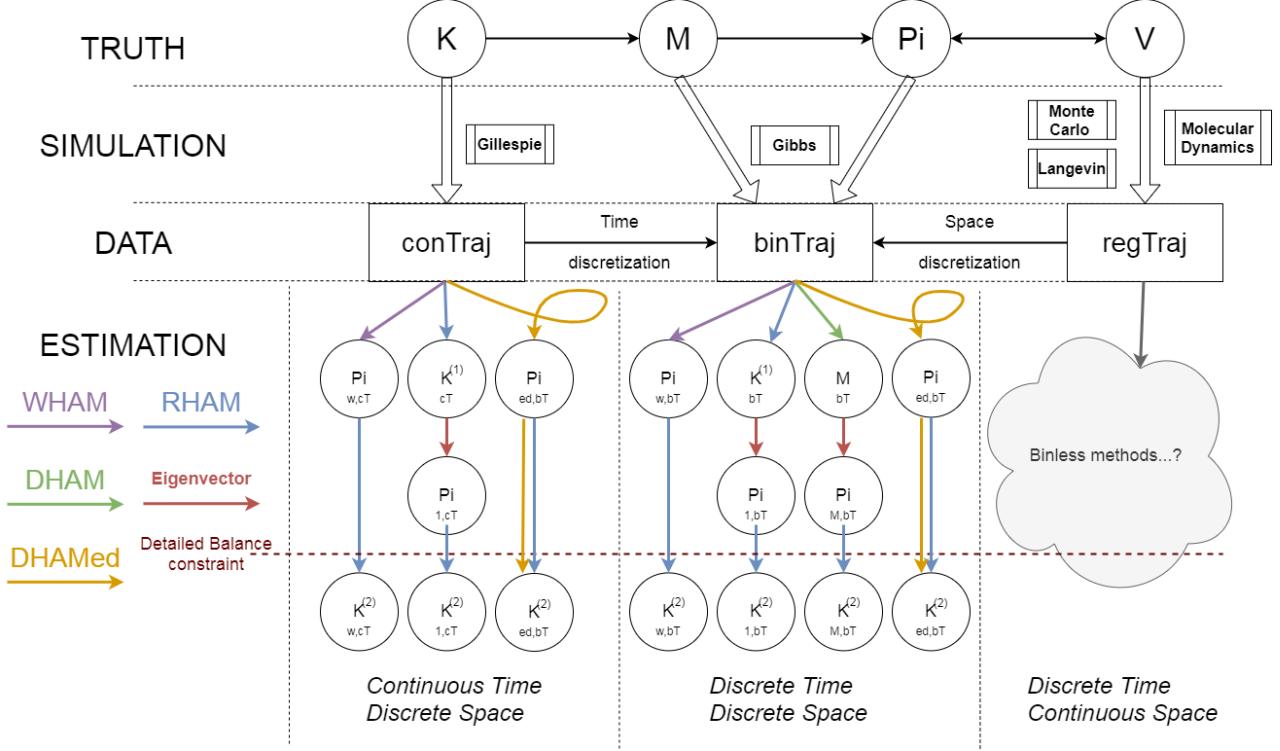


Figure 2: Flowchart of the different ways to simulate a dynamical system and estimate its transition rates matrix  $\mathbf{K}$ .

We are now certain that  $\pi$  remains unchanged by the procedure, for both non-Biased and Biased systems.

### 3.3.4.3 Arbitrary probability distribution

In fact, the same way that we have shown the result above, we can show that we can force the Rate matrix "K2" to satisfy any arbitrary equilibrium distribution.

Consider the solution K2 with an arbitrary distribution  $\pi_i^{(*)}$ :

$$k_{ji}^{(2)} = \frac{(n_{ji} + n_{ij})}{t_{i''}} \frac{1}{[1 + \frac{t_j}{t_i} \frac{\pi_j^{(*)}}{\pi_i^{(*)}}]} \quad (175)$$

K2 is constructed in such a way to always satisfy detailed balance and have the equilibrium distribution that we input into it:

$$\frac{k_{ji}^{(2)}}{k_{ij}^{(2)}} = \frac{\pi_j^{(*)}}{\pi_i^{(*)}} \quad (176)$$

Hence, what K2 is conceptually is a mapping from (Data,  $\pi^*$ ) to a maximum likelihood Rate matrix that satisfies Detailed Balance, and have  $\pi^*$  as equilibrium distribution.

Conceptually, we can write:

$$\text{eigvec}_1(K2(\text{Data}, \pi)) = \pi \quad \forall \text{Data}$$

The best use for this relation is when we already know the equilibrium distribution and want to estimate the dynamics of the system with a constraint to satisfy DB.

Otherwise, we still need to properly compare  $\pi(K1)$  and  $\pi(DHAMed)$ .

In any case it is almost certain that  $\pi(K1)$  is a better starting point than a uniform distribution for the DHAMed iteration.

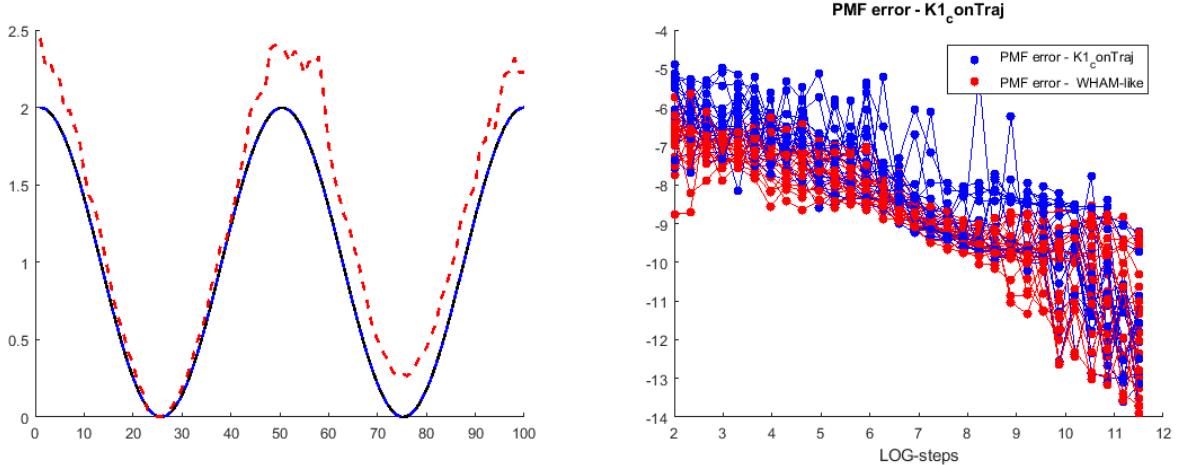


Figure 3: Left: PMF used for the test of the  $K^{(1)}$  formula (blue line) and one estimate (red dotted line). Right: robust error of PMF estimation over simulation length, for 100 replicas

### 3.3.5 Applications

Dynamical systems can be defined and described in various ways. We consider the question from a physical point of view and consider a dynamic on an energy potential  $V$ . Such a system, initially described by  $V$  can be expressed as a transition rates matrix  $\mathbf{K}$  through the Arrhenius rates, which in turn can be used to obtain the transition probabilities matrix  $\mathbf{M}$  by using  $[e^{(\tau\mathbf{K})}]$ .

We will define a few potentials on a 1-dimensional linear chain, and compute the subsequent  $\mathbf{K}$  and  $M^{(\tau)}$ . There are also multiple algorithms available to generate a trajectory of the given system: Gillespie, Gibbs, Monte-Carlo, Langevin, Molecular dynamics.

One shall be conscious of the fundamental differences between the algorithms, in particular that the Gillespie algorithms outputs a time-continuous trajectory, called "conTraj" in the following, while the Gibbs simulations outputs a trajectory discrete in both time and space called "binTraj" in the following. On the other hand, Monte-Carlo, Langevin and Molecular dynamics alrogithms outputs a space-continuous trajectory, that requires to be discretized (See Fig. (2)).

Most of the kinetics estimation algorithms rely on some sort of space discretization, and are built and used for trajectories in discrete space, which we will call "binTraj". While some of them can be used from continuous trajectories ("conTraj"), it is unreasonable to expect having proper time-continuous trajectories in a real-life setting .

For this reason we consider first the estimates from "conTraj" using the Gillespie algorithm. This will be useful to then compare the results obtained when applying the exact same methods, but using the "binned trajectory". It is possible because we are able to perform the time-discretization directly, indeed simulation the act of measuring the state of the system at a fixed time interval, that we will call "measure-lagtime".

#### 3.3.5.1 Gillespie simulations

The Gillespie algorithm is the most computationally efficient in a sense, because it can be used to generate either multiple events in a given time interval, or directly simulate the time between two rare events. We consider the latter use of the Gillespie algorihm, to simulate transitions one by one from the "true" Rates matrix  $K^{(T)}$ .

Another simulation algorithm to try in the future would be the Langevin dynamics.

### 3.3.5.2 Error measure

Once we have estimated  $\pi, \mathbf{K}$  or  $\mathbf{M}$  from the trajectory, what are the robust, reliable, and meaningful ways to quantify the deviation of our estimate from the ground truth?

The simplest and universal approach would be to compare the resulting stationary distribution  $\pi$ . At first glance, two approaches are possible: RMSD or KL-divergence. On closer inspection, when the sampling is too short or the state partitioning too fine, and in particular at the beginning of a simulation, one often encounters situations where only a small fraction of the phase space to estimate has been explored yet.

In this case, the stationary distribution may be either undefined, or at best filled with zeros at the entries corresponding to the non-explored bins. In such case, the KL-divergence fails due to the zeros, and while the RMSD can still be defined if the sum is restricted to the entries  $\pi_i$  where bin  $i$  has been explored, it gives erratic results until all the bins have been explored.

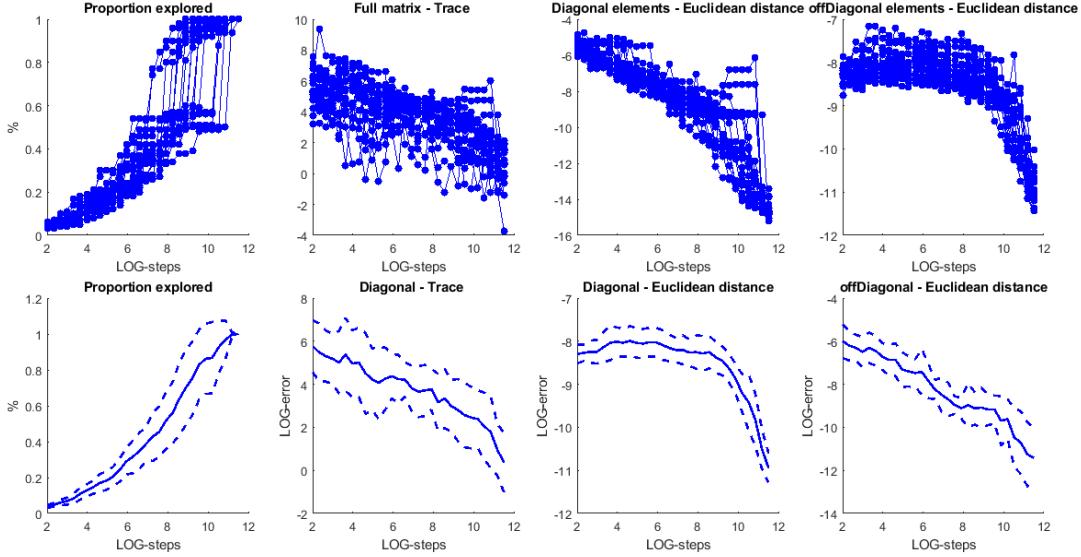


Figure 4: Various measures of the deviation of the estimated rates matrix from the ground truth.

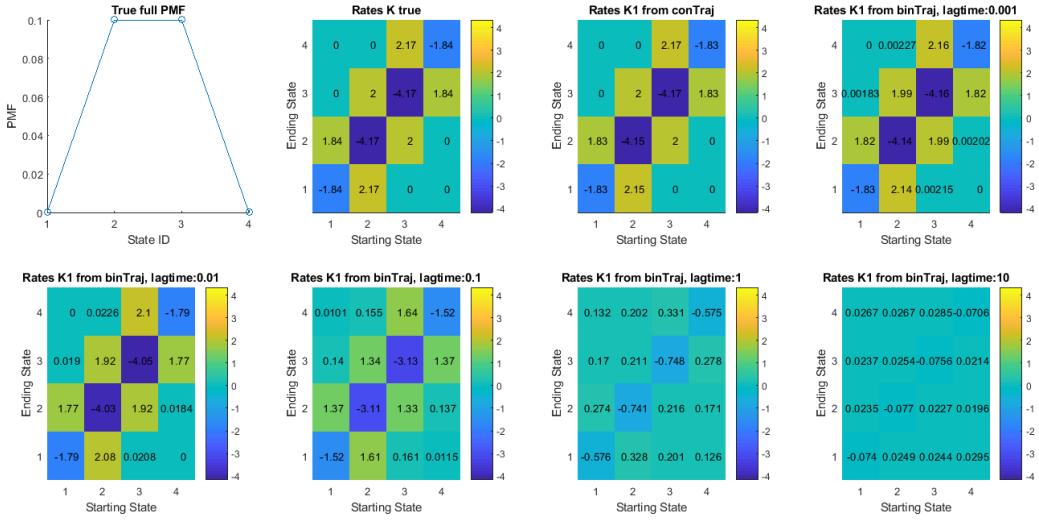


Figure 5: Top-left: PMF used for the test on 4 states, and the ground truth rates matrix "Rates K true". The rest: Comparison of ground truth matrix and the rates matrix estimated using the  $K^{(1)}$  formula at different lag-times.

### 3.3.5.3 Robust error measures of the vector PMF

A robust error for the PMF,  $E_{PMF}$ , can be defined the following way:

$$E_{PMF} = \frac{1}{N} \left[ \sum_{i_F^{n_F}} (x_i - y_i)^2 + \alpha_{PMF}(N - n_F) \right] \quad (177)$$

The parameter  $\alpha_{PMF}$  reflects the trade-off between the reward for exploration and the reward for precision. We have found that using  $\alpha_{PMF} \equiv \frac{1}{N^2}$  was yielding a relatively smooth description of the convergence, therefore providing a reasonable mixing parameter as shown in Fig. (3).

### 3.3.5.4 Robust error measures of the matrix K

In addition to the exploration measure,  $N_{sampled}/N_{total}$ , the estimated rates matrices K can be decomposed into 3 meaningful parts:

- 1-The Trace of the matrix  $Tr(K)$
- 2-The diagonal element's absolute value ( $k_{ii}$ )
- 3-The off-diagonal element's relative value ( $k_{ji}/k_{ii}$ )

This way we can define three separate error measures, and compare the estimates of these quantities to the ground truth, as shown in Fig. (4).

These error measure will be used to quantify the error of estimated transition rates and/or probability matrices in future benchmarking work.

## 3.3.6 Conclusion

We have derived two expression for the maximum-likelihood estimate of the transition rates matrix from biased simulations, with and without the constraint of Detailed Balance.

We have seen that the estimate with Detailed Balance can be used to force an arbitrary equilibrium distribution, which might be known beforehand, and obtain the maximum-likelihood transition rates satisfying Detailed Balance and having this exact stationary distribution.

We tested our estimation formula on trajectories generated using the Gillespie algorithm, to obtain continuous-time trajectories, and assessed the estimate are different lag-times.

As expected, the rates matrix estimate converges to the ground truth on continuous-time trajectories, but diverges from it as lag-time increase. For small lag-times, the estimation error is similar to the DHMA estimate.

### 3.3.6.1 Measure lag-time problem

The whole model assumes that our measure of the transitions and time spent in each micro-state is perfect.

We can easily see that this can lead to problems when multiple transitions happen between each measurement, because we assume that we have observed only one.

A typical example being a Linear Chain network with nodes only connected to their nearest neighbours.

If we simply take 'screenshots' of the 'frames' (current state of the system), with a constant time-interval that we call 'measure lag-time', we are likely to observe the system going from state [i] to state [j > i+1].

In such a case applying the procedure discussed above will lead to assigning non-zero rates to non-existent edges on the Network.

However, we could think of taking advantage of a prior knowledge of the structure of the network (Adjacency matrix)..

### 3.3.6.2 Leveraging on prior knowledge of the Adjacency

Considering the simple Network of a Linear Chain with Nearest neighbours adjacency, we could stop counting the observed lifetimes in multiples of our measure lag-time and divide the elapsed time by the number of visited

states:

Imagine that we measure the system in State [i] and then in State [i+2] after our measure lag-time  $\tau$ .

Knowing that there is no edge between [i] and [i+2], we can consider having observed 2 transitions: [i]->[i+1] and [i+1]->[i+2] during the lag-time  $\tau$ .

Also, we would need to split the elapsed time  $\tau$  and assign the parts to the states involved. Here there might be multiple choices on the splitting, and this can be investigated further.

### 3.3.6.3 Prior knowledge of $k_{ii}$ /Lifetimes

In the formula for  $K^{(2)+reDB}$ , we have the  $N$  entries of the equilibrium probability vector as free parameters. If we have prior knowledge of all Lifetimes (Mean First Time to Leave)  $k_{ii}$ , and set the corresponding constraint on  $\sum_{j \neq i} k_{ji}$ , I might resolve this issue.

### 3.3.6.4 Invariance of $k_{ii}$ ?

It has been observed numerically that the following equation seems to hold up to  $10^{-6}$ :

$$k_{ii}^{(1)} = k_{ii}^{(2)} \quad (178)$$

However it seems that it is not necessarily true analytically, it might be interesting to look into.

### 3.3.7 Side note: Markov Neural Network

Usually one estimates the Markov Matrix from one or multiple trajectories at a specific lag-time  $\tau$  only.

The information given by the same trajectory when looking at the transitions at other lag-times is not used.

We will below describe an approach to estimate a Markov Matrix at lag-time  $\tau \equiv M^\tau$  from sampled transitions that happened at lag-time  $\tau, 2\tau, 3\tau, \dots, n\tau$ .

#### 3.3.7.1 Likelihood Maximization at multiple lag-times simultaneously

A common approach to estimate probabilistic models is to maximize the Likelihood of such models.

Usually to estimate a Markov Model at a given lag-time, one maximizes the likelihood using sampled transitions at this lag-time.

For this approach a Closed Form Solution is known (For non-biased trajectories), and an Approximate Closed Form solution known for biased trajectories (DHAM).

In principle we could also use the transitions at higher lag-times for the computation of the Likelihood of our Model, however there is no closed form solution known for this approach.

Below we describe both approaches to show how adding higher lag-time transitions increases the difficulty of the analytical Likelihood Maximization problem.

For the sake of simplicity again, we will consider only  $\tau$  and  $2\tau$  in the following, but the extension to further lag-times is straightforward.

$$L = \prod_{i,j} (M^{(\tau)})_{ij}^{T_{ij}^{(\tau)}} * (M^{(2\tau)})_{ij}^{T_{ij}^{(2\tau)}} \quad (179)$$

Where  $T_{ij}^{(\tau)}$  and  $T_{ij}^{(2\tau)}$  are the transition counts from  $i$  to  $j$  when processing the trajectory at lag-time = 1 and 2 respectively.

$$\log L = \sum_{i,j} \log((M^{(\tau)})_{ij}) T_{ij}^{(\tau)} + \log((M^{(2\tau)})_{ij}) T_{ij}^{(2\tau)} \quad (180)$$

Since  $(M^2)_{ij} = \sum_k M_{ik} * M_{kj}$

$$L = \prod_{i,j} (M^{(\tau)})_{ij}^{T_{ij}^{(\tau)}} * (\sum_k M_{ik}^{(\tau)} M_{kj}^{(\tau)})^{T_{ij}^{(2\tau)}} \quad (181)$$

$$\log L = \sum_{i,j} \log((M^{(\tau)})_{ij}) T_{ij}^{(\tau)} + \log(\sum_k M_{ik}^{(\tau)} M_{kj}^{(\tau)}) T_{ij}^{(2\tau)} \quad (182)$$

With this novel approach, we have the same number of free parameters to estimate ( $N(N-1)$ ) but more "data" as  $\{\cup_{ij}\{T_{ij}^\tau\}\} \subset \{\cup_{ij}\{T_{ij}^\tau, T_{ij}^{2\tau}\}\}$ .

However we have no analytical way to maximize this function yet.

We can overcome this unfortunate difficulty by approaching it from the perspective of Machine Learning. In particular, we will show that we can build an **interpretable** Neural Network architecture designed to represent a Markov Matrix and train from trajectories.

### 3.3.7.2 Markov Neural Network - Idea

The idea behind the Markov Neural Network comes from the fact that Deep Neural Network can in principle replicate any function.

Now, we can construct the **architecture** (topology) of the neural network specifically in such a way to make it work in ways we want it to.

Convolutional and Recurrent Neural Networks are a good example of this "Architectural constraints" approach.

In this case we will construct the architecture of the Neural Network to represent a Markov Matrix, which is why we will call it 'Markov Neural Network', and train it to maximize the likelihood by training its stochastic predictions (output) against the sampled trajectories (target), in a Supervised Training approach.

### 3.3.7.3 Markov Neural Network - The architecture

To replicate a Markov Matrix of size  $N$ , the MNN will require (only)  $N^2$  distinct weights. These  $N^2$  will constitute a 'MNN-Block' (called simply Block in the following).

Each Block takes a  $N$ -dimensional vector as input and outputs a  $N$ -dimensional vector. This way it is supposed to propagate the probabilities.

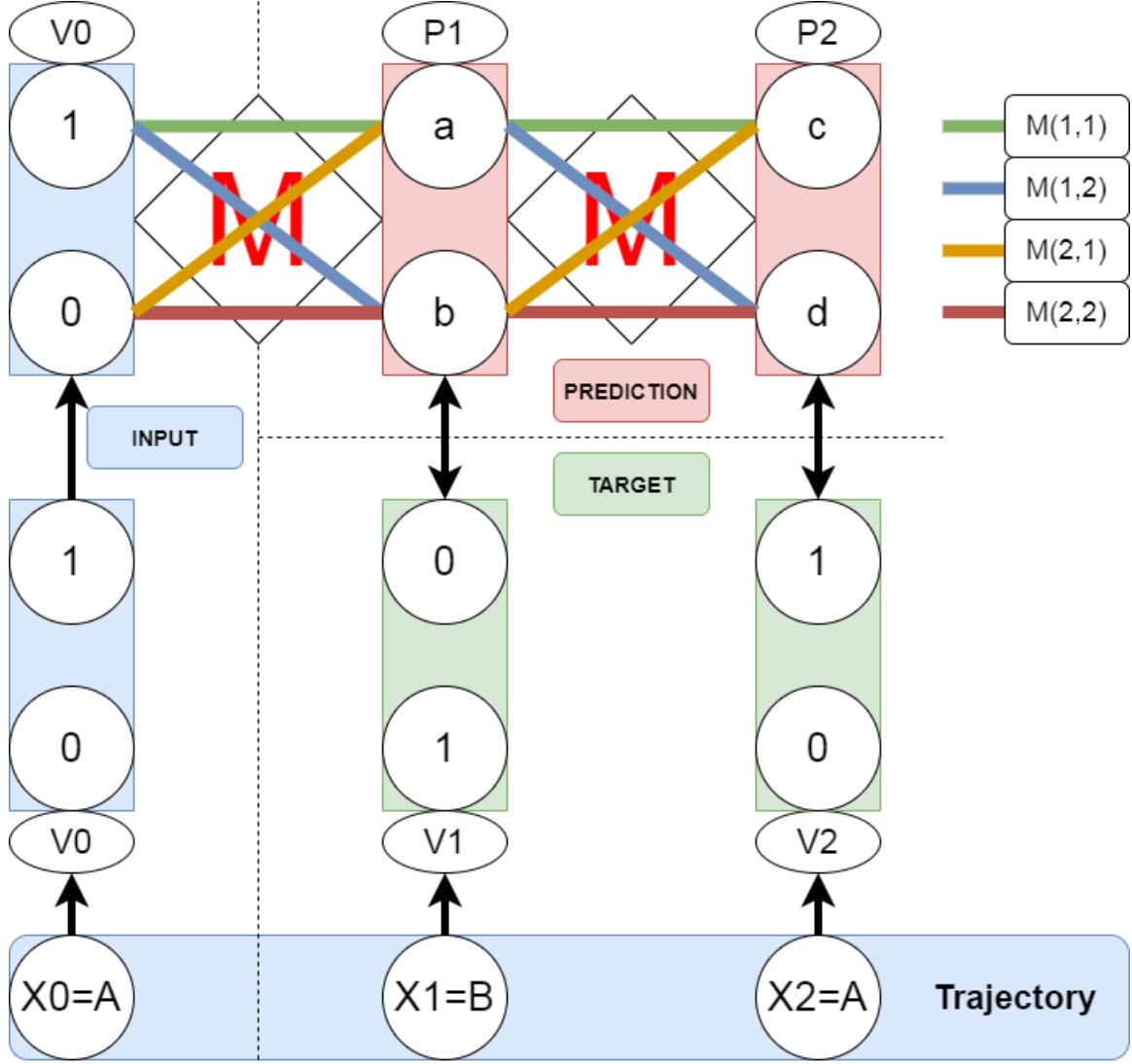


Figure 6: The Architecture of the network, for  $N=2$  and  $l = 2$ . On the top, circles represent nodes of the network, each block of weights is equal to the next as shown by the four colours of the edges.

The aim is to apply the same Block of weights, representing the matrix  $\mathbf{M}$  and containing the trained transition probabilities, iteratively to produce a series of predictions, and compare it to the observed trajectory for training.

There are two ways to build the output: as probabilities ( $p(x, t|y, t - 1)$ ) or as position prediction (1 state chosen and assigned following  $p(x, t|y, t - 1)$ ). We consider the former approach in the explanation, but both should be considered and compared in terms of results in future work.

### 3.3.7.4 Training Procedure as Likelihood maximization

For one input sub-trajectory, we have  $l$  model output vectors  $\{P_i\} \quad \forall i \in [[1; l]]$ , and  $l$  target vectors  $\{V_i\} \quad \forall i \in [[1; l]]$ , corresponding to one-hot vectors encoding the observed position.

To compute the Likelihood defined by eq (4) we simply have to multiply element-wise the vector  $P_i$  and  $V_i$  for each  $i$ , and then multiply all the elements in this vector to obtain the Likelihood.

Then we can simply take  $-L$  as the Error function to minimize and turn on Back-Propagation, provided we have a good way to enforce the constraints on the weights of the network.

Since we want the weights  $\mathbf{W}$  to represent transition probabilities, we need them to satisfy some regularity constraints, i.e:

$$\begin{cases} W_{ij} > 0 & \forall (i, j) \\ \sum_j W_{ij} = 1 & \forall i \end{cases} \quad (183)$$

Here  $W_{ij}$  is the value of the Weight that we intend to represent  $M_{ij}$  with.

One approach that might seem straightforward in its application but not in its rigour, which we will call the 'Brute method', would be to bound the values of  $W_{ij}$  to  $[0, +\infty]$  and then re-normalize the weights after each Back-Propagation. However it is not certain that this method will behave well, as the initial update of the Back-Propagation will not be 'aware' of the "bounding and re-normalization" step and hence the training might be unstable.

Because of those concerns it could be interesting to us a more elegant approach.

### 3.3.7.5 Elegant constraint reduction

In this approach we consider the constraints:

$$\begin{cases} W_{ij} > 0 & \forall(i, j) \\ \sum_j W_{ij} = 1 & \forall i \end{cases} \quad (184)$$

Defining a matrix  $\mathbf{Q}$  as the "square root" of  $\mathbf{W}$

$$(Q_{ij})^2 \equiv W_{ij} \quad (185)$$

We can re-express the same problem in terms of  $Q_{ij}$ , and optimize the Likelihood (learn the weights) for  $Q_{ij}$ .

In practice, it corresponds to placing two consecutive layers with the exact same weight and linear activation functions, as shown in Fig. (58).

Fortunately,  $W_{ij} > 0 \quad \forall(i, j)$  is **always** satisfied via the form of the change of variables, and now the constraints are reduced.

### 3.3.7.6 Reduced Constraints

The new and only constraint now on the weights is:

$$\sum_j (Q_{ij})^2 = 1 \quad \forall i \quad (186)$$

This new constraint strictly corresponds to constraining the vectors  $Q_{i:} \equiv Q_i$  to be positioned on the surface of the hyper-sphere of dimension N.

### 3.3.7.7 Corresponding Architecture

We can construct a corresponding NN Architecture that will do what we want:

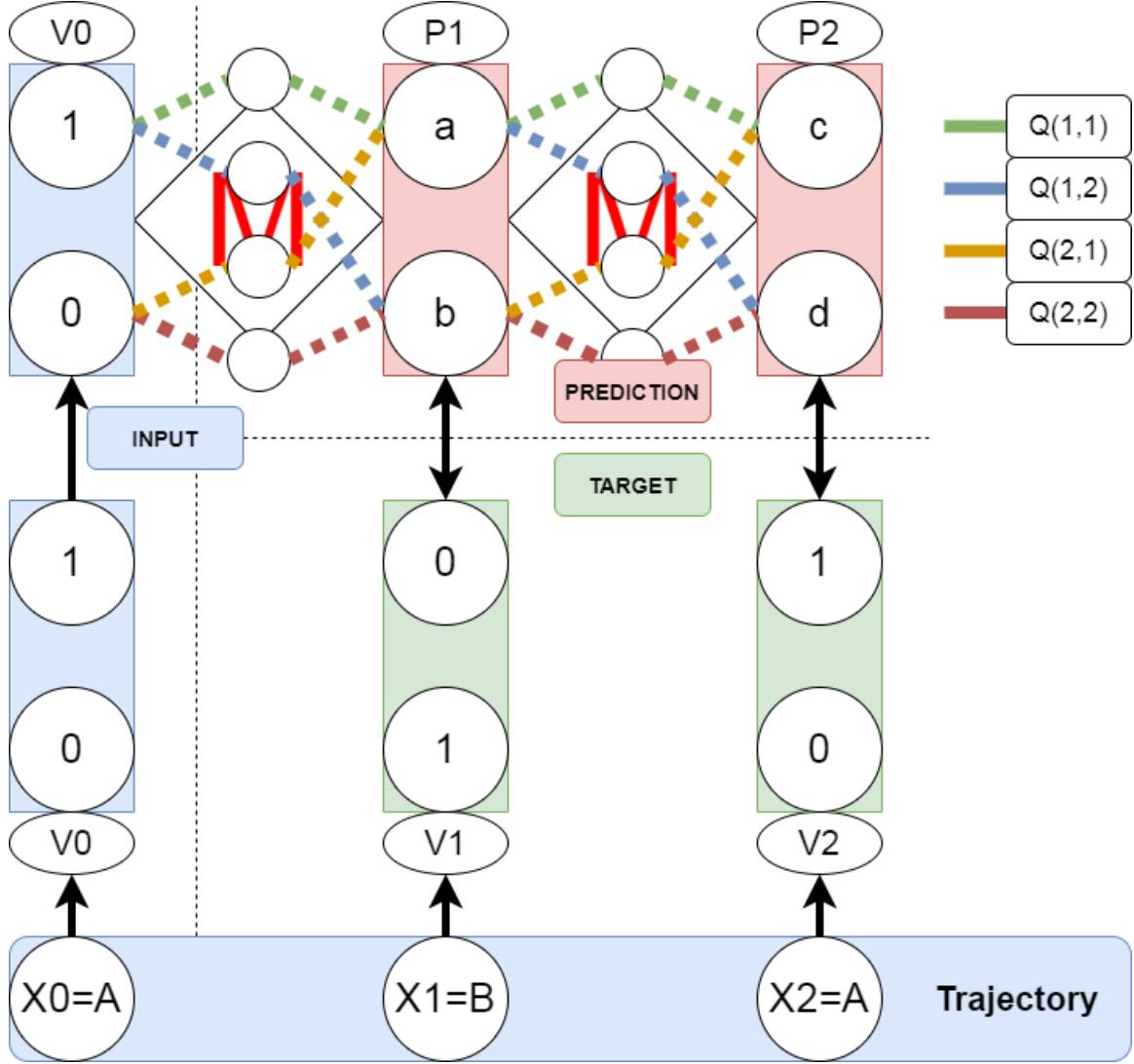


Figure 7: The Architecture of the Q-network, for  $N=2$  and  $l = 2$

Note: Linear Activation functions **everywhere**, since normalization of the weights and input will ensure normalization of the Block output.

### 3.3.7.8 Enforcing new constraints

To enforce the new constraint

$$\sum_j (Q_{ij})^2 = 1 \quad \forall i \quad (187)$$

We can try to inspire from the field of spherical spin glass models, where the velocity of a spin [i] is given by.

$$\dot{x}_i = \sum_{k=1}^N J_{ik}x_k + h_i + \xi_i(t) - \mu(t)x_i \quad (188)$$

In such models the sum of the squares of the spin values is constrained to a constant,

$$\sum_{i=1}^N x_i^2 = N \quad (189)$$

Just like in our case

$$\sum_{j=1}^N Q_{ij}^2 = N \quad (190)$$

There is a clear parallel between the Euler integration scheme and Back-Propagation.

Euler integration scheme:

$$x(t+1) = x(t) + \tau \frac{\partial x(t)}{\partial t} \quad (191)$$

Basic Back-Propagation:

$$\Theta^{new} = \Theta^{old} - \lambda \frac{\partial E(\Theta)}{\partial \Theta} \quad (192)$$

from the perspective of Euler integration scheme,  $[-\frac{\partial E(\Theta)}{\partial \Theta}]$  acts as the velocity of the parameters and  $\lambda$  acts as the time-step, and

from the perspective of Back-Propagation,  $[-\frac{\partial x(t)}{\partial t}]$  acts as the Error Gradient, and  $\tau$  as the Learning Rate.

Now, inspiring from [spherical spin-glass field constraint to sphere], we take the derivative of Eq. (187) with respect to  $Q_{ij}$ :

$$\sum_{j=1}^N Q_{ij}^2 = N \iff \sum_{j=1}^N \frac{\partial(Q_{ij}(\Theta))^2}{\partial \Theta} = 0 \quad (193)$$

To obtain:

$$\sum_{j=1}^N Q_{ij}(\Theta) \frac{\partial Q_{ij}(\Theta)}{\partial \Theta} = 0 \quad \forall i \in [[1; N]] \quad (194)$$

This means that the vector  $\frac{\partial}{\partial \Theta} Q_{i:}(\Theta)$  shall be orthogonal to the vector  $Q_{i:}$ .

Now, we could use this to project the update to the space orthogonal to the one spanned by the current weights. However because of the fact that the Euler integration formula is only exact in the low lag-time/learning rate limit, we would end up with deviations from the surface of the sphere.

A solution to be certain to remain on the surface on the sphere is redefining the system in terms of spherical/polar coordinates, where updating only the angles will ensure to remain on the surface.

### 3.4 RESULT 3 (Unpublished) BIASING - MDP-like Biasing

We propose a novel approach to the enhanced sampling (importance sampling) problem, inspired from Markov Decision Processes, allowing to improve the sampling of high-dimensional phase space further than more traditional methods. Previously proposed biasing protocols have been shown to increase the sampling efficiency of Molecular Dynamics, while relying mainly on adding a constant biasing potential to the underlying Free Energy potential, but the constance of the bias limits the range of possibilities. Inspiring from the growing field of Machine Learning, and in particular Markov Decision Processes, we developed a biasing method that applies a different biasing potential to the system depending on its current position in the phase space. We apply this biasing protocol to a variety of networks with dynamics defined via the Lagrangian, and an arbitrarily defined underlying energy potential. We provide an analytical proof that although the Global Balance condition is violated, using transition-based methods allow us to successfully unbias the resulting dynamics and reconstruct the right underlying potential.

#### 3.4.1 Introduction

A Markov Decision Process (MDP) provide a mathematical framework to model a decision-making process in a stochastic environment. It is defined by 4 key elements [55–57]:

- the state space ( $S$ ), on which the system is evolving
- the set of actions (or Action space  $A$ ) that can be taken at each position in the state space - the result probabilities  $P_a(s, s')$ , quantifying the probability of the next state  $s'$  conditional on the current state  $s$  and the action taken  $a$
- the reward ( $R_a(s, s')$ ) obtained for the transition from state  $s$  to  $s'$  with the action  $a$ .

The objective of a Markov Decision Process is to find the optimal policy, i.e. a function  $a*(s)$  that provides the optimal action to perform given the current state  $s$ .

The main difficulty lies in the proper definition of the reward function  $R$  for the problem at hand. In the case of enhanced sampling, the goal is often to reduce the variance of the estimator of interest, usually the Potential of Mean Force (PMF). Also, in the simple case of standard MDP, the transition probabilities  $P_a(s, s')$  are known beforehand, which is not true for Molecular Dynamics. When the transition probabilities are unknown, the problem becomes one of Reinforcement Learning.

### 3.4.2 Heuristics

#### 3.4.2.1 Defining the Reward function

Our objective is to reduce the variance of estimator of interest, which are the transition probabilities.

For a given starting state, under the assumption of Markovianity, the distribution of transition probability from a given state  $i$  follows a multinomial distribution.

The variance of the multinomial distribution have a highly non-trivial expression, but can be approximated by:  $n\pi_j(1 - \pi_j)$ . For a DHAM estimate it translates to

$$\begin{cases} Var(T_{ji}) = M_{ji}(1 - M_{ji})n_i^2 \\ Var(T_{ji}/n_i) = \frac{M_{ji}(1 - M_{ji})}{n_i} \end{cases} \quad (195)$$

Where  $n_i$  is the number of sampled transitions starting from  $i$ , and  $M_{ji}$  is the biased probability to transition from  $i$  to  $j$ .

Hence in order to minimize the variance of all our estimates, we shall maximize the 'mobility' of the system, i.e. minimize the probability of remaining in the same state, and maximize the uniformity of the transition probabilities.

We can consider the following requirements:

- (A) - Uniformly distributed *position* counts.
- (B) - Uniformly distributed of conditional *transition* counts.
- (C) - Minimized *self-transition* counts.

Example of loss function:

$$L(b) = \alpha_A \sqrt{\sum_j (n_j - \frac{N}{m})^2} + \alpha_B \sum_i \sqrt{\sum_j (\frac{T_{ij}}{n_i} - \frac{1}{m})^2} + \alpha_C \frac{T_{ii}}{n_i}$$

Where  $n_j$  is the number of positions counts in bin  $j$ ,  $N$  is the total number of position counts,  $m$  the number of bins,  $T_{ij}$  the count of transition from  $i$  to  $j$ , and  $\alpha_A$  is a weighting parameter for requirement (A).

#### 3.4.2.2 The MDP action as selection of a bias function

Here we want to consider making the bias dependent on the *current state*, which is the current position of the system but also the past trajectory or some observable of it.

Biases are traditionally already defined once over all space and hence the force applied to the system already depend on the starting point/current position for each transition.

But the *shape* of the bias is the same whatever the position of the system.

So that our idea is to make the *shape* of the bias different for each space-state, and each such bias will be adapted using the past trajectory.

If we place ourselves in the free energy landscape (approach), we can imagine the bias as modifying this energy landscape. Then usually once this landscape have been biased, it remains the same during the simulation, and the simulated system moves freely on this biased landscape. The system movement and behaviour is determined by the shape of the landscape around it, so in some sense we can say that the system "sees" the landscape around it, and moves as a function of what it sees. In this image, traditional biasing makes the real landscape change and what the system sees at step  $t$  is coherent with what he have seen at step  $t - 1$ . However in the biasing protocol we investigate, the system can be seen as moving around the same states as in traditional simulations, but the landscape he sees is always changing, and not necessarily coherent with what the system "saw" at the previous step.

Also, our biasing method seems to correspond to the Lifting biasing method [58, 59] where we have, ironically, the lifting probability is of 1 when the system moves out of its current position.

Finally what happens is that we have a trajectory with multiple biases, we can easily compute the DHAM estimate.

### 3.4.3 Independence of estimated lines in our MSM with DHAM

Taking a close look at the DHAM method for estimating the unbiased markov matrix:

$$M_{i'j'} = \frac{\sum_k T_{i'j'}^k}{\sum_k \sum_j T_{i'j}^k b_{i'j'}^k} \quad (196)$$

we can see that each unbiased probability  $M_{i'j'}$  is obtained using the counts of transitions starting in a given position, i.e.  $M_{i'j} \quad \forall j \in [1, N]$

Equally, the unbiasing of  $M_{i'j'}$  does not depend on  $b_{yj}^k \quad \forall y \neq i'$ . Hence,  $M_{ij}$  and  $M_{yz}$  are computed totally independently, if  $i \neq y \quad \forall j, z$ , and their computations can indeed be parallelized.

#### 3.4.3.1 Position-dependent bias

If the bias depends on the position, then  $k = i$ :

$$M_{i'j'} = \frac{\sum_i T_{i'j'}^{(i)}}{\sum_i \sum_j T_{i'j}^{(i)} b_{i'j'}^{(i)}} \quad (197)$$

Since  $T_{i'j'}(i) = 0 \quad \forall i \neq i'$ , we can simplify:

$$M_{i'j'} = \frac{T_{i'j'}^{(i')}}{n_{i'}^{(i')} b_{i'j'}^{(i')}} \quad (198)$$

(Or if we have updated the bias we have multiple trajectories starting at same bin but with different bias)

From the perspective of DHAM, it looks like for each bias  $(i')$  we made a 1-step simulation,  $n_{i'}$  times. Which is actually the case when the step simulation algorithm is the direct-Markov Matrix (/special Gibbs Sampling), and is not totally true for simulation algorithms involving moving on a continuous phase space, like continuous MC, Langevin or many others.

In fact nothing is stopping us from taking the method to its extreme and indeed perform 1-step simulations, so that we can manually enforce the fact that  $n_{i'}^{(i')}$  are the same for all  $i'$

#### 3.4.3.2 Perspective on the simulated system

In a sense, given the simulation settings, although the system switches biases at each step, we can see the system as moving along a unique markov matrix  $M''$ :

Lets construct the matrix  $M''$  so that each row  $i$  corresponds to the row  $i$  of the biased matrix  $i$  (with the bias used when the starting point is  $i$ ).

$$\{M''_{ij} = M_{ij}^{(i)} \forall j\} \forall i \quad (199)$$

Each line that we import is normalized on its own, and hence we obtain a legitimate transition probability matrix.

We can clearly see that the behaviour of the simulated system indeed follows the transition matrix  $M''$  we just constructed.

This is because we use only the row  $i$  of the biased matrix  $i$  when the system is in state  $i$ , and indeed this fact can be used for the optimization of computational time.

#### 3.4.3.3 Detailed Balance Concerns

One might have concerns about the fact that the simulation do not satisfy Detailed Balance, or even Global Balance (or Semi-Detailed Balance).

First, it is important to set the distinction between the *simulated system* mentionned previously and referring to the system responsible for a 'complete' trajectory (between two bias updates, if updates there are), and the *simulating systems*, i.e. sub-systems that correspond to each bias and corresponding biased potential.

Let's call them Macro-system and Micro-systems respectively for now.

At any moment of our simulation we have  $N$  Micro-systems, one per bias  $i$ , defined by the Markov Matrix  $M^{(i)}$ .

Where we drop the time dependence of the bias in bin  $i$  for simplicity, without loss of generality.

The corresponding *True* Macro-system  $M$  is defined and constructed using the procedure described in Section 3.2 above.

At this point we can already see that:

- Each Micro-system satisfies Detailed Balance.

- The Macro-system is described by a Markov Matrix, and hence have a stationary distribution satisfying Global Balance.

When we generate a next position (/sample a transition from a starting point), we use the Micro-system corresponding to our starting position.

Hence, each transition is sampled using a process satisfying Detailed Balance.

When un-biasing using DHAM, what we do in fact is estimating the line  $i$  of the *unbiased* version of  $M^{(i)}$ , the Markov Matrix describing the Micro-system  $i$ , for each  $i$ , as described in Section 3.1 .  
Each of those matrices  $M^{(i)}$  satisfying Detailed Balance on their own.

Once we have the  $i^{th}$  line of each  $i^{th}$  M-matrix, we 'stitch' them conserving the line index, and we obtain our final unbiased Markov Matrix  $M^u$ .

Just like for the True Macro-system Transition Matrix, since we 'stitch' the lines of Transitions Matrices describing different Micro-systems, we are not guaranteed that our final Matrix follows DB, but it will always satisfy GB/SDB.

Alternatively, instead of seeing the bias as changing as we change bin, we can consider that our system jumps to pre-existing Micro-system (/ is 'lifted'), with probability 1 when it changes a bin.

### 3.4.4 First approach

First we consider that we wish to observe the less self-transitions possible (minimize residence time).

For now we will consider targeting a small self-transition probability  $M_{ii} \equiv \epsilon$

Also we consider that we want to observe different transitions approximately the same number of times (rare transitions should be made more likely and inversely).

Hence we can consider targeting a biased transition probability matrix of the form (here  $N = 5$ ):

$$M^{(b)} = \begin{pmatrix} \epsilon & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & 0 & 0 \\ \frac{1-\epsilon}{N-1} & \epsilon & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & 0 & 0 \\ \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \epsilon & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & 0 & 0 \\ 0 & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \epsilon & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & 0 \\ 0 & 0 & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \epsilon & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} \\ 0 & 0 & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \epsilon & \frac{1-\epsilon}{N-1} \\ 0 & 0 & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \frac{1-\epsilon}{N-1} & \epsilon \end{pmatrix} \quad (200)$$

Which have uniform equilibrium distribution and satisfies Detailed Balance.

Now assuming

$$M_{ij}^{(b_i)} = M_{ij} * e^{-\frac{\beta}{2}(b_j^i - b_i^i)} \propto M_{ij} * e^{-\frac{\beta}{2}(b_j^i - b_i^i)} \quad (201)$$

We have the following equation

$$\frac{1-\epsilon}{N-1} = M_{ij} * e^{-\frac{\beta}{2}(b_j^i - b_i^i)} \quad (202)$$

Hence

$$\frac{1}{M_{ij}} \frac{1-\epsilon}{N-1} = e^{-\frac{\beta}{2}(b_j^i - b_i^i)} \iff b_j^i - b_i^i = \frac{2}{\beta} (\log(M_{ij}) \frac{N-1}{1-\epsilon}) = \frac{2}{\beta} (\log(M_{ij}) + \log(\frac{N-1}{1-\epsilon})) \quad (203)$$

where  $b_i^i$  is a free parameter that we can first consider to be set to zero.

### 3.4.5 Application

In the following we simulated a unique instance of a system in a 1-Dimensional double well potential, using the previously derived biasing formula (8) with  $b_i^i = 0$  at first,  $\epsilon = 0$  and  $N = 9$ .

However, we deviated from the previously designed procedure by manually resetting  $b_i^i = (1.3) * \max_j(b_j^i)$  to make self-transitions less probable than without biasing (if we keep  $b_i^i = 0$  they become higher than without bias), and when considering the neighbours, we optimized the bias considering *at most* bin(i-4):bin(i+4), and for  $i = 1$  we would consider only the bins [1:5] and not [1:9] as we would if we followed the form of the target given by (5).

We simulate 1.000.000 Steps, with corrections of the bias in  $i$  every 300 counts in  $i$ .

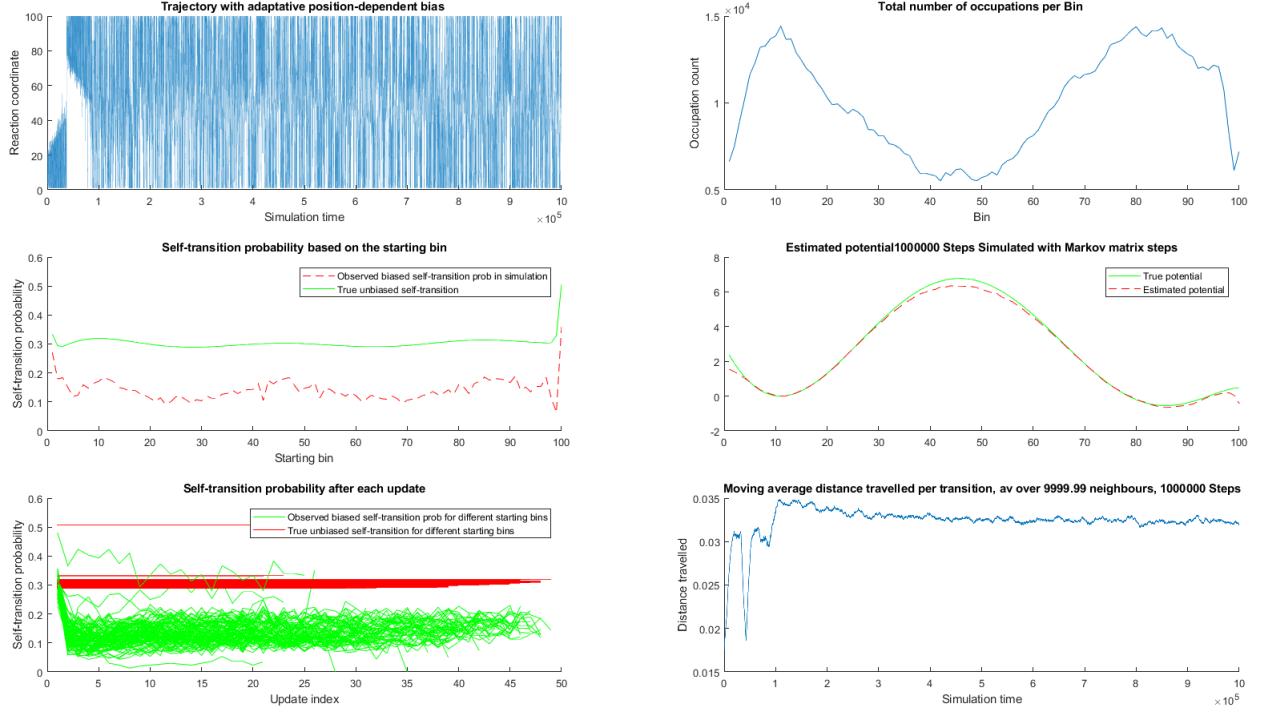


Figure 8: Markov Matrix simulation of our first approach to MDP-like biasing (1.000.000 steps, correction every 300 counts)

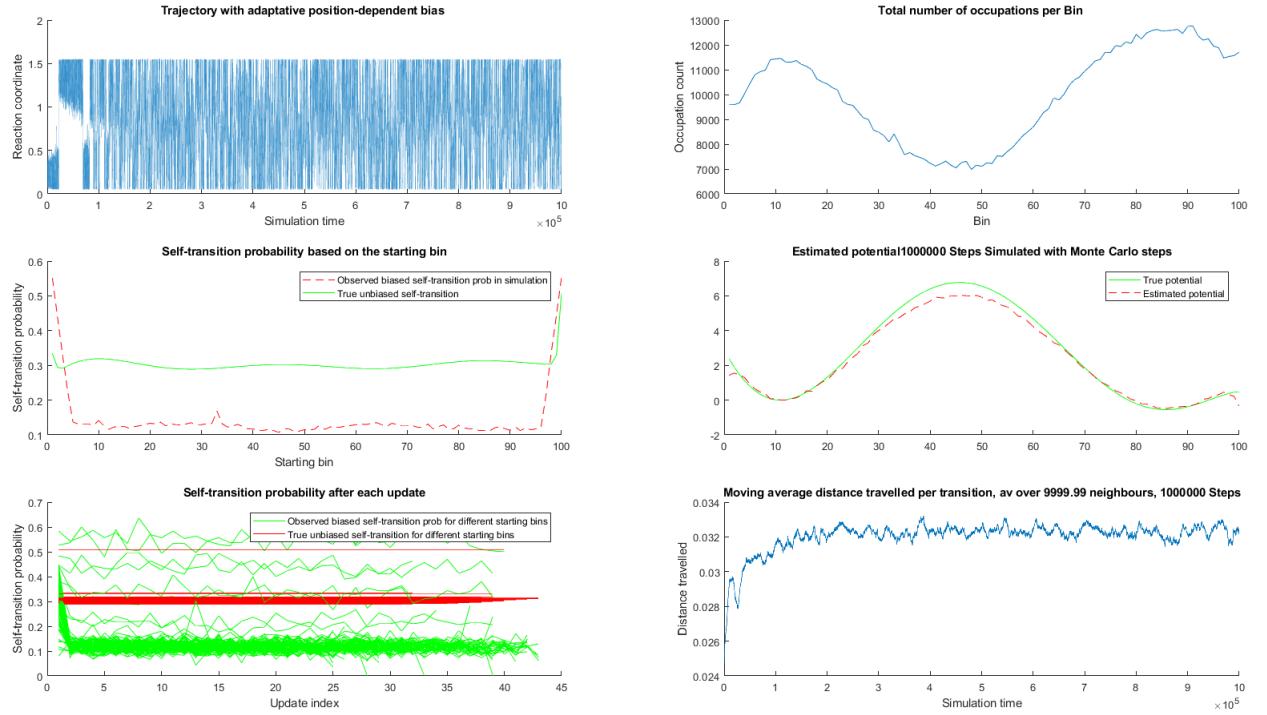


Figure 9: Monte Carlo simulation of our first approach to MDP-like biasing (1.000.000 steps, correction every 300 counts)

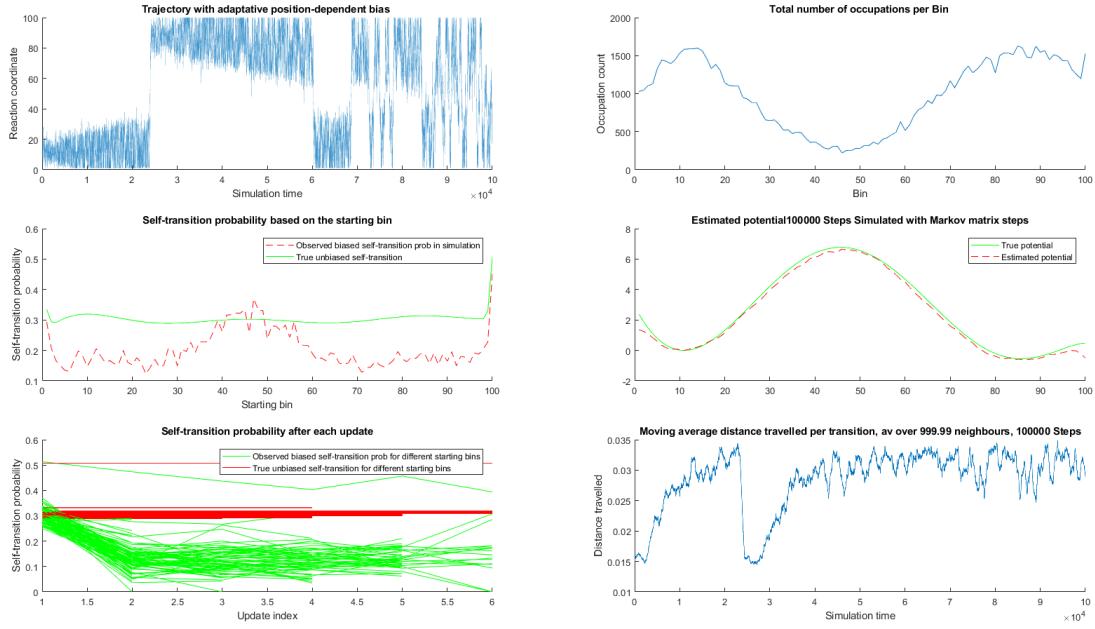


Figure 10: Markov Matrix simulation of our first approach to MDP-like biasing (100.000 steps, correction every 300 counts)

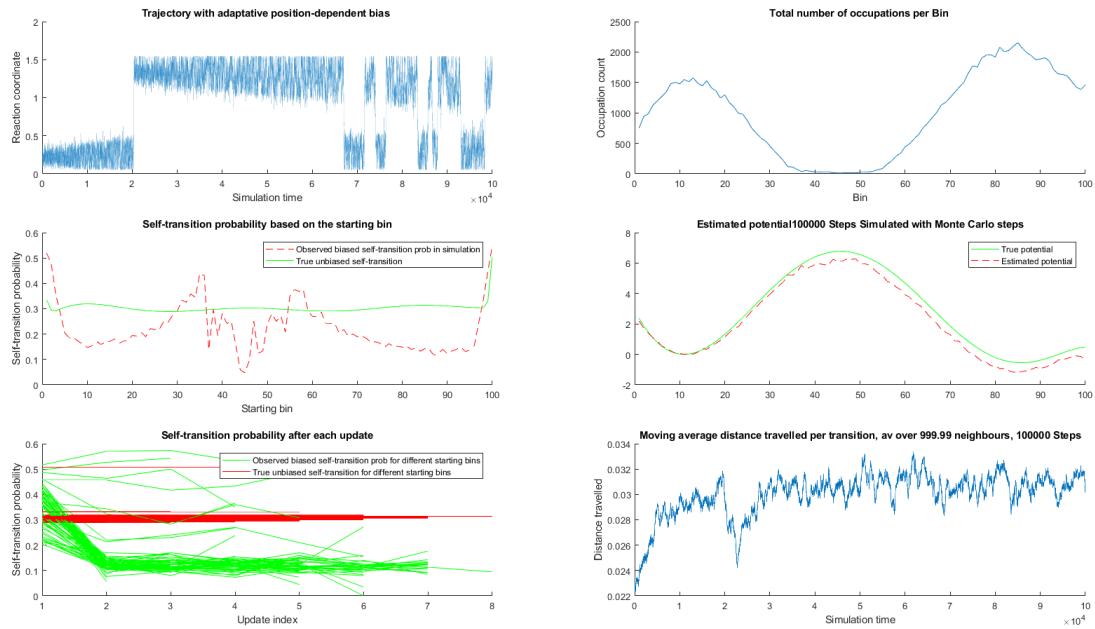
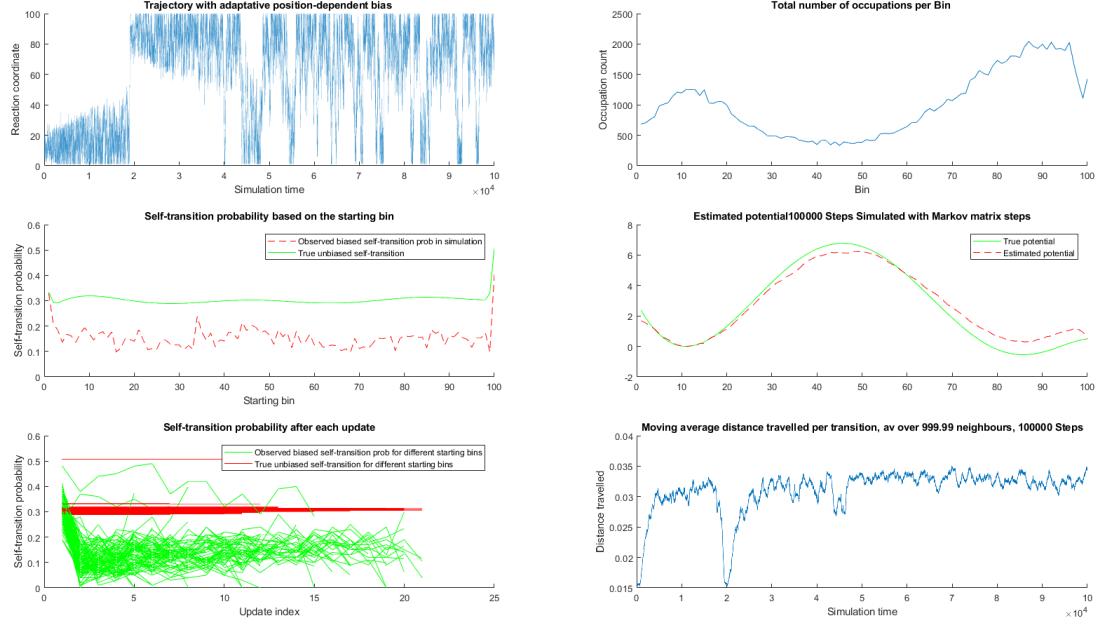
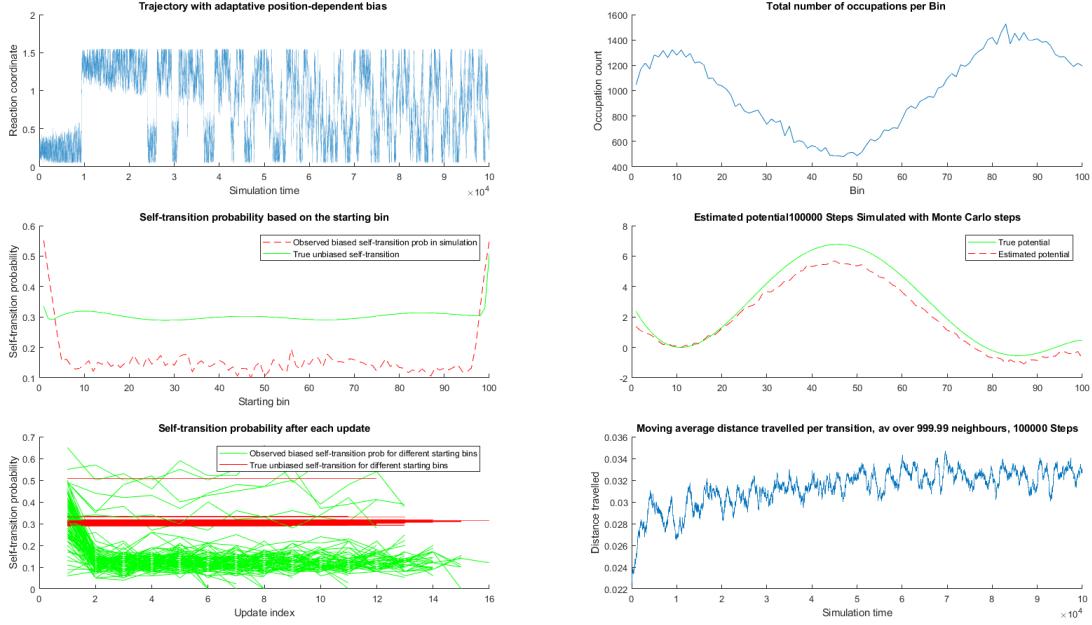


Figure 11: Monte Carlo simulation of our first approach to MDP-like biasing (100.000 steps, correction every 300 counts)



□

Figure 12: Markov Matrix simulation of our first approach to MDP-like biasing (100.000 steps, correction every 100 counts)



□

Figure 13: Monte Carlo simulation of our first approach to MDP-like biasing (100.000 steps, correction every 100 counts)

### 3.4.6 Conclusion

We have discussed heuristics on how to adapt Markov Decision Processes to the problem of enhanced sampling, and provided a first approach for the construction of the reward function.

We have shown that our estimates are independent from the starting position, hence we can use position-dependent bias, and we move from constructing a bias function to constructing a bias functional.

We provided an algorithm based on the minimization of the probability to stay in the same state, and uniformization of the transition probabilities.

We test the algorithm on an asymmetric 1-D potential using two simulation methods, and we show that we are indeed able to update de position-dependent biases along the simulation to increase space exploration, while estimating the right equilibrium probability.

This new approach opens the way of a whole new range of biasing techniques, where the bias is adapted depending on the current configuration of the system, and the history of the simulation.

### 3.4.7 Side Notes from Journal

Using the formula for the neighbours within a range, and setting bias in starting bin to zero, leads to increasing the av travelled distance, i.e. the probability to move further, however it also increases the probability to stay in the same bin (creates a local well).

To temporary fix this we used the max of the bias on the neighbouring bins to set the bias of the starting bin. We observe that, on the borders, the PMF estimates goes down and the self-transition probabilities are particularly high, even in the initial (non biased) markov matrix. The effect is stronger on the right side where the potential is close to be flat at the border, and supposedly the absence of explorable states further to the right makes so that the transitions that would have gone right are just going to the starting bin, this suggest that we shall bias the side-bin more, and decrease the bias even further on the left.

Also the estimates are not very good even if we have very long simulations. An hypothesis is that it is due to the fact that when estimating the pmf we use all the trajectory, even if for a given starting bin, the last bias may have been visited much less times than the others.

An idea is to discard the counts for biases that are not 'filled up' in terms of occupation.

It didn't seem to change anything.

Also we tried to change the range around the start bin, low range 'converge' quickly but do not improve the travelled distance much, larger range makes the system travel further but needs time to converge well and find the right bias, also even for long simulations it doesn't seem to converge properly, playing with the bias value at the starting bin allows to improve the result, but there is yet no analytical expression for the right value of  $b_i$ .

Increasing the range too far away seems to create errors even for very long simulations, and unexpected behaviour of the average travelled distance.

## 3.5 RESULT 4 (Unpublished) BIASING - Dancing Umbrellas

We can inspire from the idea to modify the bias as the simulation goes on from the MDP-like biasing method presented in the previous section, and apply it to the Umbrella Sampling method, where we adapt the two parameters (position and force) of the umbrella bias.

When one have chosen a reaction coordinate dimension along which he wishes to position umbrellas for umbrella sampling simulations, the most common choice is to place them equally separated, uniformly along the reaction coordinate of choice.

However, non uniformity of the free energy along this reaction coordinate leads this method to persistently non-uniform sampling along this coordinate, although more uniform than without umbrella sampling at all.

We will show below an automated procedure to adapt umbrella positions to uniformize the count distribution further as the simulation goes on, and will show that this leads to smaller estimation errors for an equal number of total steps.

The procedure is quite simple indeed and can be easily improved further.

### 3.5.1 The method

#### 3.5.1.1 Using Umbrella Simulations Histograms

Comparing the histograms obtained from different umbrella simulations gives us information about the sampling that have been performed, and in particular in the presence of barriers and wells, the histograms tends to diverge or cluster, i.e. deviate from the position of the umbrella.

In the case of uniformly positioned umbrellas, two adjacent umbrellas can provide disjoint or low-overlapping histograms, implying that some area between the two umbrellas have been poorly sampled, as can be seen as an example on Figure (58) below.

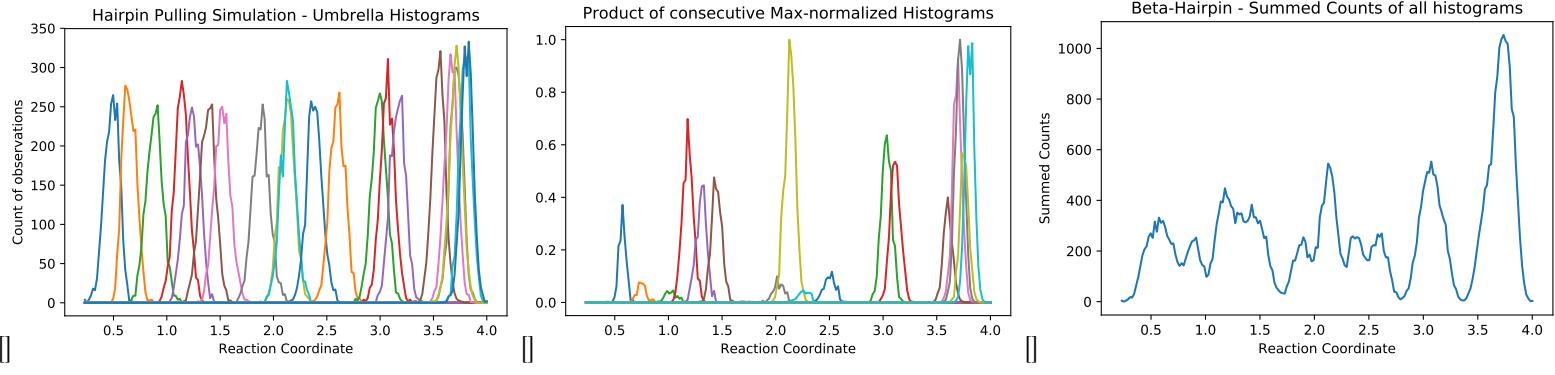


Figure 14: (a): Histograms from each umbrella simulation  
 (b): Product of consecutive max-normalized histograms  
 (c): Sum of the count of all the umbrella histograms along the Reaction Coordinate

In such a case it seems natural to think about adding another umbrella, somewhere in the middle of the two.

We would like to investigate how one can use the information provided by the histograms to determine the optimal parameters for umbrellas to use.

Two first ideas of variables to look at:

- Total Sampling (Sum of all the histograms) -> Position umbrellas at/around the minimas of the total histogram.
- Overlap (Product of adjacent max-normalised histograms) -> Position umbrellas at/around minimas of the overlap.

### 3.5.1.2 Varying the position and force of umbrellas

One can expect that varying the position of the umbrellas could allow for better sampling because if the umbrellas are positioned on a slope of the Free Energy function, the histogram of samples from this umbrella simulation are going to be biased towards the downside of the slope, thereby creating under-sampled or non-sampled ranges of the Reaction Coordinate.

We wish to investigate different umbrella sampling parametrization schemes for the positioning of the umbrellas, the objective of which is going to allow either better sampling along the reaction coordinate for a given computational power/quantity, or a similar sampling to the standard umbrella sampling at lower computational cost.

### 3.5.1.3 Defining a Target / Loss function to then apply gradient descent on parameters / Machine Learning

A first approach can be to assume than when we put uniformly distributed umbrellas, what we target is getting uniformly distributed histogram centers.

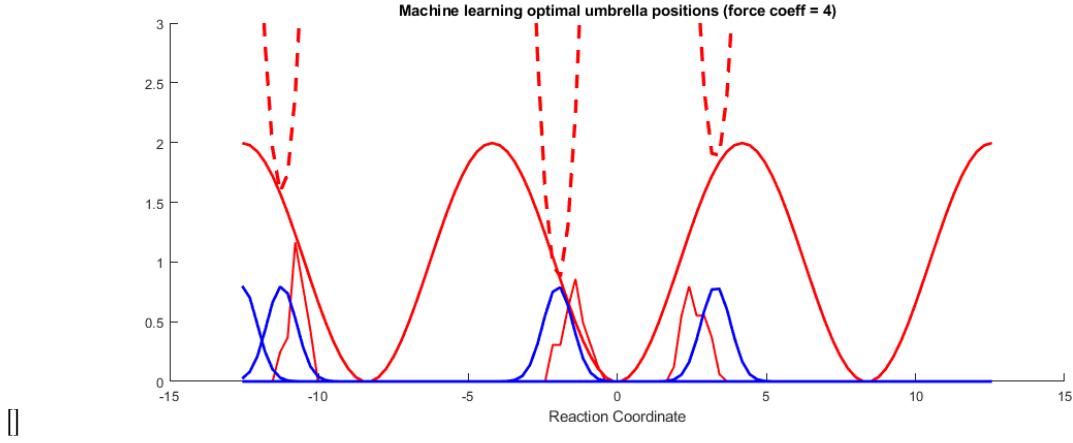


Figure 15: Sinusoidal true potential, three displayed umbrellas at their initial position (dotted red)  
Below, target histograms (blue) and sampled histograms (red)

Then each umbrella have its own target, and after some simulation time, for each umbrella the histogram of sampled positions is assessed and the position of the umbrella is adjusted in the opposite direction to the deviation of the sample histogram from the target.

From this idea we can build a function quantifying the distance of the umbrella output (count histogram), from the umbrella target.

We can in fact treat each umbrella simulation separately with respect to its own target. And parallelize the simulations.

In full generality, the loss function should have the following form:

$$L(\theta_U) = f(d(H^{Target}, H^{Obtained})) \quad (204)$$

Which, if the parameters of the umbrellas  $\theta_U$  are reduced to their positions, becomes:

$$L(X_U) = f(d(X_U, X_H)) \quad (205)$$

Where  $f(x)$  is the function defining how the distance is weighted, so that  $0 \leq \frac{\partial f(x)}{\partial |x|} \quad \forall x > 0$   
And  $d(x)$  is the distance metric.

A first Loss function to use is the quadratic distance:

$$L_1(X_U, F_U) = \frac{1}{2}(X_U - X_H)^2 \quad (206)$$

Then the gradient is simply

$$Grad = \frac{\partial L_1}{\partial X_U} = X_U - X_H \quad (207)$$

And we can update each umbrella's center with gradient descent

$$X_U = X_U - \alpha * Grad \quad (208)$$

Where  $\alpha$  is the learning rate.

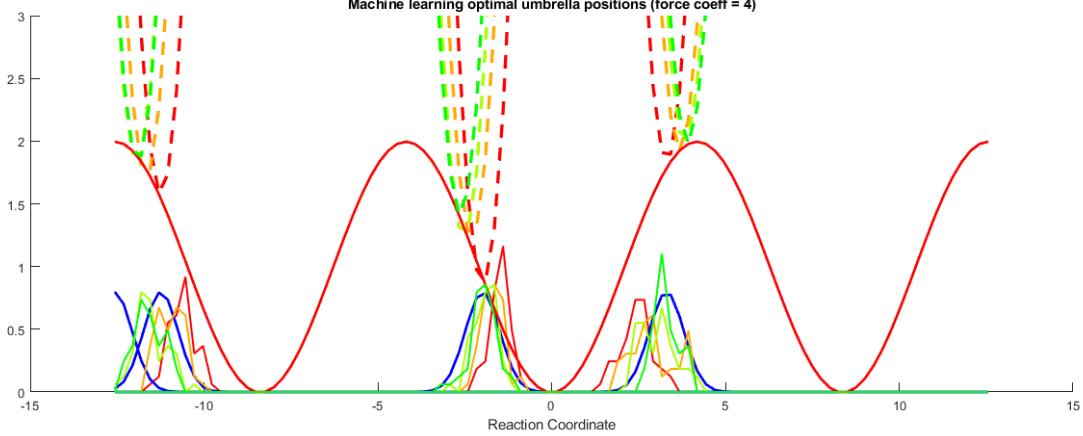


Figure 16: Same potential, three umbrellas after each parameter update  
Below, corresponding histograms for each update and unchanged target histograms (blue)

In the example above a potential of the form

$$V = \sin\left(\frac{3}{4}X - \frac{\pi}{2}\right) \quad (209)$$

was used, the umbrella parameters were updated 3 times using the quadratic loss function presented above, and a learning rate of 1.

One can note first that this method is totally unsupervised, except for the number of umbrellas and their force.

In addition, if the target is also updated each time, we can expect the umbrellas to perform unsupervised gradient descend *upwards* the free energy landscape.

This could allow to increase the reaction coordinate dimensionality and let the umbrellas walk towards the barriers, which is expected to be interesting.

### 3.5.2 Application and comparison to standard umbrella sampling

We have performed Monte Carlo simulations on the potential described above, for each replicas using 50 umbrellas with force constant 3.

The total number of simulated steps was taken to be 4000.

We compare the results between simulations with 2 corrections (umbrella position update) to similar simulations using standard umbrella sampling.

The total computational time was split first among all umbrellas equally ( $4000/50 = 80$  steps for each initial umbrella).

Then, for the case with umbrella update, the computational time was split further equally for each session:  
Steps between updates for each umbrella = Steps for each umbrella/ $(1 + \text{Nb updates})$

Each such set of parallel simulations was used to estimate a PMF using standard DHAM, each estimate have been vertically shifted to be equal to the true potential at the first well.

We repeated this process 100 times (replicas) to obtain average PMFs and their estimated deviations.  
For all umbrella position updates, the learning rate was 0.5.

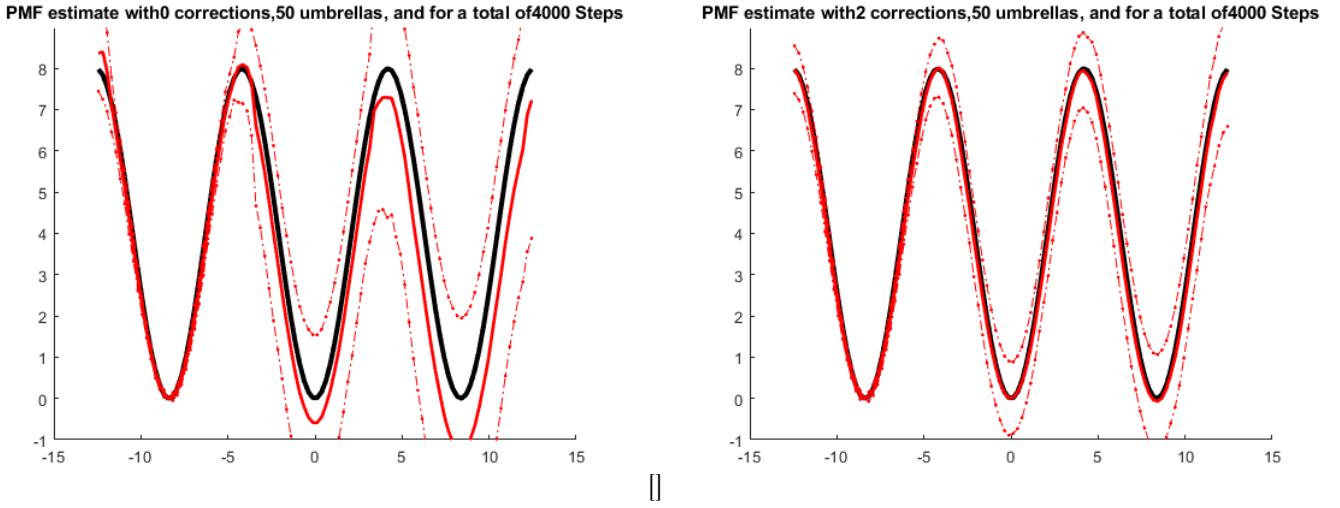


Figure 17: Red line: average PMF estimate over 100 replicas, Dotted line:  $\pm$  average  $\text{Abs}(\text{Vest}-\text{V})$

Where we see sharp increases in the variance oh the estimate at the position of the barriers.

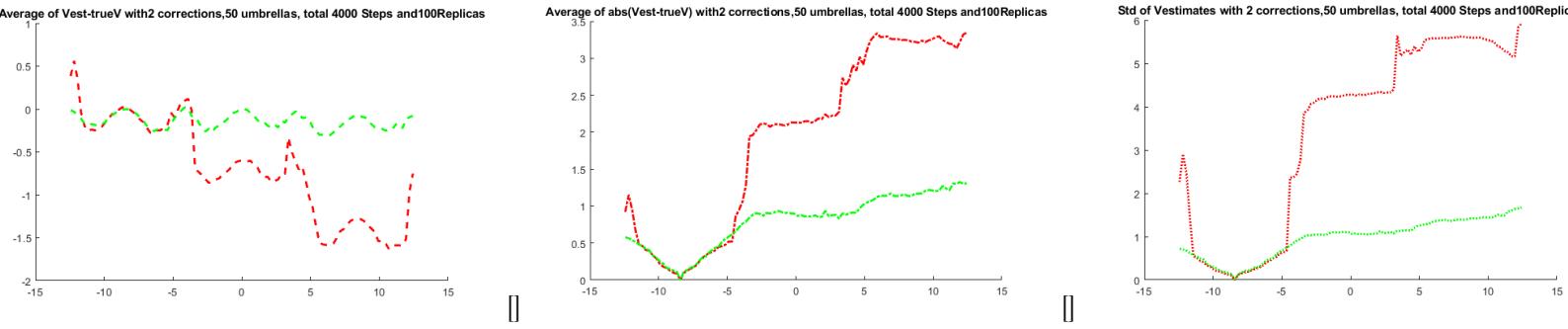


Figure 18: Red: Standard Umbrella Sampling / Green: 2 Umbrella position Updates

### 3.5.2.1 Changing perspective: RMSD minimizing shifted V estimates:

Instead of equating all the estimates at a given point, we can shift them vertically to minimize the RMSD between them and the true potential.

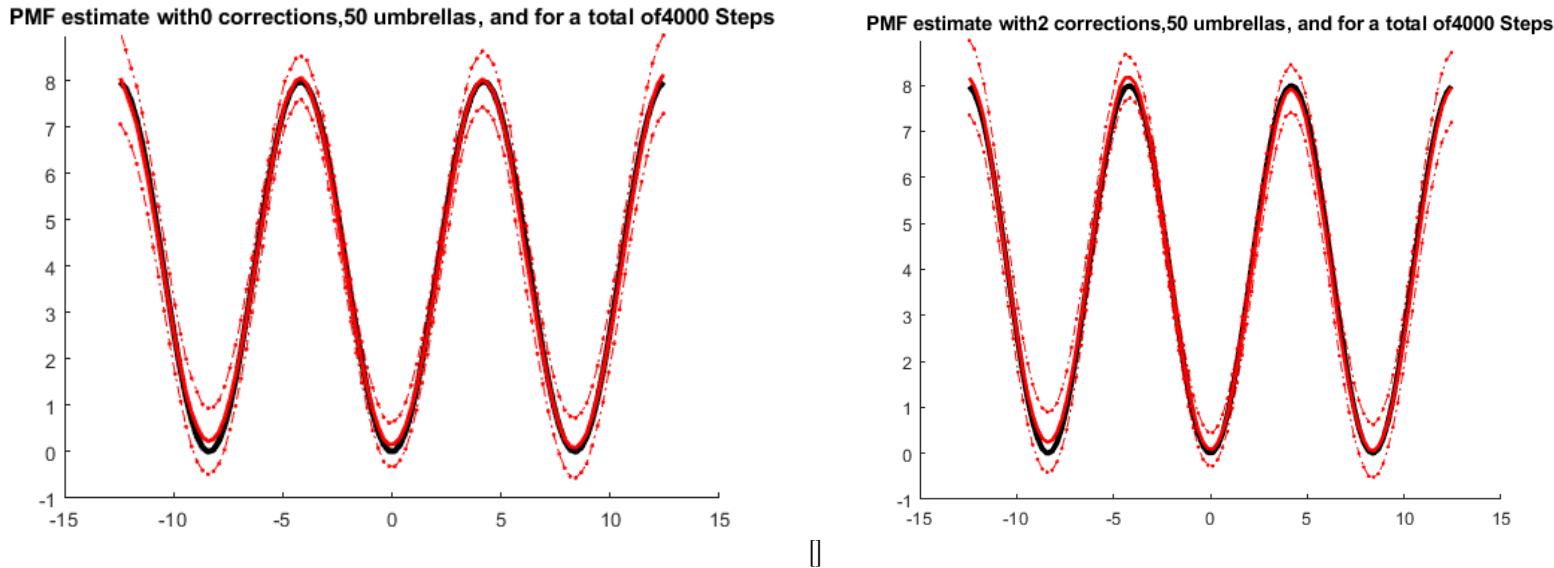


Figure 19: Red line: average PMF estimate over 100 replicas, Dotted line:  $\pm$  average  $\text{Abs}(\text{Vest}-\text{V})$

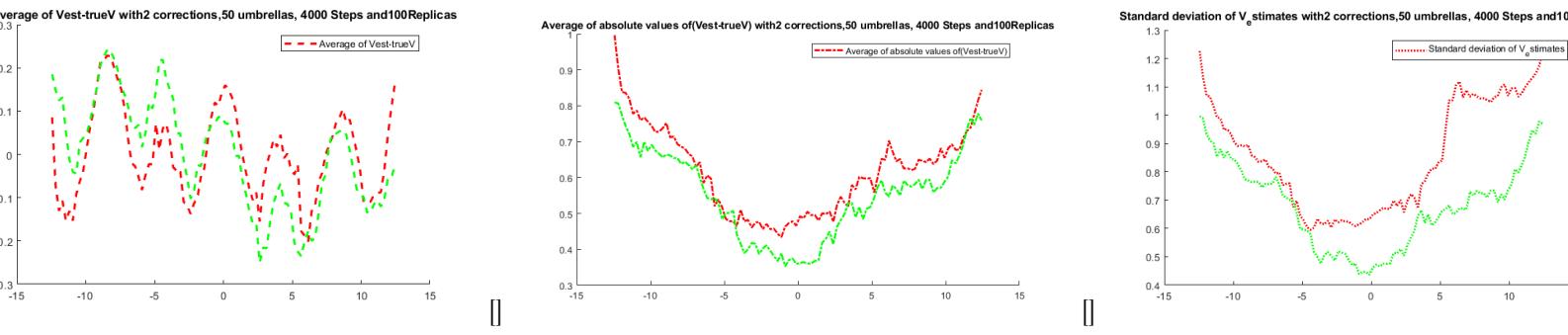


Figure 20: Red: Standard Umbrella Sampling / Green: 2 Umbrella position Updates

### 3.5.2.2 Less steps

We have performed the same simulations with twice less steps (2000), also with 100 replicas for statistics, and kept the RMSD shifting.

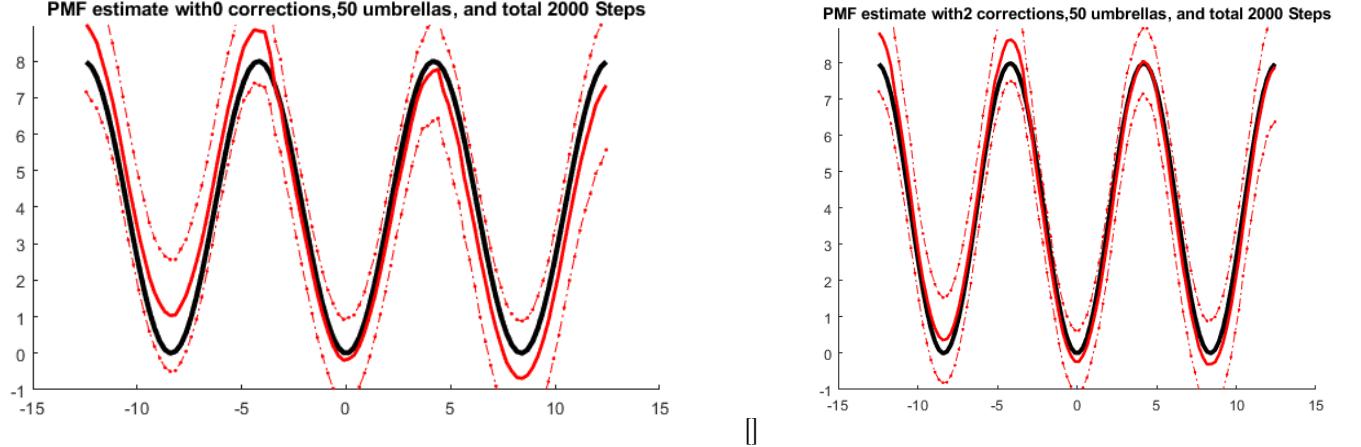


Figure 21: Red line: average PMF estimate over 100 replicas, Dotted line:  $\pm$  average  $\text{Abs}(\text{Vest}-\text{V})$

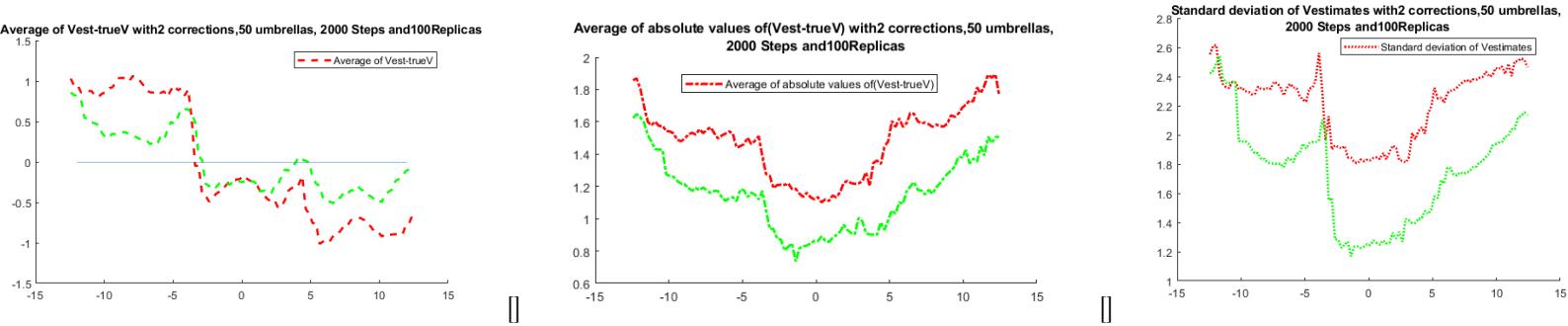


Figure 22: Red: Standard Umbrella Sampling / Green: 2 Umbrella position Updates

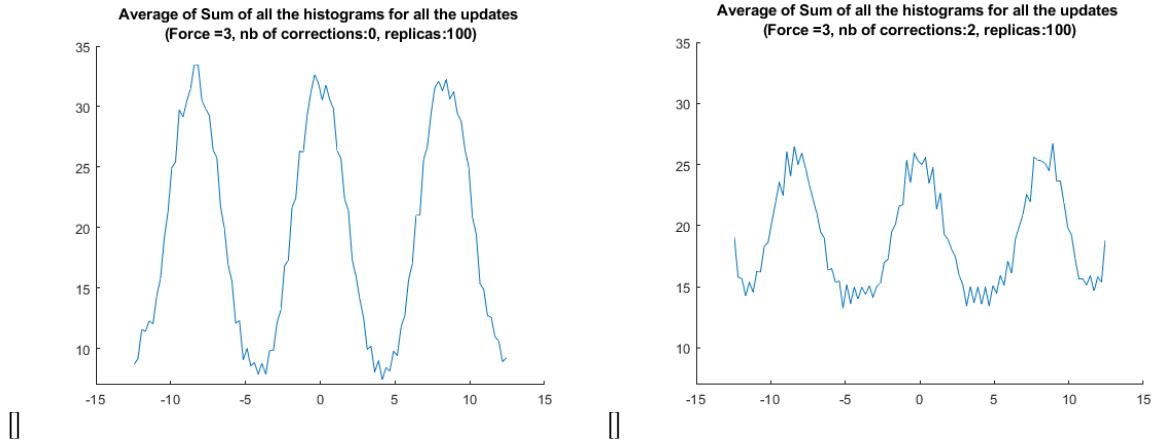


Figure 23: Average of total counts for each bin along the reaction coordinate.

This method indeed uniformizes the total counts better than regular uniformly positioned umbrella sampling simulations, and reduces the estimation error.

### 3.5.2.3 Asymmetric double well

We also applied this method to a non-symmetric double well potential, using also 50 umbrellas and a force constant of 100, and a total of 3000 Steps shared among each umbrella, for each replica. Only 50 replicas in this case.

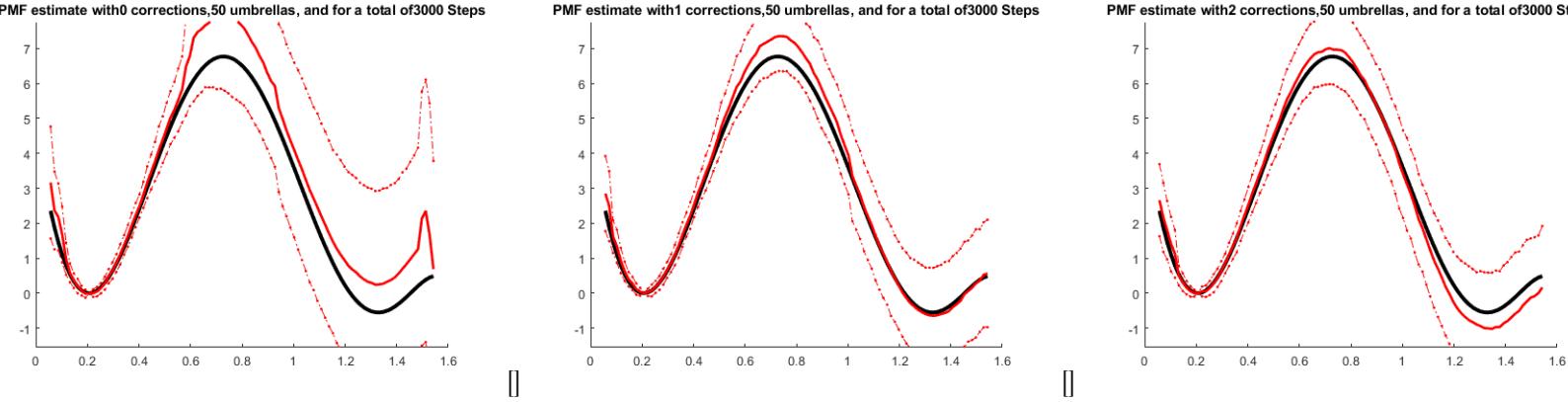


Figure 24: Red line: average PMF estimate over 50 replicas, Dotted line:  $\pm$  average  $\text{Abs}(\text{Vest}-\text{V})$

Various error measures of PMF estimates with 2 corrections, 50 umbrellas, 3000 Steps and 50Replicas

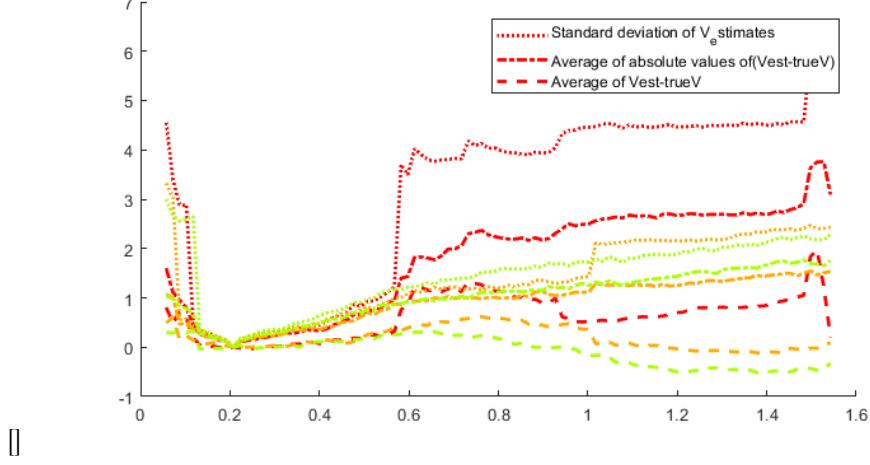


Figure 25: Red: Standard Umbrella Sampling / Orange: 1 Umbrella position update/ Green: 2 Umbrella position updates

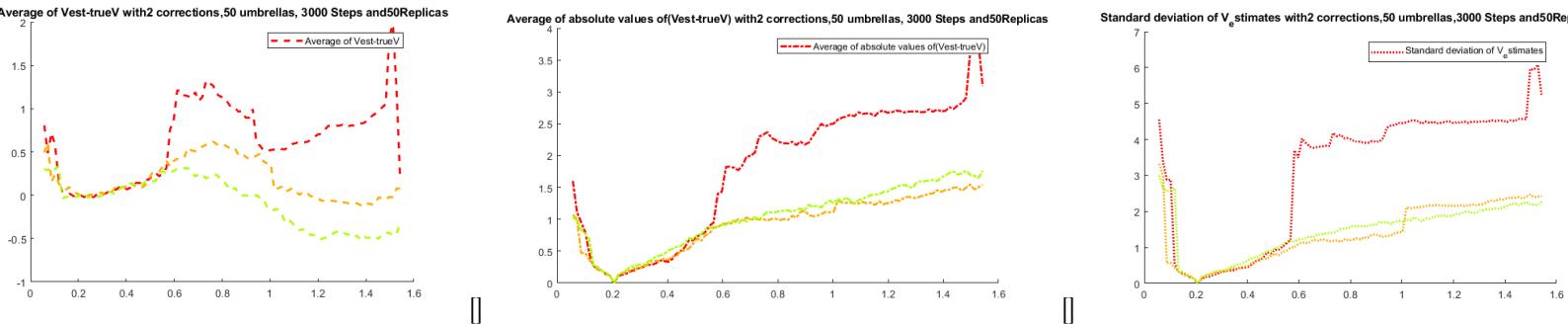


Figure 26: Red: Standard Umbrella Sampling / Orange: 1 Umbrella position update/ Green: 2 Umbrella position updates

### 3.5.3 Analytic result for position correction / PMF gradient $\leftrightarrow$ histogram deviation

Consider the first-order Taylor series expansion of the potential  $V(x)$  around  $x_0$

$$V(x) \sim V(x_0) + \frac{\partial V(x_0)}{\partial x}(x - x_0) \quad (210)$$

Upon redefining the terms as following:

$$V_{x_0}(x) \equiv V(x) - V(x_0) \quad \text{And} \quad d(x_0) \equiv \frac{\partial V(x_0)}{\partial x} \quad (211)$$

We can write

$$V_{x_0}(x) = d(x_0)(x - x_0) \quad (212)$$

Now consider this potential under a quadratic bias such as the biased potential  $V^{(b)}$  reads as follows

$$V_{x_0}^{(b)}(x) = k(x - x_0)^2 + d(x_0)(x - x_0) \quad (213)$$

Which can be rewritten as

$$V_{x_0}^{(b)}(x) = kZ^2 - \frac{d(x_0)}{2k} \quad \text{Where} \quad Z \equiv (x - x_0) + \frac{d(x_0)}{2k} \quad (214)$$

Which shows that the biased potential remains quadratic and the position of its minimum is for  $Z = 0$

$$Z = 0 \Leftrightarrow x = x_0 - \frac{d(x_0)}{2k} \Leftrightarrow d(x_0) = 2k(x_0 - x) \Leftrightarrow x_0 = x + \frac{d(x_0)}{2k} \quad (215)$$

This result allows us to estimate the gradient of the potential using the deviation of our observed histograms  $x$  from the bias center  $x_0$ , and the force constant used  $k$ .

Moreover, once the gradient is estimated, one can use the equation in reverse to obtain the optimal bias center  $x_0$  for an arbitrary target position  $x_T$

$$x_0 = x_T + \frac{d(x_0)}{2k} \quad (216)$$

### 3.5.4 Adapting the Umbrella forces

#### 3.5.4.1 Optimal Target Parameters for overlap

When setting our target histograms, we want the histograms to overlap well.

The first trivial step is to set the umbrella centers uniformly along the RC/string.  
However then follows the question of the target variance for the histograms.

Here we quickly address the question of the optimal target variance for histograms obtained from umbrella sampling simulations.

#### 3.5.4.2 Derivation

Formulation of the problem:

Given  $n_u$  and  $D$  (respectively nb of umbrellas and Total length of the RC/string) What are the optimal histogram characteristics ( $\mu_i$  and  $\sigma$ ) to target?  
(Assumed  $\sigma_i = \sigma_j \forall i, j$ )

We also introduce a variable  $\xi$ , representing the spacing between the border of the RC/string and the first umbrella, same for all the borders.

Hence follows:

$$\mu_i = \xi + (i - 1)[\frac{D - 2\xi}{n_u - 1}] \quad (217)$$

To decide which variance we want, we need to determine the overlap we look for.

We consider that a good approach is to seek for the distance between two consecutive umbrellas to be  $2\sigma$  or slightly less, and the distance to border to be of  $\sigma$  so that we have a new condition:

$$\begin{cases} 2\sigma \equiv \mu_{i+1} - \mu_i = \frac{D - 2\xi}{n_u - 1} \\ \xi \equiv \sigma \end{cases} \quad (218)$$

Hence

$$2\sigma = \frac{D - 2\sigma}{n_u - 1} \iff \sigma = \frac{D}{2n_u} \quad (219)$$

For an arbitrary multiple of the standard deviation  $\alpha$ , we have

$$\alpha\sigma \equiv \mu_{i+1} - \mu_i = \frac{D - 2\xi}{n_u - 1} \iff \sigma = \frac{D - 2\xi}{\alpha(n_u - 1)} \quad (220)$$

If  $\xi = \sigma$  we get

$$\sigma = \frac{D}{\alpha n_u + (2 - \alpha)} \quad (221)$$

### 3.5.4.3 Additional

Consider a biasing potential on a flat surface:

$$b = k(x - \mu_0)^2$$

This gives the following distribution / histogram:

$$P(x) = \frac{1}{Z} e^{-\beta k(x - \mu_0)^2} \quad (222)$$

Which is equivalent to the normal distribution with  $\mu = \mu_0$  and  $\sigma^2 = \frac{1}{2\beta k}$

Now with a non-flat surface:

$$V^b = V + b = \delta(x - \mu_0)^2 + k(x - \mu_0)^2$$

Hence

$$P(x) = \frac{1}{Z} e^{-\beta(\delta+k)(x - \mu_0)^2} \quad (223)$$

Which is equivalent to the normal distribution with  $\mu = \mu_0$  and  $\sigma^2 = \frac{1}{2\beta(\delta+k)}$

### 3.5.4.4 Updating parameters to match target

We are in a case where we want to observe a histogram with some defined variance, and are ready to bias the system towards it, but we don't know the true potential.

The procedure is to start with an umbrella with some force/variance, and then update the umbrella parameter to obtain histograms with observed variance closer and closer to the target.

**Method 1** Loss function:

$$L(k) = \frac{1}{2} [\sigma_T^2 - \sigma_O^2(k)]^2 \quad (224)$$

Where  $\sigma_T^2$  and  $\sigma_O^2$  are the Target and Observed histogram variance.

Considering  $\sigma_O^2(k) = \frac{1}{2\beta(k+\delta)}$  we have:

$$\frac{\partial E}{\partial k} = (\sigma_T^2 - \sigma_O^2(k)) \frac{\partial \sigma_O^2(k)}{\partial k} = \frac{\sigma_O^2 - \sigma_T^2}{2\beta(k + \delta)} \quad (225)$$

Using a learning rate  $\lambda = 0.2$  we update the umbrella force using the following formula:

$$k_{new} = k_{old} - \lambda \frac{\partial E}{\partial k} \quad (226)$$

**Method 2**

$$\sigma_O^2(k) = \frac{1}{2\beta(k + \delta)} \quad (227)$$

$$\sigma_O^2(k)^{new} = \sigma_O^2(k)^{old} + \xi \iff \frac{1}{2\beta k^{new}} = \frac{1}{2\beta k^{old}} + \xi \iff k^{new} = \frac{k^{old}}{2\beta\xi + 1} \quad (228)$$

Using a speedup coefficient  $\lambda$  such that the update formula becomes:

$$k^{new} = \frac{k^{old}}{2\beta\xi\lambda + 1} \quad (229)$$

We get quicker convergence.

**Method 3**

$$\sigma_T^2 = \frac{1}{2\beta k_T} \quad (230)$$

$$k_T - k_O = Err$$

$$k_{new} = k_{old} + \lambda(Err) \quad (231)$$

### 3.5.4.5 Application

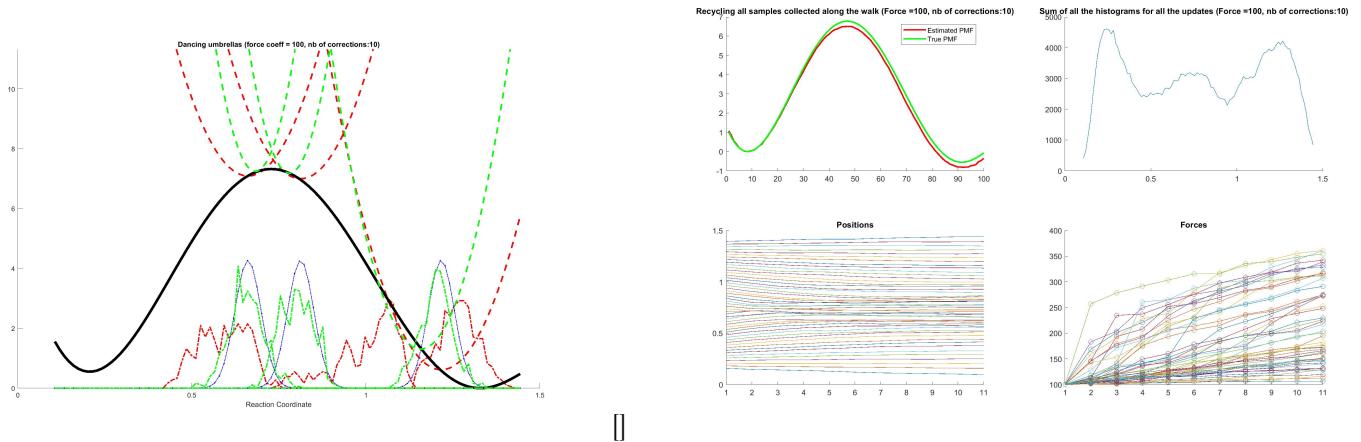


Figure 27: 10 Corrections over 300.000 simulated steps

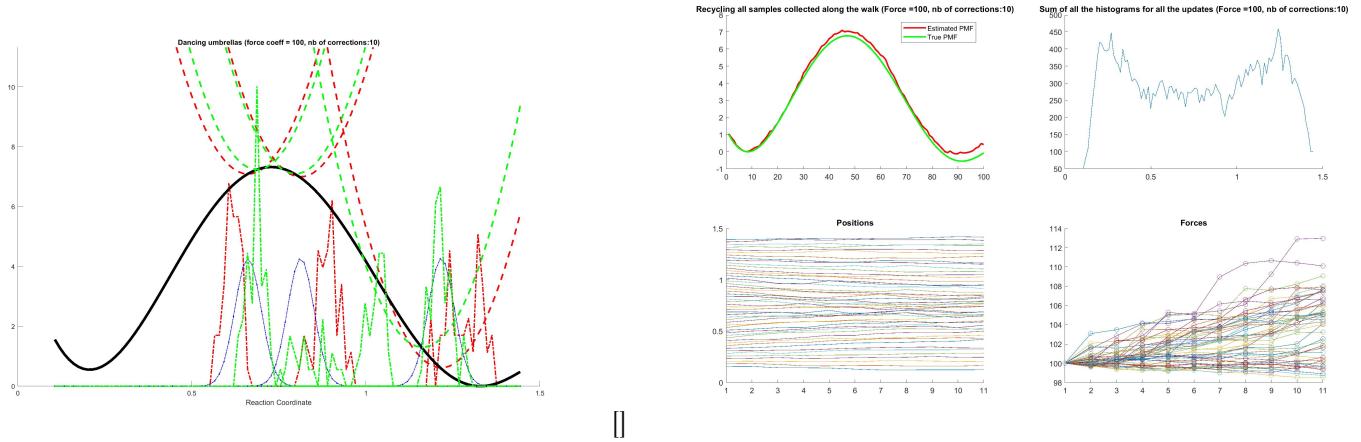


Figure 28: 10 Corrections over 30.000 simulated steps

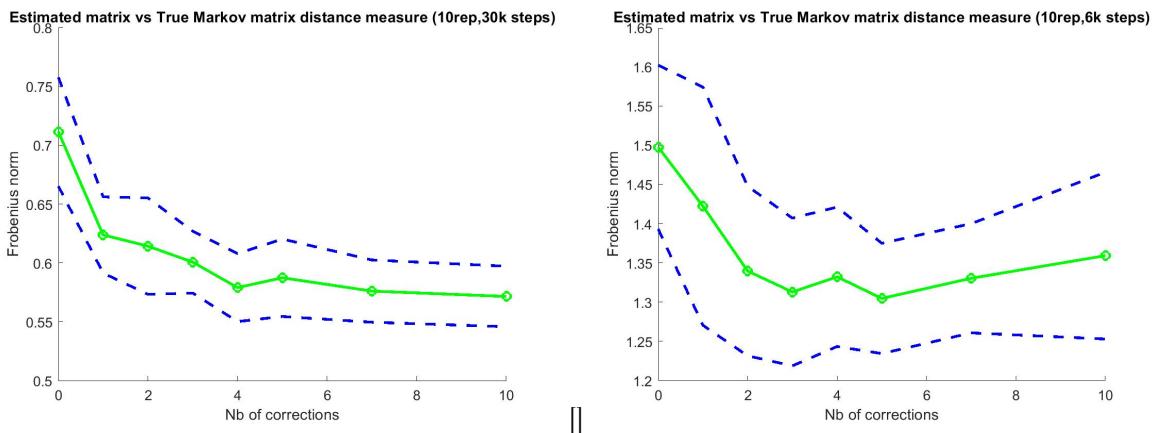


Figure 29: Estimated matrix error as a function of nb of corrections, over 30.000 (a) and 6.000 (b) simulated steps

### 3.5.5 Application to 2-dimensional potential

Simulation settings:

Number of bins: 20\*20 / Number of umbrellas: 9 / Force of umbrellas: 8

We use a 2D potential with 3 wells, and we construct a basic string by connecting the well centers by a straight line. (This string can later be optimized, but it is out of the scope of the present work and this very simple string is good enough for the point we ought to make.)

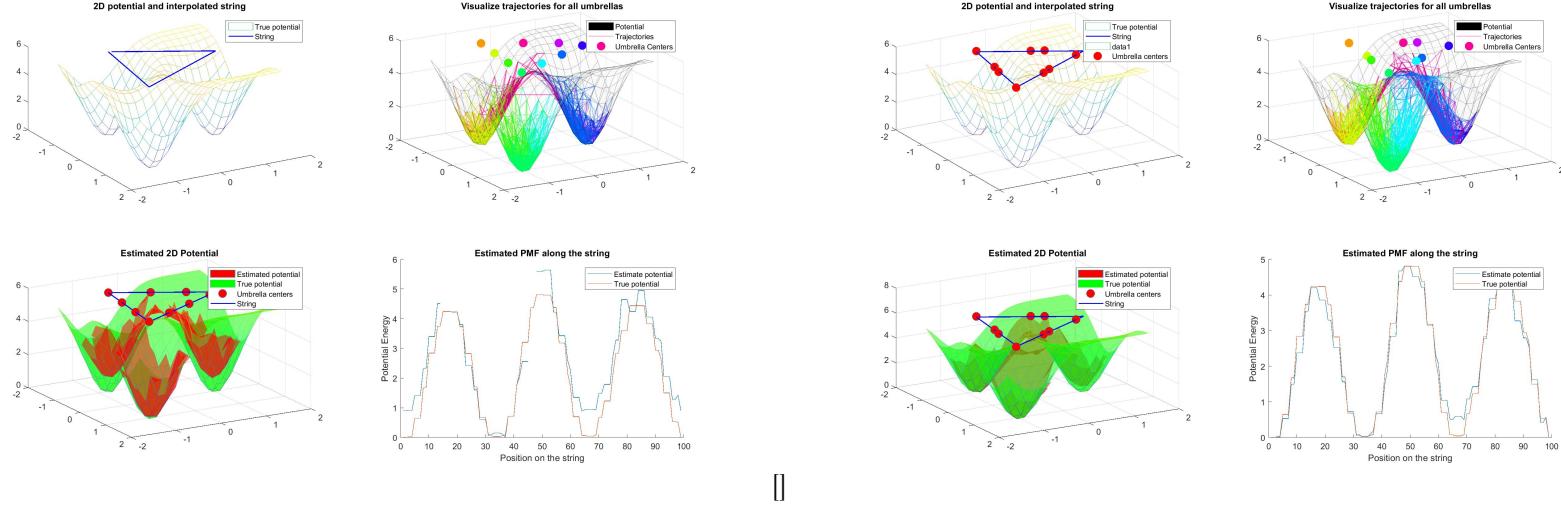


Figure 30: 30.000 simulated steps, uniformly positioned (a) and with position adapted by 1 correction (b)

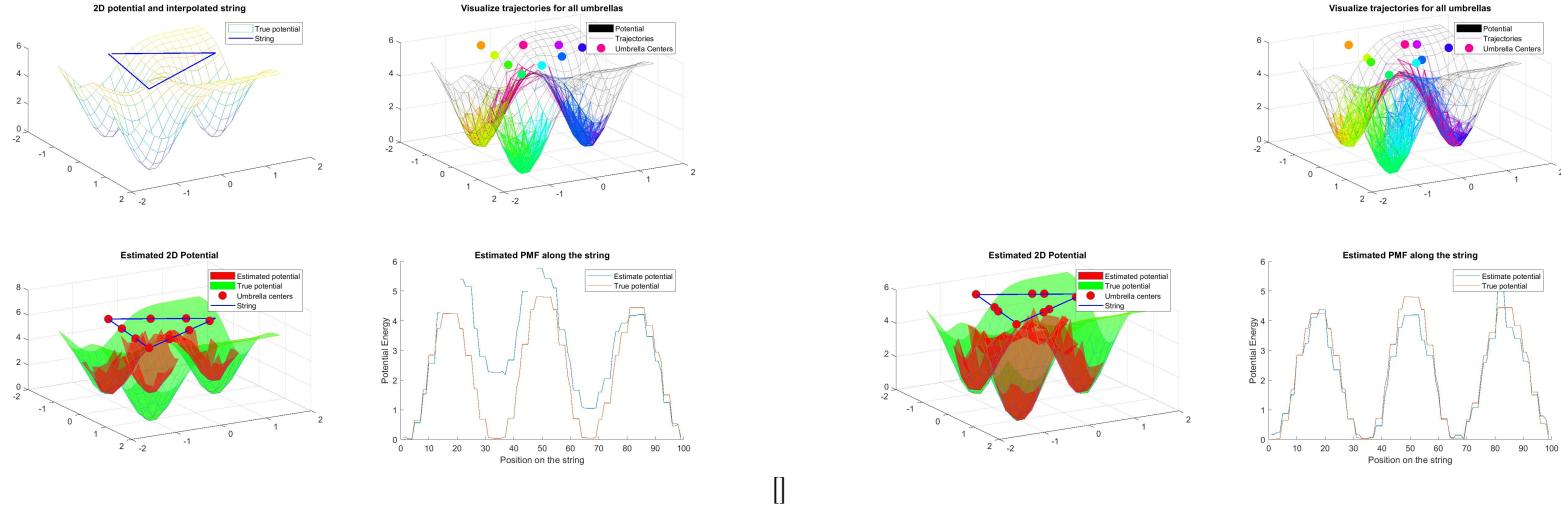


Figure 31: 8.000 simulated steps, uniformly positioned (a) and with position adapted by 1 correction (b)

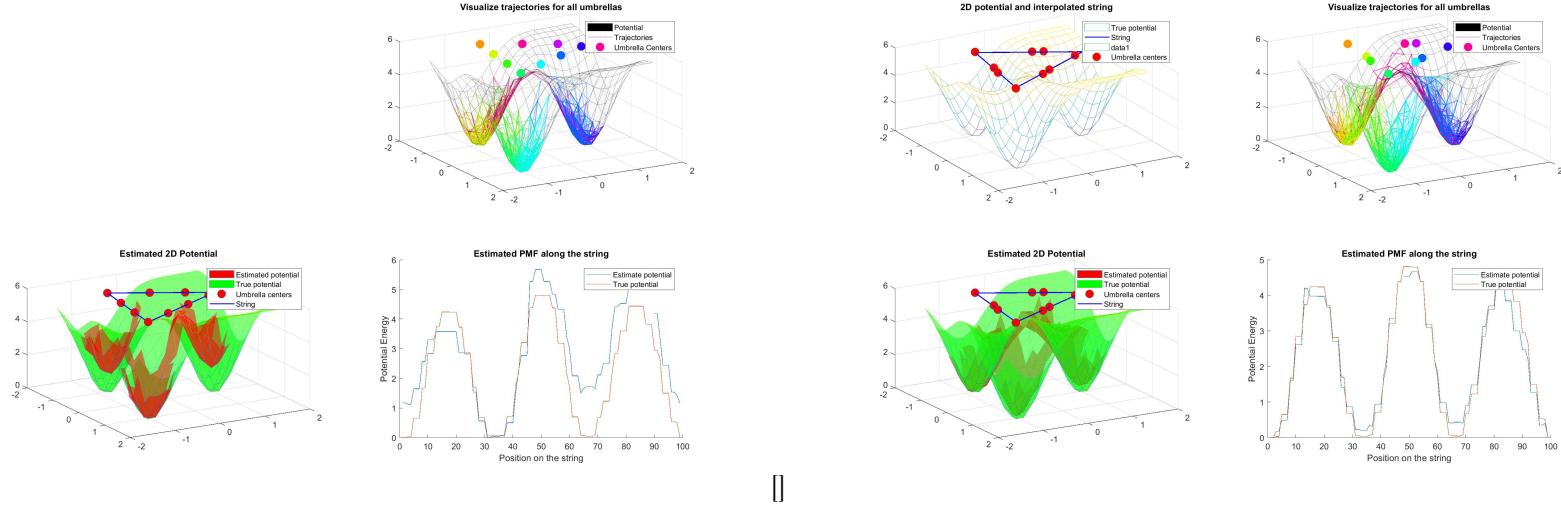


Figure 32: 4.000 simulated steps, uniformly positioned (a) and with position adapted by 1 correction (b)

Here we only optimized the position of the umbrellas along the string, and we were able to obtain much better PMF estimates with 4.000 simulated steps than with 30.000 simulated steps and uniformly positioned umbrellas...

### 3.5.6 Conclusion

We have introduced a modified version of the well-known Umbrella Sampling methods, inspired from MDP-like biasing and Supervised Learning methods in Machine Learning. We presented an algorithm to adapt the position and force parameters of the umbrella biases during the simulation, by "learning" the parameters that allows to obtain the target sample histogram.

We have tested the algorithm on an asymmetric 2-well 1-D potential, and a 3-well 2-D potential. This method effectively allows to obtain more uniform position counts, and it appears that only a few corrections along the simulation are optimal for the considered systems.

In future work, we could try to adapting the number of umbrellas during the simulation, and use non-symmetric umbrellas to compensate the local gradient and create a "flat" surface locally.

## 3.6 RESULT 5 (Unpublished) BIASING - Cyclic Lifting

### 3.6.1 Introduction

It has been shown that breaking Detailed Balance can accelerate the convergence of the system to equilibrium [60].

When using DHAM for estimating the dynamics of the system, by performing simulations we are actually (interested in) sampling the *transition* space instead of the *configuration* space. All we consider and estimate the probabilities of, are transitions starting from a given state.

Since the sets of transition counts from one "starting state" are independent from the counts from another "starting state", we can design a different bias for each state to create a net flux and break Detailed Balance.

#### 3.6.1.1 Origin of the idea

Consider a system with  $N$  nodes (set to 100), only nearest neighbours transitions, self-transition probability equal to zero, and the clockwise transition probabilities ( $P_{forward}$ ) are equal for all nodes.

In this case, there is only one free parameter for the system,  $P_{forward}$ , and setting it to  $\frac{1}{2}$  corresponds to the case where Detailed Balance (DB) is satisfied, otherwise the system breaks DB.

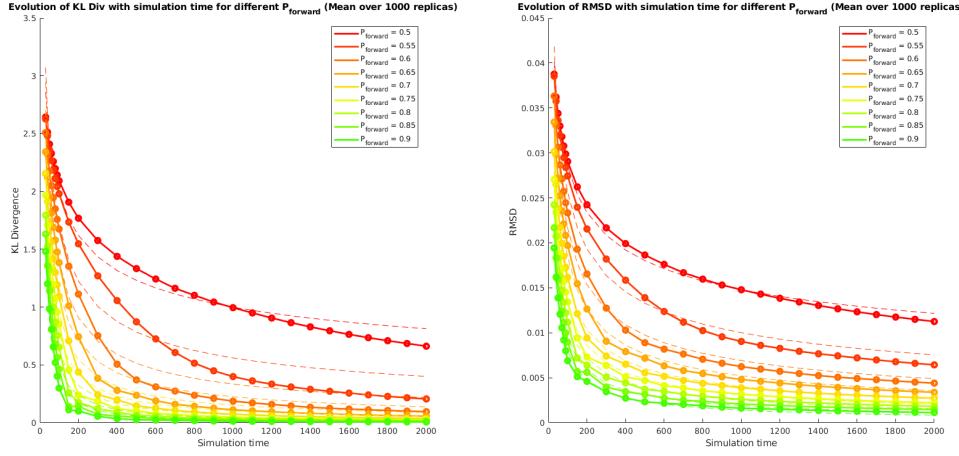


Figure 33: Errors vs Simulation time for different  $P_{forward}$  values, and their power law fit (dotted)

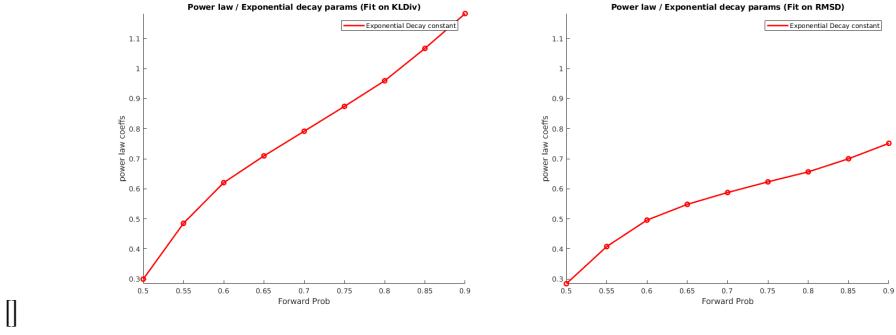


Figure 34: Exponent parameter of the Power Law fits

We can see that when the system have a flux, it converges quicker to its equilibrium distribution. From this observation, and from the emerging field of Lifted trajectories for spin-systems and particle systems [58, 59], we aim to develop a new biasing technique that breaks Detailed Balance in order to accelerate configurational exploration and convergence to equilibrium distribution.

The novelty of our approach is multiple:

We use a novel biasing method that breaks detailed balance by using a position-dependent bias shape. We are doing a sort of Lifting [58], but we do not require nor wish for the equilibrium distribution to remain unchanged by the lifting, in fact our target will usually imply a uniform equilibrium distribution after lifting. We pair the lifting with an enforcement of periodic boundaries, to further improve space exploration. Breaking Detailed Balance is not a problem because we use DHAM to estimate the MSM, and DHAM do not actually require DB to estimate Markovian dynamics accurately (See Sec. (3.4.3.3)).

### 3.6.2 Turning a 1-D space into a circular network

If we are given a 1-Dimensional flat potential, how can we turn the 1-Dimensional support into a circular support, so that we can run simulations and still get the initial Markov Matrix back ?

The general idea is to enforce periodic boundaries, to connect the two ends of our 1D linear support. However the two ends of our support do not have the same potential value and cannot be connected 'smoothly'.

Here we will consider that we are manipulating a markov model, extracted from a potential or just given, and the simulations will be made using the Markov Matrix to walk on the corresponding Markov Chain.

If we artificially 'add' a probability to an element of the markov matrix, and that we re-normalize the matrix after this change, only the corresponding line (starting node) will be affected.

Also, If we run a simulation with the new matrix, collect the transitions, discard the 'periodic boundary' transitions from the transition count matrix, as if they were sort of 'lifting moves' and not 'physical moves', and then estimate the markov matrix, we can directly obtain the 'unbiased' estimate.

In our case we want to modify the probabilities  $M_{1,N}$  and  $M_{N,1}$ , which are zero if the system is sufficiently large, so that after the simulations we can simply discard *all* the transitions  $T_{1,N}$  and  $T_{N,1}$  and obtain the unbiased transition probability matrix.

As previously mentioned, here we consider dealing with a Markov Matrix (obtained from a given Potential through Arrhenius rates), and simulating from it, which corresponds to proposing moves with the conditional/posterior probability  $\mathbf{P}(x|i)$  when being in state  $i$  and accepting them with probability 1.

If we were simulating using a different Proposal-Acceptance algorithm, like MC, we might enforce the Boundary Condition differently. (usually the borders have barriers with acceptance probability to cross being 0, we can increase the acceptance probability of crossing the border and then 'teleport' the system to the other side when the crossing is accepted).

At the end all that matters is that we can discard the 'periodic boundary'/'teleport' moves from the transition count matrix before estimating the MSM.

### 3.6.3 Create a net flux

Creating a net flux is equivalent to breaking Detailed Balance.

There are reasons to think that we cannot break DB by simply adding a bias to the potential, but it is possible by using position-dependent biases. We can construct the bias to enforce a condition, called 'Target' in the following, to obtain a specific flux:

#### 3.6.3.1 Target A1

$$\alpha = \frac{M''_{i,i+1}}{M''_{i,i-1}} = \frac{\pi_i M''_{i,i+1}}{\pi_i M''_{i,i-1}} = \frac{\mathbb{E}(i \rightarrow (i+1))}{\mathbb{E}((i-1) \leftarrow i)} \approx \frac{T_{i,i+1}}{T_{i,i-1}} \quad \forall i \in [2 : N - 1] \quad (232)$$

#### 3.6.3.2 Explicit solution - Flux

Assuming that  $M_{ij} \propto e^{-\frac{\beta}{2}[V_j - V_i]}$ , we have:

$$\alpha = \frac{M''_{i,i+1}}{M''_{i,i-1}} = \frac{e^{-\frac{\beta}{2}[V''_{i+1} - V''_i]}}{e^{-\frac{\beta}{2}[V''_{i-1} - V''_i]}} = e^{-\frac{\beta}{2}[V''_{i+1} - V''_{i-1}]} = e^{-\frac{\beta}{2}[V_{i+1} - V_{i-1}]} e^{-\frac{\beta}{2}[b_{i+1}^{(i)} - b_{i-1}^{(i)}]} \quad (233)$$

$$\iff \log(\alpha) = -\frac{\beta}{2}[(V_{i+1} - V_{i-1}) + (b_{i+1}^{(i)} - b_{i-1}^{(i)})] \quad (234)$$

$$\iff -\frac{2}{\beta} \log(\alpha) = V_{i+1} - V_{i-1} + b_{i+1}^{(i)} - b_{i-1}^{(i)} \iff b_{i+1}^{(i)} - b_{i-1}^{(i)} = -[V_{i+1} - V_{i-1} + \frac{2}{\beta} \log(\alpha)] \quad (235)$$

Since we have equal sized bins, we can define  $\Delta_x \equiv x(i+1) - x(i) \quad \forall t$ , where  $x(i)$  is the center of bin  $i$  in Reaction Coordinates, and get:

$$\partial b(i) = \frac{b_{i+1} - b_{i-1}}{2\Delta_x} = -\frac{1}{2\Delta_x}[V_{i+1} - V_{i-1} + \frac{2}{\beta} \log(\alpha)] \quad (236)$$

This defines the gradient of our bias at the bin  $i$ ,  $\partial b(i)$ .

We use this to build the following *linear* bin-dependent bias:

$$b^{(i)}(x) = \partial b(i)(x - x(i)) \quad \forall i \in [2, N - 1] \quad (237)$$

Note that it is defined over the Reaction Coordinate space.

We can extend the definition of the bias gradient to  $i = 1$  and  $N$ :

$$V_{i+1} - V_{i-1} \approx 2(V_i - V_{i-1}) \approx 2(V_{i+1} - V_i) \Rightarrow \partial b(i) \approx \begin{cases} -\frac{1}{2\Delta_x}[2(V_{i+1} - V_i) + \frac{2}{\beta} \log(\alpha)] & i = 1 \\ -\frac{1}{2\Delta_x}[2(V_i - V_{i-1}) + \frac{2}{\beta} \log(\alpha)] & i = N \end{cases} \quad (238)$$

Where the approximation corresponds to approximating the potential to be locally linear, which makes sense when the size of the bins is small compared to the curvature of the potential.

### 3.6.3.3 Explicit solution - Periodic Boundary

If we consider allowing 'periodic moves' only for the very border bins (1 and N):

$$\begin{cases} i \equiv 1 \Rightarrow \alpha = \frac{M'_{1,2}}{M'_{1,N}} \\ i \equiv N \Rightarrow \alpha = \frac{M'_{N,1}}{M'_{N,N-1}} \end{cases} \Rightarrow \begin{cases} M'_{1,N} \equiv \frac{1}{\alpha} M'_{1,2} \\ M'_{N,1} \equiv \alpha M'_{N,N-1} \end{cases} \quad (239)$$

In practice we assume that the initial (non-biased) probabilities  $M_{1,N}$  and  $M_{N,1}$  are zero, we add the given probability to the moves, we end up with non-normalized lines 1 and N, which we can simply normalize without breaking the flux condition.

After the simulation is performed we will simply discard all transitions between bin 1 and N. ('periodic moves')

### 3.6.3.4 Adaptive solution - Flux

### 3.6.3.5 Adaptive solution - Periodic Boundary

### 3.6.3.6 Target A2

$$\alpha = \frac{\sum_{j>i} M''_{i,j}}{\sum_{j< i} M''_{i,j}} = \frac{\sum_{j>i} \pi_i M''_{i,j}}{\sum_{j< i} \pi_i M''_{i,j}} = \frac{\mathbb{E}(i \rightarrow +)}{\mathbb{E}(i \rightarrow -)} \approx \frac{T_{i,+}}{T_{i,-}} \quad \forall i \in [2 : N-1] \quad (240)$$

### 3.6.3.7 Explicit solution - Flux

Assuming that  $M_{ij} \propto e^{-\frac{\beta}{2}[V_j - V_i]}$ , we have:

$$\alpha = \frac{\sum_{j>i} M''_{i,j}}{\sum_{j< i} M''_{i,j}} = \frac{\sum_{j>i} e^{-\frac{\beta}{2}[V_j'' - V_i'']} }{\sum_{j< i} e^{-\frac{\beta}{2}[V_j'' - V_i'']} } = \frac{\sum_{j>i} e^{-\frac{\beta}{2}V_j''} }{\sum_{j< i} e^{-\frac{\beta}{2}V_j''} } = \frac{\sum_{j>i} e^{-\frac{\beta}{2}V_j} e^{-\frac{\beta}{2}b_j^{(i)}} }{\sum_{j< i} e^{-\frac{\beta}{2}V_j} e^{-\frac{\beta}{2}b_j^{(i)}}} \quad (241)$$

No real explicit solution found so far...

### 3.6.3.8 Explicit solution - Periodic Boundary

But there is a nice one for the periodic boundary:

$$\begin{cases} i \equiv 1 \Rightarrow \alpha = \frac{\sum_{j>1} M'_{1,j}}{M'_{1,N}} \\ i \equiv N \Rightarrow \alpha = \frac{M'_{N,1}}{\sum_{j<1} M'_{N,j}} \end{cases} \Rightarrow \begin{cases} M''_{1,N} \equiv \frac{1}{\alpha} (1 - M'_{1,1}) \\ M''_{N,1} \equiv \alpha (1 - M'_{N,N}) \end{cases} \quad (242)$$

### 3.6.3.9 Target B1

$$\alpha = \frac{\pi_i M''_{i,i+1}}{\pi_{i+1} M''_{i+1,i}} = \frac{\mathbb{E}(i \rightarrow (i+1))}{\mathbb{E}(i \leftarrow (i+1))} \approx \frac{T_{i,i+1}}{T_{i+1,i}} \quad \forall i \in [2 : N-1] \quad (243)$$

### 3.6.3.10 Explicit solution - Flux

Assuming again that  $M_{ij} \propto e^{-\frac{\beta}{2}[V_j - V_i]}$ , and that  $\pi_x \propto e^{-\beta V_x}$ , we have:

$$\alpha = \frac{\pi_i M''_{i,i+1}}{\pi_{i+1} M''_{i+1,i}} = \frac{e^{-\beta V_i''} e^{-\frac{\beta}{2}[V_{i+1}'' - V_i'']} }{e^{-\beta V_{i+1}''} e^{-\frac{\beta}{2}[V_i'' - V_{i+1}'']} } = \frac{e^{-\frac{\beta}{2}[V_{i+1}'' + V_i'']} }{e^{-\frac{\beta}{2}[V_i'' + V_{i+1}'']} } = \frac{e^{-\frac{\beta}{2}[V_{i+1} + V_i]} }{e^{-\frac{\beta}{2}[V_i + V_{i+1}]} } \frac{e^{-\frac{\beta}{2}[b_{i+1}^{(i)} + b_i^{(i)}]} }{e^{-\frac{\beta}{2}[b_i^{(i+1)} + b_{i+1}^{(i+1)}]} } \quad \forall i \in [2 : N-1] \quad (244)$$

(NOTE that the matrix  $M''_{i,j}$  describes the system to which was applied the bias  $b^{(i)}$ , and therefore the condition above describes a degree of Detailed Balance between two different systems, but also describes the degree of Detailed Balance of the matrix constructed for the lines  $i$  of the matrices  $M_{i,j}^{(i)}$ ), BUT the use of  $V^{(i)}$  for  $\pi_i$  is (very) questionable.)

$$\alpha = \frac{e^{-\frac{\beta}{2}[b_{i+1}^{(i)} + b_i^{(i)}]} }{e^{-\frac{\beta}{2}[b_i^{(i+1)} + b_{i+1}^{(i+1)}]} } \quad \forall i \in [2 : N-1] \quad (245)$$

Since the choice of bias is relevant only up to a constant term, we chose our linear bias to be  $b^{(i)}(x) = \partial b(i)(x - x(i))$ , hence  $b_i^{(i)} = 0, \forall i$ .

$$\alpha = e^{-\frac{\beta}{2}[b_{i+1}^{(i)} - b_i^{(i+1)}]} \Leftrightarrow b_{i+1}^{(i)} - b_i^{(i+1)} = -\frac{2}{\beta} \log(\alpha) \quad (246)$$

Here we obtain a relation between two different (consecutive) biases.

$$-\frac{2}{\beta} \log(\alpha) = \partial b(i)(x(i+1) - x(i)) - \partial b(i+1)(x(i) - x(i+1)) = \Delta_x [\partial b(i) + \partial b(i+1)] \quad (247)$$

$$\partial b(i+1) = -[\partial b(i) + \frac{2}{\Delta_x \beta} \log(\alpha)] \quad (248)$$

We can note that also taking  $\pi_i = \pi_{i+1}$  yields the exact same result.

If we were to build our biases iteratively following this condition we would have  $\partial b(i+2) = \partial b(i)$ .

### 3.6.3.11 Explicit solution - Periodic Boundary

$$\alpha = \frac{\pi_N M_{N,1}''}{\pi_1 M_{1,N}''} = \frac{e^{-\frac{\beta}{2}(b_N^{(N)} + b_1^{(N)})}}{e^{-\frac{\beta}{2}(b_N^{(1)} + b_1^{(1)})}} = e^{-\frac{\beta}{2}(b_1^{(N)} - b_N^{(1)})} \quad (249)$$

$$-\frac{2}{\beta} \log(\alpha) = b_1^{(N)} - b_N^{(1)} = \partial b^{(N)} + \partial b^{(1)} \quad (250)$$

### 3.6.3.12 Target B1 alt

$$\alpha = \frac{\pi_i M_{i,i+\delta}''}{\pi_{i+\delta} M_{i+\delta,i}''} = \frac{\mathbb{E}(i \rightarrow (i+\delta))}{\mathbb{E}((i+\delta) \leftarrow i)} \approx \frac{T_{i,i+\delta}}{T_{i+\delta,i}} \quad \forall i \in [2 : N-1] \quad (251)$$

### 3.6.3.13 Explicit solution - Flux

Assuming again that  $M_{ij} \propto e^{-\frac{\beta}{2}[V_j - V_i]}$ , we have:

$$\alpha = \frac{\pi_i M_{i,i+\delta}''}{\pi_{i+\delta} M_{i+\delta,i}''} = e^{-\frac{\beta}{2}(b_{i+\delta}^{(i)} - b_i^{(i+\delta)})} \quad (252)$$

$$\Leftrightarrow \partial b^{(i)} + \partial b^{(i+\delta)} = -\frac{2}{\beta} \frac{1}{\Delta_x \delta} \log(\alpha) \quad (253)$$

### 3.6.3.14 Explicit solution - Periodic Boundary

Same as B1 since  $\delta = 1$  between 1 and N.

### 3.6.3.15 Target B2

$$\alpha = \frac{\sum_{j>i} \pi_j M_{i,j}''}{\sum_{j>i} \pi_i M_{j,i}''} = \frac{\mathbb{E}(i \rightarrow)}{\mathbb{E}(i \leftarrow)} \approx \frac{\sum_{j>i} T_{i,j}}{\sum_{j>i} T_{j,i}} \quad \forall i \in [2 : N-1] \quad (254)$$

### 3.6.3.16 Explicit solution - Flux

Nothing much can be done explicitly here, it seems.

### 3.6.3.17 Explicit solution - Periodic Boundary

...

### 3.6.3.18 Iterative solution

Every target can be aimed for iteratively, using the corresponding Observable  $O$ .

Taking simply the following quadratic Loss function:  $L(b) = \sqrt{(O(b) - \alpha)^2}$ , where the loss depend on our bias  $b$ , we can decompose:

$$\frac{\partial L(O(b))}{\partial b} = \frac{\partial L(O)}{\partial O} \frac{\partial O(b)}{\partial b} = (O - \alpha) \frac{\partial O(b)}{\partial b}. \quad (255)$$

Now, we might not know the value of  $\frac{\partial O(b)}{\partial b}$ , but we can determine its sign, which needs and is assumed to be constant, using heuristics.

We can then update  $b$  using the following update:

$$b_{new} = b_{old} + (\text{Sign}(\frac{\partial O(b)}{\partial b})) \lambda (O - \alpha) \quad (256)$$

Where the learning rate  $\lambda$  have to be chosen arbitrarily.

### 3.6.4 First implementation

In our first implementation on a symmetric potential, for the n-biasing we used condition (A1) and for periodic boundary we use the condition (A2):

$$(A2)(i = N) \quad \alpha = \frac{\sum_{j>i} M''_{i,j}}{\sum_{j>i} M''_{i,j}} =_{i=N} \frac{M''_{N,N+1}}{1 - M_{N,N}} \iff M''_{N,N+1} = M''_{N,1} = \alpha(1 - M_{N,N}) \quad (257)$$

$$(A2)(i = 1) \quad \alpha = \frac{\sum_{j>i} M''_{i,j}}{\sum_{j>i} M''_{i,j}} =_{i=1} \frac{1 - M_{1,1}}{M''_{1,0}} \iff M''_{1,0} = M''_{1,N} = \frac{1}{\alpha}(1 - M_{1,1}) \quad (258)$$

Then we renormalize the 1st and last lines by  $1 + \alpha(1 - M_{N,N})$  and  $1 + \frac{1}{\alpha}(1 - M_{1,1})$  respectively.

### 3.6.5 First implementation, results

- The target  $\alpha = 1$ , equivalent to detailed balance, worked surprisingly well.
- When increasing the flux we notice a shift of the estimated PMF towards the direction of the shift, even for very long simulations (See Fig. (46), bottom right). We will show in the next section that this shift can be totally suppressed by performing two parallel simulations with fluxes of equal strengths going to opposite directions.
- There is a problem with the borders and periodic boundaries, the system spends too much time 'down-the-riven', as can be seen on the 'Observed Transition Counts' plots (See Fig. (39), bottom left). This problems reduces a lot when performing the backwards flux simulation, but is also caused by the fact that we did not apply any bias to the bins 1 and N, so that the system was jumping from bin 1 to bin 4/5, without visiting bins 2/3.

### 3.6.5.1 Symmetric potential, 2000 simulation steps, Alpha = 1,2, and 5

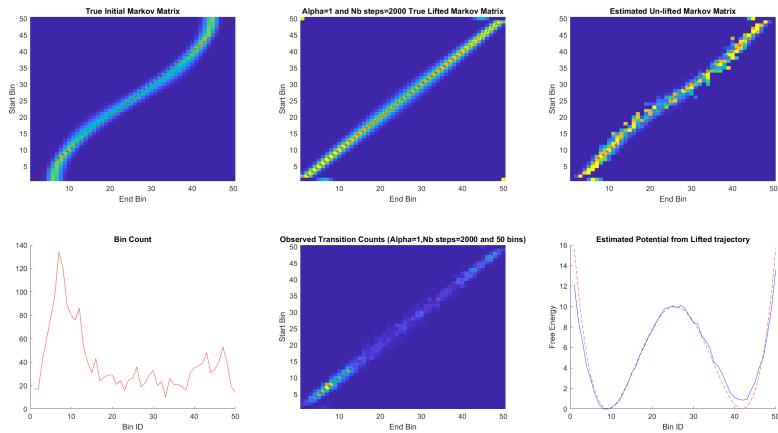


Figure 35: 50 Bins, 2000 Steps, Alpha = 1

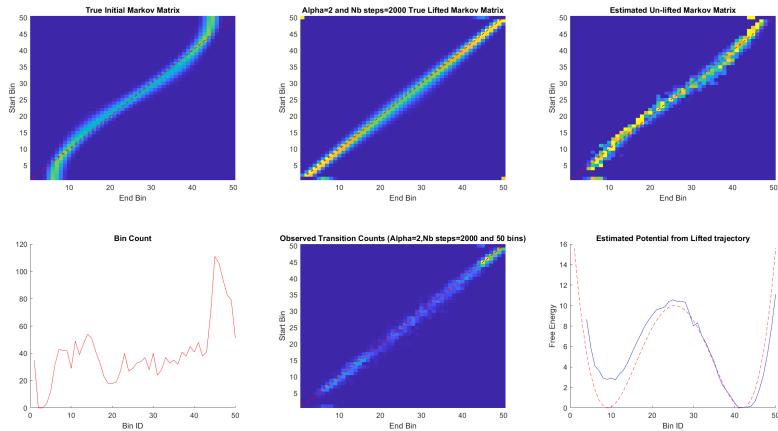


Figure 36: 50 Bins, 2000 Steps, Alpha = 2

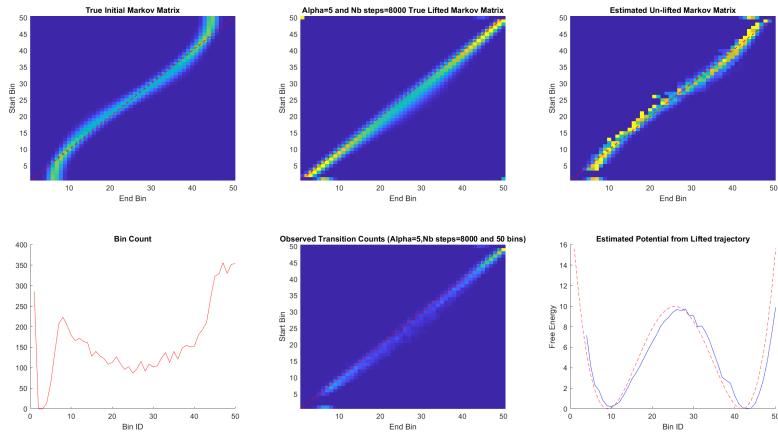


Figure 37: 50 Bins, 8000 Steps, Alpha = 5

### 3.6.5.2 Symmetric potential, 500.000 simulation steps, Alpha = 1,2, and 5

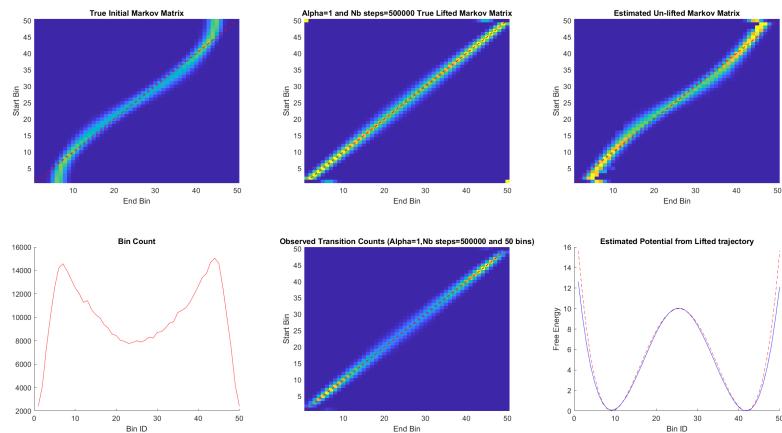


Figure 38: 50 Bins, 500000 Steps, Alpha = 1

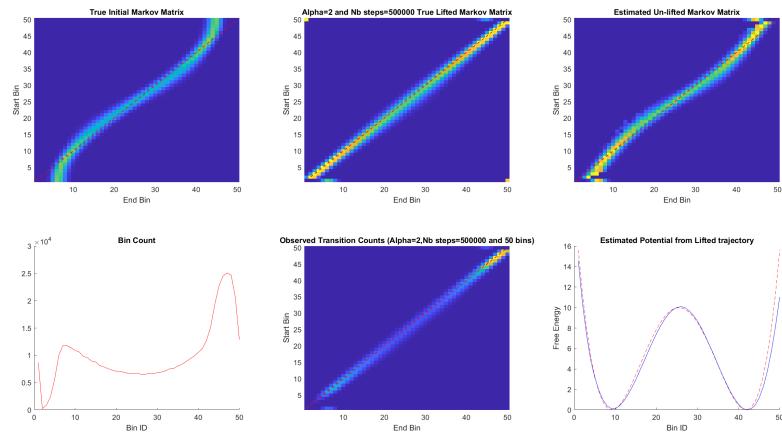


Figure 39: 50 Bins, 500000 Steps, Alpha = 2

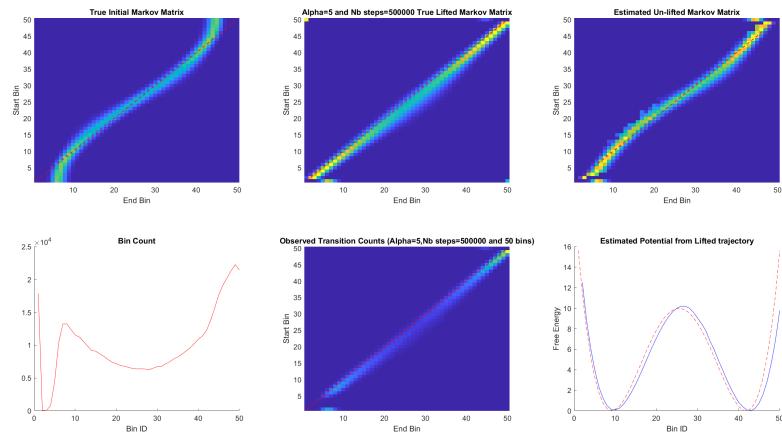


Figure 40: 50 Bins, 500000 Steps, Alpha = 5

### 3.6.6 Application to assymmetric 2-well potential

For this test we used the condition (A1) for both the flux and the periodic boundary, and an assymmetric 1-D potential.

#### 3.6.6.1 Assymmetric potential, 2.000 simulation steps, Alpha = 1,2, and 5

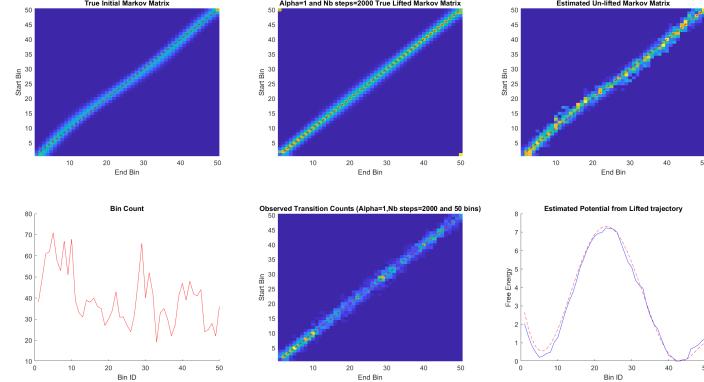


Figure 41: 50 Bins, 2000 Steps, Alpha = 1

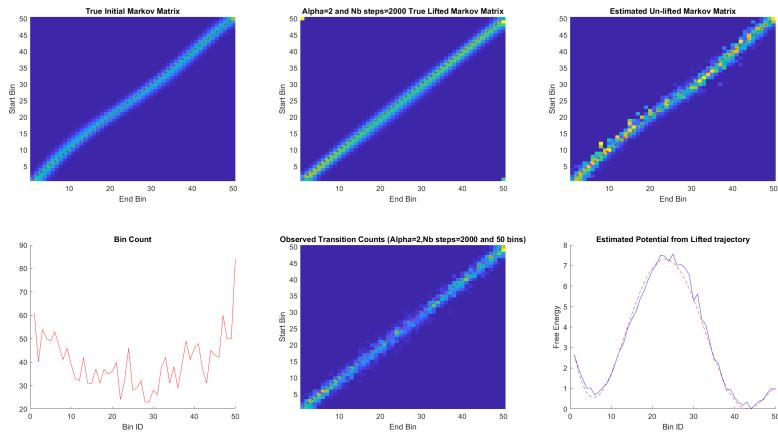


Figure 42: 50 Bins, 2000 Steps, Alpha = 2

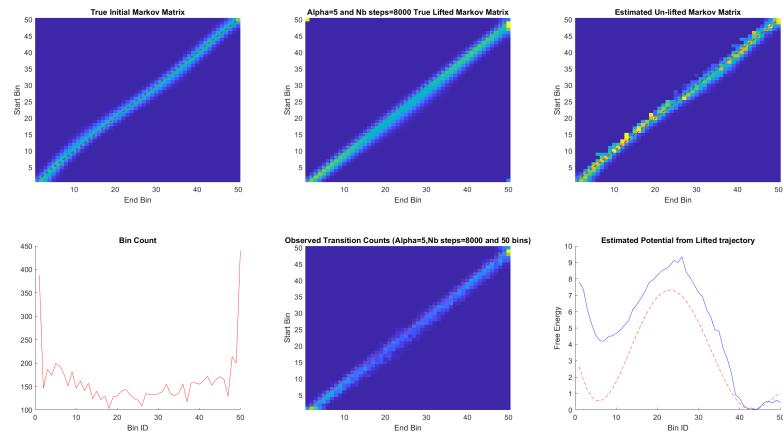


Figure 43: 50 Bins, 8000 Steps, Alpha = 5

### 3.6.6.2 Asymmetric potential, 500.000 simulation steps, Alpha = 1,2, and 5

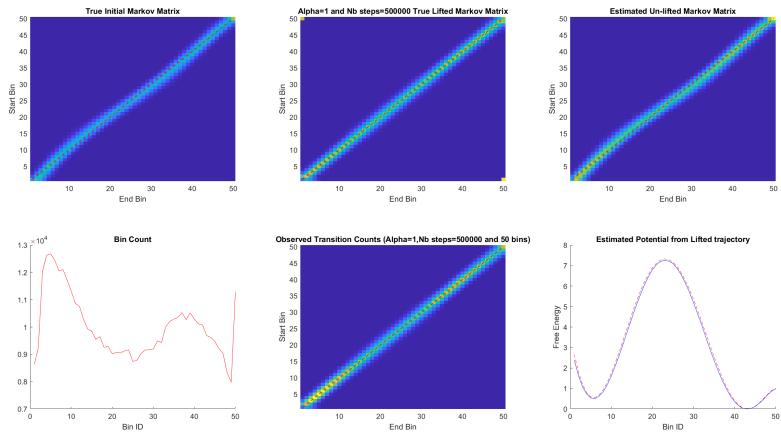


Figure 44: 50 Bins, 500000 Steps, Alpha = 1

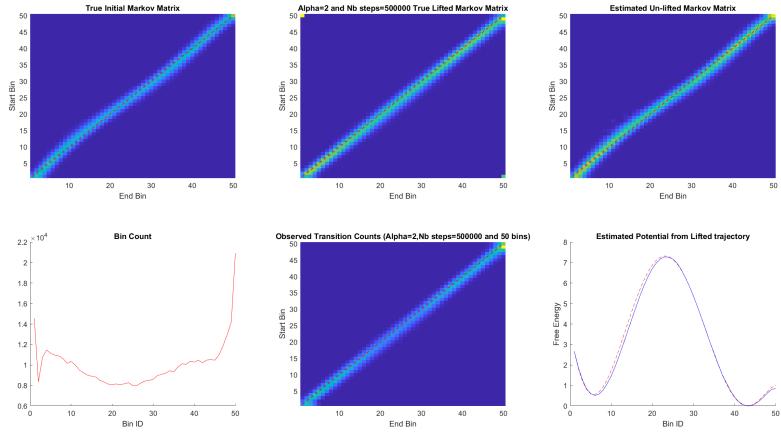


Figure 45: 50 Bins, 500000 Steps, Alpha = 2

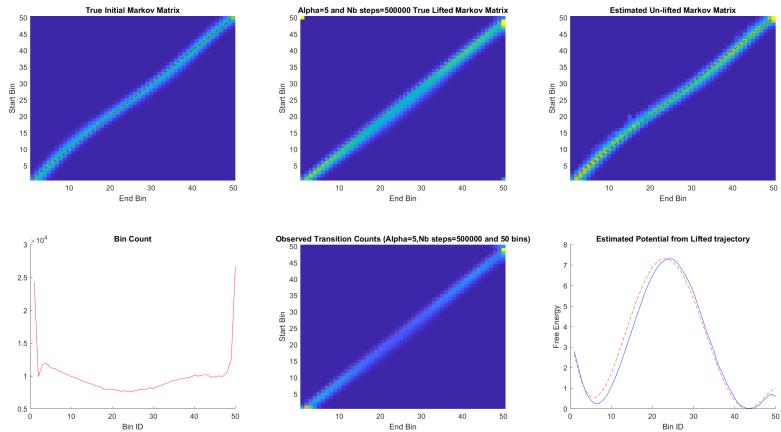


Figure 46: 50 Bins, 500000 Steps, Alpha = 5

### 3.6.7 Parallelize with another simulation lifted to opposite direction

To try and increase the performance of our method, we will perform two parallel simulations, each with opposite direction.

The strength of the flux is the same so the number of parameters (1) doesn't change.

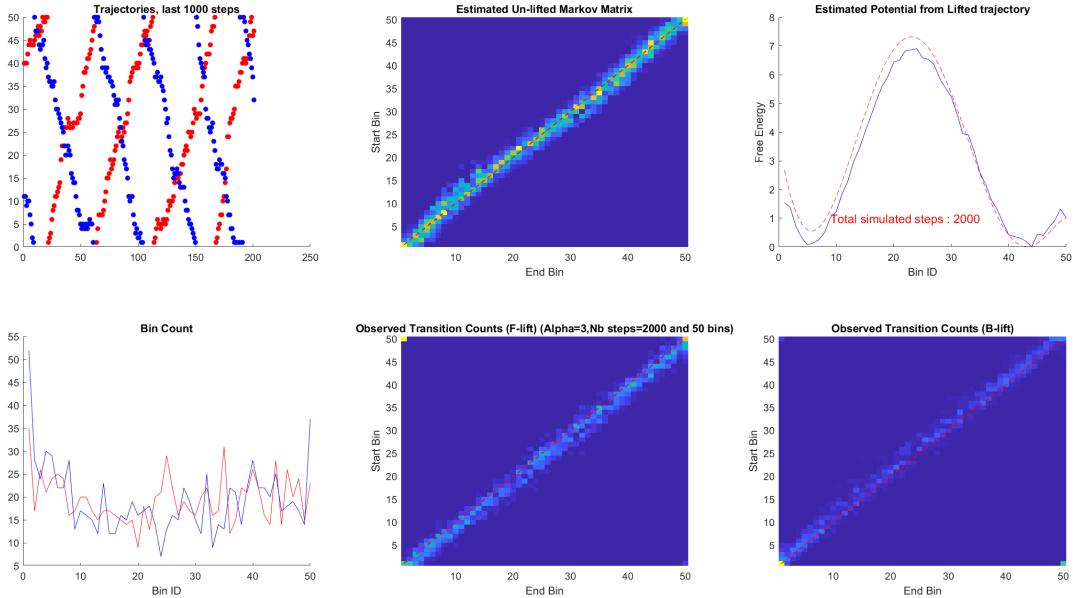


Figure 47: Single simulation 50 Bins, 2000 Steps, Alpha = 3

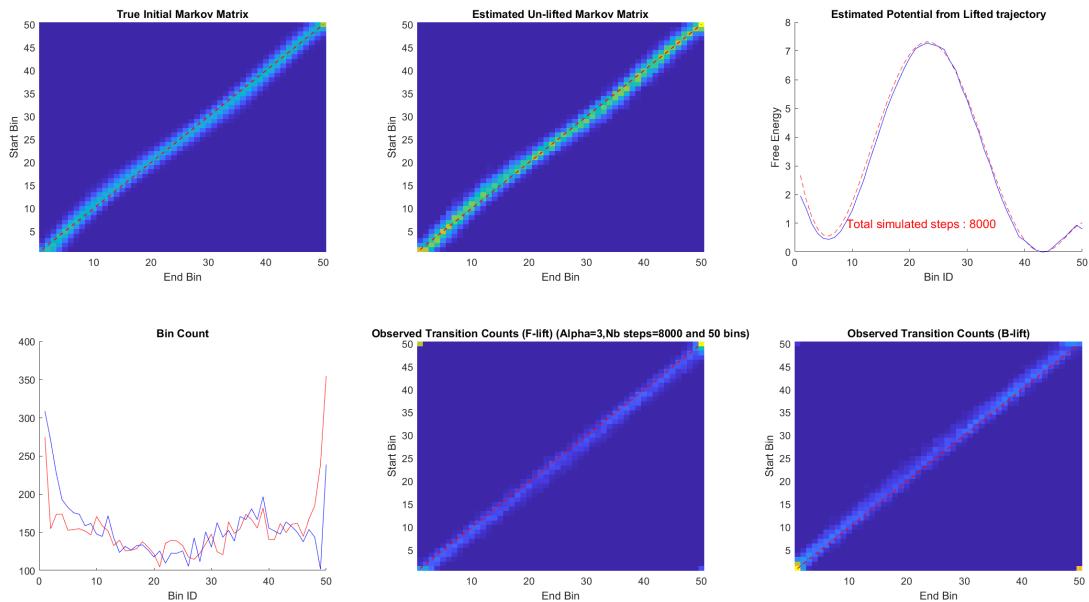


Figure 48: Single simulation 50 Bins, 8000 Steps, Alpha = 3

### 3.6.7.1 Multiple replicas and more States (100)

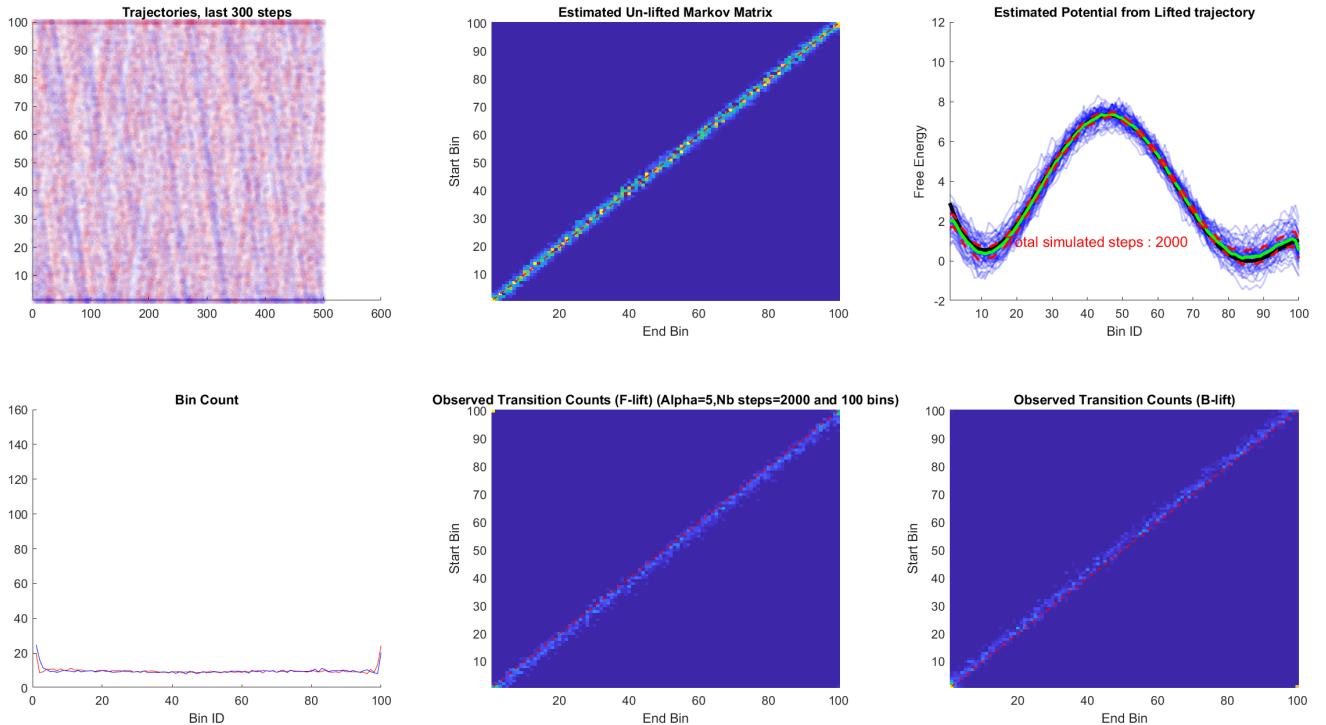


Figure 49: 100 Replica simulation 100 Bins, 2000 Steps, Alpha = 5

The top-right plot is the average estimate PMF (green) with errorbars of 1 variance (Red), and all the PMF estimates (blue transparent).  
The bottom-left is the average bin-count for Forward flux (Blue) and Backwards flux (Red).

All other plots are just for one example trajectory

### 3.6.7.2 Error Measures of estimation (precision and exploration)

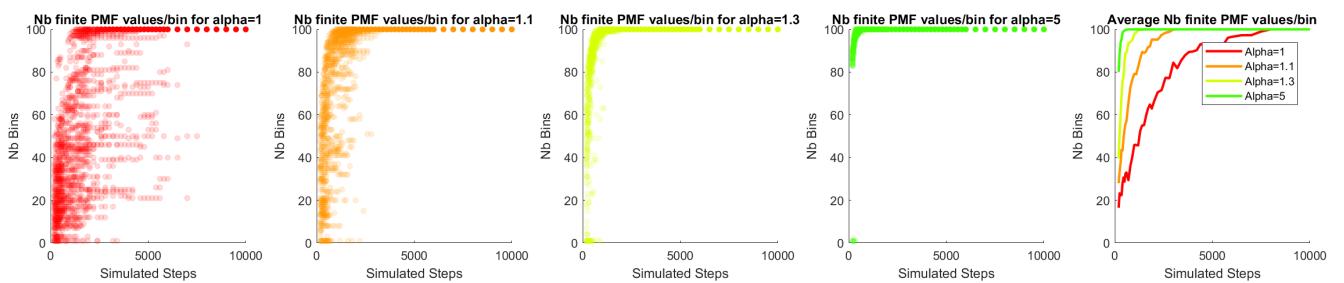


Figure 50: 100 Replica simulation 100 Bins, 2000 Steps, Alpha = 5

Increasing  $\alpha$  seems good for exploration, even  $\alpha = 1$  have a positive effect as it close to emulate a flat potential, it might be interesting to compare this to umbrella sampling and unbiased simulation.

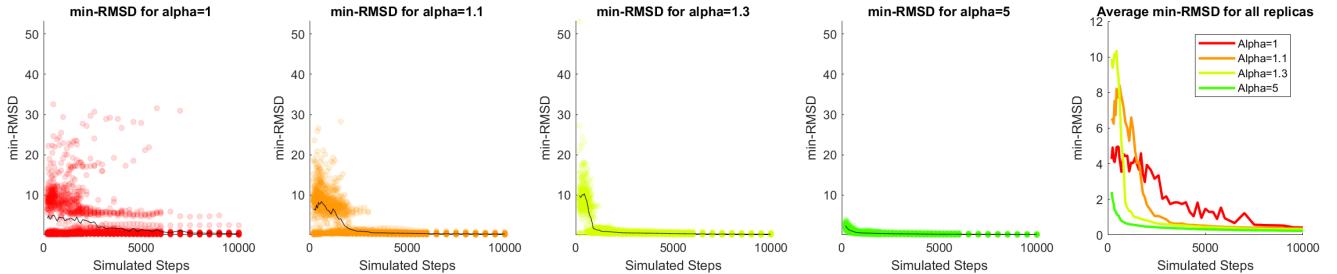


Figure 51: 100 Replica, 100 Bins

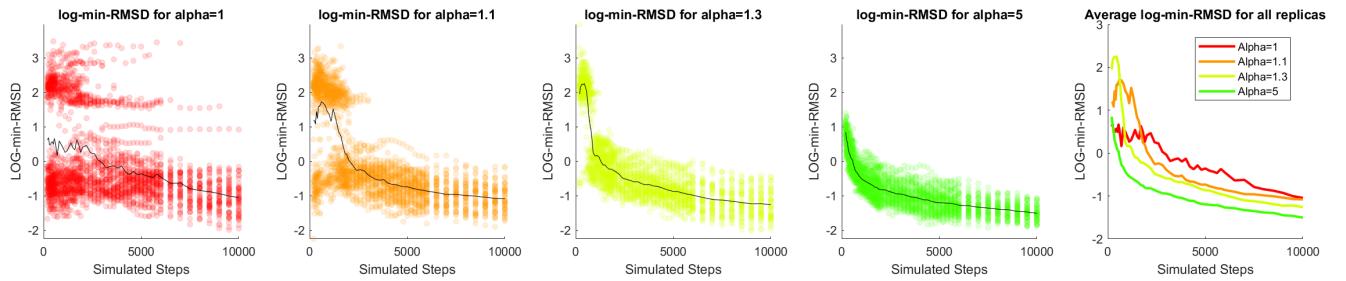


Figure 52: 100 Replica, 100 Bins

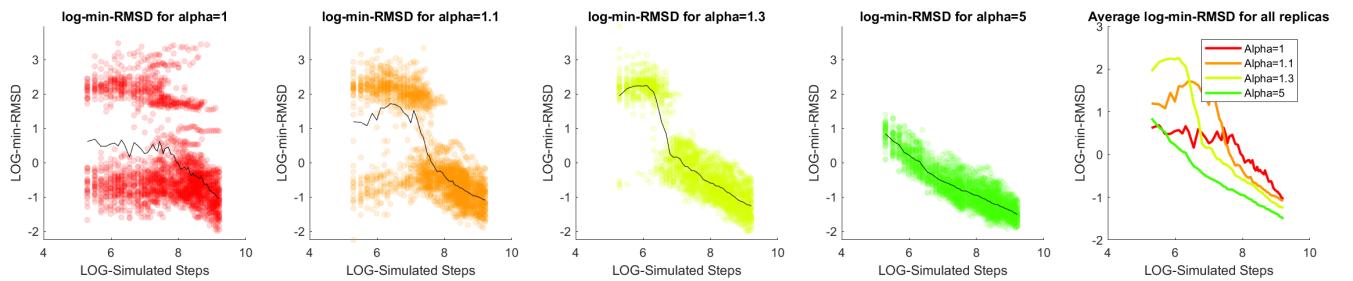


Figure 53: 50 Replica, 100 Bins

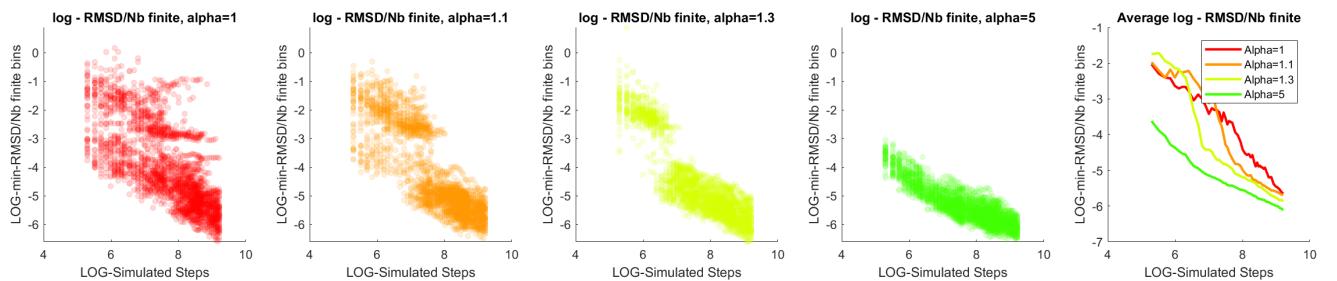


Figure 54: 50 Replica, 100 Bins

From this it seems like the long-term convergence of the estimation error follows a power law, with a power coefficient that is the same for any  $\alpha$ , but higher  $\alpha$  values allows to enter this mode of convergence sooner, probably because it greatly improves the exploration, without altering the estimation precision much.

### 3.6.8 Comparing to Umbrella Sampling and Unbiased Simulations

We run 50 replica trajectories and estimate a PMF for each to compute an estimation error.

#### 3.6.8.1 Umbrella Sampling

To compare our method to Umbrella Sampling, we used 50 uniformly positioned Umbrellas with force constant of 150.

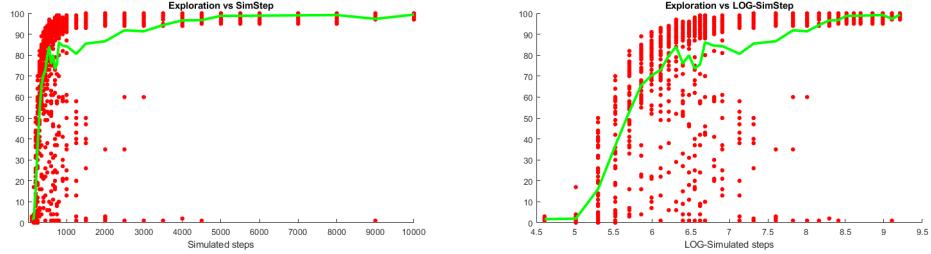


Figure 55: Umbrella Sampling simulations, exploration measure. (50 Replica, 100 Bins)

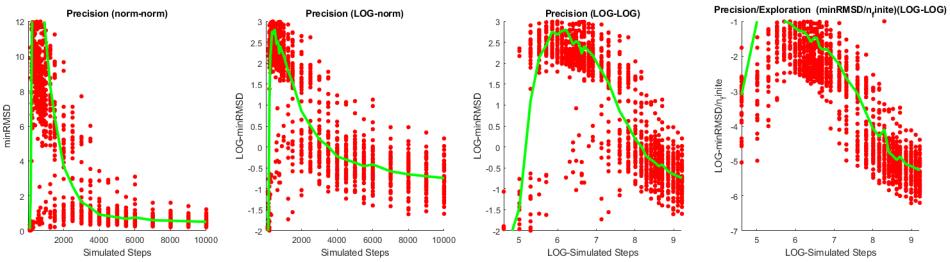


Figure 56: Umbrella Sampling simulations, estimation error measures. (50 Replica, 100 Bins)

The uniform Umbrella Sampling performs worse than our method, with an average log-minRMSD of about -0.6 after 10.000 Simulated Steps, whereas our algorithm reaches an average log-minRMSD of -1.5 after 10.000 Steps for  $\alpha = 5$ .

### 3.6.8.2 Non-biased Simulations

All replicas started in one of the wells ( $RC = 0.6$ ).

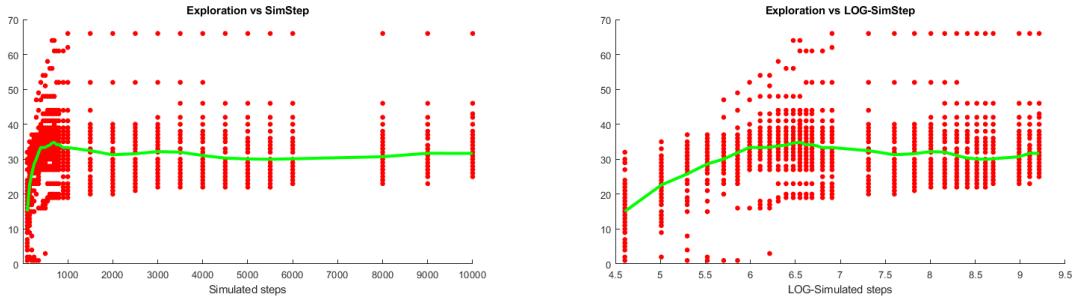


Figure 57: Non-biased simulations, exploration measure. (50 Replica, 100 Bins)

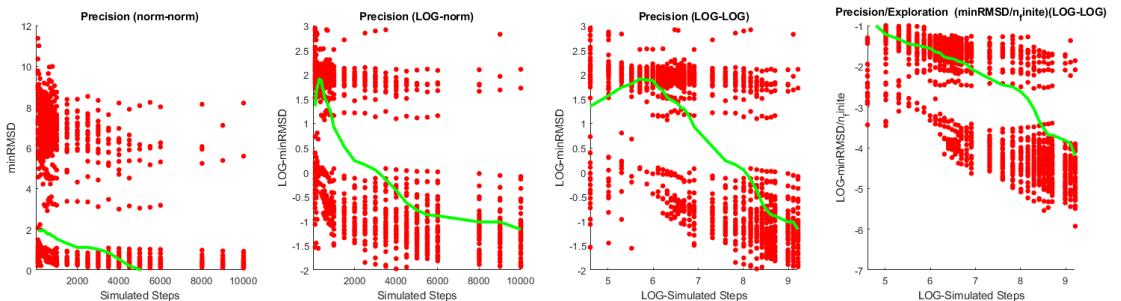


Figure 58: Non-biased simulations, estimation precision measure. (50 Replica, 100 Bins)

With Unbiased Simulations, the system never leaves the well within 10.000 Steps.

### 3.6.9 Conclusion

Inspiring from Markov Decision Processes and Lifting, we introduce a new biasing which breaks Detailed Balance by creating a net flux, in order to reduce the exploration and convergence time.

Using a 1-D potential as a test system, we show that we are indeed able unbias and estimate the right transition probabilities, which do follow Detailed Balance.

We find that running a parallel simulation with a flux in the opposite direction, and estimating the dynamics using both trajectories yields a better estimate than with only one simulation, for equal total number of samples. We compared the estimation error over simulation time of our algorithm compared to a simple Umbrella Sampling approach, and our algorithm have shown better results.

### 3.6.10 Additional Ideas

**Bias as Umbrella + Linear:** We could satisfy local flux conditions by still restaining the sub-system to a certain locality.

Use polyfit to get the right gradient at the borders, and consider not only the super borders, but also N-2, N-3, as long as the observed alpha starts to deviate from the defined one.

## 4 Clustering Problem

### 4.1 Introduction (Abstract)

MSMs of many biological and physical systems have typically a large dimensionality, which makes them prohibitively expensive to work with. Coarse graining methods have been introduced to reduce the dimensionality of these systems, capturing in particular their slowest kinetic processes [8, 19, 20, 61–63]. We address the definition of computationally efficient coarse-graining protocols, that I will call "reduction method", based on enforcing specific relations between MFPTs in the original and in the coarse-grained system . We will also consider various existing coarse-graining algorithms, introduce a new algorithm based on maximizing an arbitrary variational parameter inspired from Parallel Tempering,

### 4.2 Introduction (Technical)

Here we collate and describe a variety of results and pertaining to the Kinetic Network Clustering framework.

#### 4.2.1 Clustering in complex networks

When working with large or complex networks, one often encounters problems with visualisation and interpretation of observations due to the absence of a low dimensional space to project the graph onto. Often, it is helpful to convert a large network into a smaller one, while preserving the features of interest from the original network. This conversion is known as network clustering, [64] coarse-graining[65], or graph partitioning [66], and is equivalent to assigning each node of the graph to a community (cluster)[67].

The clustering of an  $N$ -node network into  $m$  clusters can be formally defined by a rectangular  $N \times m$  crisp assignment matrix  $\mathbf{S}$ , where each entry  $S_{iI} \in \{0, 1\}$  indicates whether (1) or not (0) node  $i \in \{1, \dots, N\}$  belongs to cluster  $I \in \{1, \dots, m\}$ , satisfying  $\sum_{I=1}^m S_{iI} = 1$ .

Choosing  $\mathbf{S}$  optimally usually entails maximizing or minimizing a certain objective function, which depends on the partitioning itself. It has been widely accepted that the objective function to use depends on the system modelled by the network and the exact task at hand, hence many metrics have been introduced to quantify clustering quality, [68, 69] which can be separated into two classes: topology-based and kinetics-based. Kinetic clustering uses kinetic properties of Markov processes and it is widely used to cluster kinetic networks, whereas topology-based clustering relies on structural properties of the network and it is more common with unweighted networks.

#### 4.2.1.1 Kinetic networks

Any finite Markov process can be modelled as a weighted directed network, also called kinetic network, where nodes represent the discrete states of the Markov process and edge weights represent the transition rates between different states. The state occupancy probabilities evolve according to the Master equation, which relates their rate of change to the difference in the probability flux in and out of the states

$$\frac{dp_j(t)}{dt} = \sum_{i(\neq j)} \left[ K_{ji} p_i(t) - K_{ij} p_j(t) \right] \quad (259)$$

where  $p_j(t)$  is the occupancy probability of state  $j$  and  $K_{ji}$  is the transition rate from state  $i$  to state  $j$ , with  $K_{jj} = -\sum_{i \neq j} K_{ij}$  being the rate of exits from state  $j$ . Inserting this definition into Eq. (259) one arrives at the matrix form

$$\frac{d\mathbf{p}}{dt} = \mathbf{K}\mathbf{p}. \quad (260)$$

All of the kinetic information of the system is encoded in the eigenvalues  $\lambda_n$  of the rate matrix  $\mathbf{K}$ , where  $n \in \{1, \dots, N\}$  for a system with  $N$  states, and their corresponding (right and left) eigenvectors.

If the Markov process is irreducible, one of the eigenvalues will be zero and the rest will all have a negative real part. We will focus on systems satisfying detailed balance, where the eigenvalues are guaranteed to be real and can be ordered as follows

$$0 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_N \quad (261)$$

From Eq. (260) it can be seen that the stationary (equilibrium) distribution  $\boldsymbol{\pi}$  is given by the right eigenvector corresponding to the zero eigenvalue  $\lambda_1$ . The other eigenvalues are related to the timescales  $\tau_n$  with which the rate matrix moves probability density between the oppositely signed regions of the corresponding eigenvector[70]

$$\tau_n = -\frac{1}{\lambda_n} \quad n \in \{2, \dots, N\}. \quad (262)$$

When the Markov process evolves at discrete times  $t = n\tau$  ( $n \in \mathbb{N}$ ), the evolution of the state occupancy probabilities is formulated in terms of the Markov chain

$$\mathbf{p}(n) = \mathbf{M}^n(\tau)\mathbf{p}(0) \quad (263)$$

where  $\mathbf{p}(n)$  is the probability at time step  $n$  and  $\mathbf{M}(\tau)$  is the matrix of transition probabilities between pairs of states over the lagtime  $\tau$ . Trivially Eq. (263) is equivalent to Eq. (260) when identifying  $\mathbf{M}(\tau) = e^{\mathbf{K}\tau}$ .

#### 4.2.1.2 Unweighted networks

Unweighted networks are defined by an adjacency matrix  $\mathbf{A}$ , where each entry  $A_{ij} \in \{0, 1\}$  determines whether (1) or not (0) an edge is present between nodes  $i$  and  $j$ , for  $i, j \in \{1, \dots, N\}$ . In general a network may be directed. For undirected networks the adjacency matrix is symmetric, i.e.,  $A_{ij} = A_{ji} \forall i, j$ .

It is possible to define a diffusion process on an unweighted network to transforms it into a kinetic network. This allows for the application of kinetic clustering methods. For diffusion processes on unweighted networks, the rate matrix  $\mathbf{K}$  is given by the random walk normalized Laplacian matrix  $\mathbf{L}$ [71] via  $\mathbf{K} \equiv -\mathbf{L}$ . The matrix  $\mathbf{L}$  is defined by

$$\mathbf{L} \equiv \mathbf{I} - \mathbf{A}^{-1} \quad (264)$$

where  $\mathbf{I}$  is the identity matrix and  $\Delta$  is a diagonal matrix with elements  $\Delta_{ij} = d_i \delta_{ij}$ , where  $d_i = \sum_{j=1}^N A_{ij}$  is the degree of node  $i$ .

Alternatively, one can define a discrete-time random walk on the network links, by means of the transition matrix

$$\mathbf{M}(\tau) = (\mathbf{A}^{-1})^\tau \quad (265)$$

for an arbitrary integer lagtime  $\tau$ .

Since the Laplacian (or, equivalently, the transition matrix given in Eq. (265)) is fully determined by the adjacency matrix, kinetic clustering based on diffusion or random walks is to make contact with topology-based clustering, relying purely on network structure.

#### 4.2.2 Modularity

A popular metric to assess the quality of a cluster assignment  $\mathbf{S}$  in unweighted networks, is the network modularity.[72, 73] For undirected networks, the modularity  $Q$  is defined as

$$Q(\mathbf{S}) = \frac{1}{N} \sum_{J=1}^m \sum_{i,j} S_{iJ} \left( A_{ij} - \frac{d_i d_j}{N} \right) S_{jJ} \quad (266)$$

where

$$= N^{-1} \sum_{i=1}^N d_i$$

is the average degree of the network.

The most widely used methods to cluster unweighted networks aim at finding the assignment matrix  $\mathbf{S}$  that maximizes the network modularity  $Q$ , and several greedy optimization algorithms have been developed to handle large networks with over 100 million nodes[74, 75].

However, the modularity suffers from a well-known resolution limit[76] which prevents it from resolving small communities, and it has been recently criticised as a metrics for information recovery in a network [77]. Moreover, modularity is only based on structural properties of networks, hence it does not take into account the kinetic process that a network may support. However, it has been shown to be equivalent to a kinetic clustering quality metric named *stability*[78], when applied to diffusion processes on unweighted networks.

### 4.2.3 Kinetic clustering

Kinetic clustering revolves around optimizing, over the cluster assignment  $\mathbf{S}$ , objective functions which are designed to enforce desired kinetic properties in the coarse-grained system. Usually they depend on the spectral properties of the Markov process (or Markov chain) defined in the original state space. A prominent example is the popular Perron cluster-cluster analysis (PCCA)[79], and its improved version PCCA+, introduced by Deuflhardt and Weber [80–83]. An alternative approach is to optimize objective functions that encode spectral properties of the *coarse-grained* dynamics [64, 84]. An advantage of this method is that certain spectral quantities of the coarse-grained dynamics satisfy a variational principle with respect to the original dynamics [85, 86], and thus provide a framework for performing a clustering that is variationally optimal.

Below, we review these methods and, building on the framework presented in Ref. [86], we propose a new clustering method based on the optimization of the Kemeny constant of the coarse-grained system.

#### 4.2.3.1 Coarse-grained Markovian processes

Coarse-graining a Markov process formally corresponds to projecting the Markovian dynamics onto a lower dimensional space, which generally introduces memory effects and consequently loss of the Markovian property. Hence a question arises as to what the best Markovian approximation is of the resulting non-Markovian dynamics.

For a given clustering  $\mathbf{S}$ , it has been shown that projections preserving detailed balance are those for which the Laplace transformed equilibrium correlation matrix

$$\hat{\mathbf{C}}(s) = \int_0^\infty \mathbf{C}(\tau) e^{-s\tau} d\tau \quad (267)$$

in the original and clustered systems satisfy the following relation [19]

$$\hat{C}_{JI}^{CG}(s) = \sum_{ij=1}^N \hat{C}_{ji}(s) S_{iI} S_{jJ}. \quad (268)$$

The entries of the correlation matrix  $\mathbf{C}(\tau)$  are the equilibrium connected correlation functions  $C_{ij}(\tau)$  of the state occupancy for every pair of states  $(i, j)$ , in the original Markovian system, defined as

$$C_{ji}(\tau) = M_{ji}(\tau) \pi_i - \pi_i \pi_j \quad (269)$$

and  $C_{JI}^{CG}(\tau)$  is the equilibrium connected correlation function of the cluster occupancy for the pair of clusters  $(I, J)$ , in the coarse-grained system.

The definition of an effective Markovian dynamics in the lower dimensional space requires approximations in which Eq. (268) is only satisfied in specific limits. Different definitions have been considered for an effectively Markovian coarse-grained dynamics, which correspond to different requirements on the correlation function of the coarse-grained system. These include the local equilibrium (LE), and the recently introduced Hummer-Szabo (HS)[62] method.

The local equilibrium (LE) reduction method consists in equating the correlation matrices at a specific finite value of  $\tau$

$$C_{JI}^{LE}(\tau) = \sum_{ij=1}^N C_{ji}(\tau) S_{iI} S_{jJ}. \quad (270)$$

Note that this means that the correlation matrices will depend on the lagtime  $\tau$  and that, depending how the correlation is used to obtain a clustering  $\mathbf{S}$ , the optimum result may depend on  $\tau$ . The equation reads, in matrix form,

$$\mathbf{C}^{LE}(\tau) = \mathbf{S}^T \mathbf{C}(\tau) \mathbf{S} \quad (271)$$

Hence, it corresponds to the  $s \rightarrow \infty$  limit of Eq. (268). Equation (271), by using Eq. (269) and a corresponding one for  $\mathbf{C}^{LE}(\tau)$ , provides a relation for the transition matrix  $\mathbf{M}^{LE}(\tau)$  of the coarse-grained dynamics

$$\mathbf{M}^{LE}(\tau) \mathbf{D}_\Pi = \mathbf{S}^T(\tau) \mathbf{D}_\pi \mathbf{S} \quad (272)$$

where  $\mathbf{D}_\pi$  and  $\mathbf{D}_\Pi$  are diagonal matrices with the stationary distributions of the original and coarse-grained systems on the diagonal, respectively. Thus,  $\mathbf{D}_\Pi = \mathbf{S}^T \mathbf{D}_\pi$ .

On the other hand, the Hummer-Szabo (HS) method equates the integral of the correlation matrices over all lagtimes  $\tau$  and thus it corresponds to the  $s \rightarrow 0$  limit of Eq. (268):

$$\int_0^\infty C_{JI}^{HS}(\tau) d\tau = \sum_{i \in I} \sum_{j \in J} \int_0^\infty C_{ji}(\tau) d\tau \quad (273)$$

In matrix form, we have

$$\hat{\mathbf{C}}^{HS} = \mathbf{S}^T \hat{\mathbf{C}} \mathbf{S} \quad (274)$$

where we have used the short-hand notation  $\hat{\mathbf{C}} = \hat{\mathbf{C}}(0)$  to denote the time-integrated connected correlation function.

Equation (274) provides a different definition of the coarse-grained dynamics with respect to Eq. (271), for the same clustering  $\mathbf{S}$ , and it leads to a formulation of the coarse-grained dynamics, that in continuous time is given in terms of the rate matrix [87]

$$\mathbf{K}^{HS} = \mathbf{S}^T \boldsymbol{\pi} \mathbf{1}_m^T - \mathbf{D}_\Pi (\mathbf{S}^T (\boldsymbol{\pi} \mathbf{1}_N - \mathbf{K})^{-1} \mathbf{D}_\pi)^{-1} \quad (275)$$

where  $\mathbf{1}_m$  is a vector of length  $m$  with entries equal to one. The HS method is known to provide numerically equivalent results to the LE method at long lagtimes, in many model systems[88].

In the present work we will use the HS definition of the coarse-grained dynamics, as it does not require the choice of a specific lag-time, it guarantees that MFPTs in the clustered dynamics match the weighted MFPTs of the microscopic dynamics [86] and it has been shown to replicate the dynamics of the original system more closely than the Local Equilibrium method [87].

Below, we review different methods to identify the cluster assignment  $\mathbf{S}$ .

#### 4.2.3.2 Stability

The approach proposed by Barahona et. al. [78, 89] to perform a kinetic clustering, is to maximize a quantity called stability, defined as the sum of the connected autocovariances of the clusters  $J$  at a specific lagtime  $\tau$

$$\Omega^S(\tau) \equiv \sum_{J=1}^m C_{JJ}^{LE}(\tau) = \text{Tr}(\mathbf{C}^{LE}(\tau)) \quad (276)$$

where  $\mathbf{C}^{LE}$  is calculated using the LE method as in Eq. (271). The authors have shown that, for a random walk on an unweighted non-directed network, such that  $\bar{d}$  is given in Eq. (265) and  $\pi_i = d_i/N\bar{d}$ , this quantity is equivalent to the network modularity when the time parameter  $\tau$  is set to one

$$\Omega^S(1) = \sum_{J=1}^m \sum_{i,j \in J} \left( \frac{A_{ji} d_i}{d_i N} - \frac{d_i d_j}{N N} \right) = Q \quad (277)$$

The authors of the method have proposed to optimize the quantity defined in Eq. (276) over the assignment matrix  $\mathbf{S}$  for varying values of  $\tau$ .

Since the optimal number of clusters decreases monotonically when increasing  $\tau$ , this approach introduces a dependence of the optimal clustering on the parameter  $\tau$ , which controls the resolution of the clustering. This allows for the identification of multiple clusterings, with finer or coarser structure, for a single network, but it increases the overall computational cost by requiring to optimize the clustering for each value of  $\tau$ , to identify the relevant number of clusters[78].

#### 4.2.3.3 Perron clustering

The Perron cluster-cluster analysis (PCCA) is a method for identifying communities or clusters in *nearly uncoupled* Markov chains [79]. It exploits the fact that in uncoupled Markov chains, characterized by a block-diagonal transition matrix, the entries of the left eigenvectors are constant on each cluster. This allows to regroup nodes following the sign structure of the left eigenvectors. The assumption is that this sign structure remains stable under small perturbations, i.e. the transition matrix remains block-diagonal dominant, so that a similar regrouping is possible in nearly uncoupled Markov chains.

The original method has been shown to suffer from lack of robustness, and the PCCA+ has been developed to use in practical applications [80]. The main difference from PCCA is the introduction of a soft (fuzzy)

assignment matrix  $\hat{\mathbf{S}}$ , where each node  $i$  is assigned a clustering vector of length  $m$  (number of clusters) satisfying the conditions of positivity and partition of unity

$$\hat{S}_{iI} \in [0, 1] \quad \forall i, I \quad (278)$$

$$\sum_{I=1}^m \hat{S}_{iI} = 1 \quad \forall i \quad (279)$$

hence interpretable as the probability of node  $i$  to belong to each cluster [83].

The main assumption of PCCA+ is that the assignment matrix  $\hat{\mathbf{S}}$  can be related to the eigenvectors of the transition matrix of the original system, by a transformation matrix  $\mathbf{T} \in \mathbb{R}^{m \times m}$  via

$$\hat{\mathbf{S}} = \mathbf{XT} \quad (280)$$

where  $\mathbf{X} = [\Psi_1^L, \dots, \Psi_m^L] \in \mathbb{R}^{n \times m}$  and  $\Psi_k^L$  is the left eigenvector of the transition matrix associated to the  $k$ -th largest eigenvalue, normalised in such a way that  ${}^T\mathbf{D}_\pi = \mathbf{I}$ . The clustering problem then consists in finding the matrix  $\mathbf{T}$  that optimizes an objective function under the constraints defined in Eq. (280), Eq. (278) and Eq. (279). Optimizing  $\mathbf{T}$ , rather than  $\mathbf{S}$ , leads to a reduction of the number of variables to be optimised from  $(n \times m)$  to  $(m^2)$ .

Weber originally proposed the objective function, here denoted as  $\Omega^{PCCA+W}$ ,

$$\Omega^{PCCA+W} \equiv \text{Tr} \left( \hat{\mathbf{S}}^T \mathbf{M}(\tau) \mathbf{D}_\pi \hat{\mathbf{S}} \mathbf{D}_\Pi^{-1} \right) \quad (281)$$

which can be written, for crisp assignment  $\mathbf{S}$ , as the trace of the coarse-grained transition matrix obtained using the LE method defined in Eq. (272)[81].

However, it was noted in Ref. [82] that the interpretation of  $\Omega^{PCCA+W}$  as the trace of a coarse-grained transition matrix is not valid for soft assignment matrices, so Roblitz proposed a new objective function [83]

$$\Omega^{PCCA+R} \equiv \text{Tr} \left( \hat{\mathbf{S}}^T \mathbf{D}_\pi \hat{\mathbf{S}} \mathbf{D}_\Pi^{-1} \right) = \text{Tr} \left( \mathbf{T}^T \mathbf{T} \mathbf{D}_\Pi^{-1} \right) \quad (282)$$

where the second equality uses Eq. (280) and normalization of the eigenvectors matrix . Given the property of  $\hat{\mathbf{S}}$  stated in Eq. (279), the matrix  $\hat{\mathbf{S}}^T \mathbf{D}_\pi \hat{\mathbf{S}} \mathbf{D}_\Pi^{-1}$  is stochastic, hence its trace is upper bounded by its dimension  $m$ , with the value  $m$  being attained by *any* crisp clustering  $\mathbf{S}$ . Thus, maximizing  $\Omega^{PCCA+R}$  is equivalent to making the clustering  $\hat{\mathbf{S}}$  as crisp as possible.

As the maximization of Eq. (282) over the entries of  $\mathbf{T}$  is subject to the constraints Eq. (280), Eq. (278) and Eq. (279), the optimal  $\mathbf{T}$  will depend on the eigenvectors of the transition matrix  $\mathbf{M}(\tau)$ , through . Since in a truly Markovian system the eigenvectors of the transition matrix are independent of the lagtime  $\tau$ , there is, in principle, no need to optimize over different values of  $\tau$ , as in the stability method. However, it has to be noted that molecular simulations often exhibit non-Markovian behavior, hence a dependence on the lagtime is expected in practice, when Markov matrices are constructed using simulation data.

In most applications, the PCCA+ objective function defined in Eq. (282) is used. In our implementation of the PCCA+ method, we made the same choice and used the efficient Schur decomposition to compute  $\mathbf{X}$ , following Weber et al.[90, 91].

#### 4.2.3.4 Slowest timescale

Recently, a variational kinetic clustering has been proposed in Ref. [17], that uses crisp assignment matrices and aims to maximize the second largest eigenvalue of the clustered dynamics. It was shown that this method is effective in identifying transition states, alongside to key metastable states.

This approach has been shown to be variationally optimal in Ref. [19], where it was proven that the slowest timescale of the clustered system is always smaller or equal to the one of the original network

$$\tau_2^{orig} \geq \tau_2^{CG} \quad (283)$$

regardless of the protocol (LE or HS) used to coarse-grain the dynamics.

However, the method has only been tested on systems that contain either a small number of metastable states, or a spectral gap between the slowest and second slowest timescales. The performance of this method on systems with several metastable states, where  $\tau_n \approx \tau_2$  for  $n > 2$ , remains to be investigated. The existence of multiple slow processes of similar timescale may require a variational parameter which incorporates multiple timescales. An interesting generalization of the slowest timescale would be considering an objective function that is the sum of the largest relaxation times

$$\Omega^{Sum(h)} \equiv \sum_{i=2}^{h+1} \tau_i^{CG} \quad (284)$$

where  $h$  is the number of timescales considered.

#### 4.2.3.5 Kemeny Constant

In Markovian systems the sum of all timescales is known as the Kemeny constant, which is also equivalent with the weighted sum of all mean first passage times (MFPTs)  $t_{ji}$  from a selected state  $i$  to all other states  $j$ , where the weights are the equilibrium populations  $\pi_j$  of the target states  $j$ . [21, 23, 86]

$$\zeta^{orig} \equiv \sum_{n=2}^N \tau_n = \sum_{j=1}^N \pi_j t_{ji} \quad (285)$$

Remarkably, the result is independent of the choice of the initial state  $i$ . [92].

Since the MFPTs in a Markovian system can be expressed in terms of time-integrated correlation functions[86], the Kemeny constant can be expressed as the trace of a matrix called "deviation matrix" or "fundamental matrix", which is related to the time-integrated correlation matrix: [93]

$$\zeta^{orig} = \text{Tr}(\hat{\mathbf{C}}\mathbf{D}_\pi^{-1}). \quad (286)$$

The Kemeny constant of the coarse-grained system can be written similarly, and it will depend on the protocol used to coarse-grain the dynamics. For the HS method,

$$\zeta^{HS} = \text{Tr}(\mathbf{S}^T \hat{\mathbf{C}} \mathbf{S} \mathbf{D}_\Pi^{-1}). \quad (287)$$

In Ref. [86], it was shown that the Kemeny constant of the system clustered according to the HS method is bounded by the value of the Kemeny constant in the original system. In particular, the Kemeny constant of the initial and clustered systems are related by a transparent relation, when the HS reduction method is applied[86]

$$\zeta^{HS} = \zeta^{orig} - \sum_{J=1}^m \frac{1}{\Pi_J} \sum_{j \in J, i \in J} \pi_j t_{ji} \pi_i \quad (288)$$

The second term on the right-hand side represents the expectation value of the MFPTs if two states are drawn from within the same cluster with their equilibrium probabilities. This term vanishes when each cluster consists of only one node (i.e. no clustering is performed) and, for a fixed number of clusters, it becomes smaller as the clusters become increasingly metastable, *i.e.*, intra-cluster dynamics is fast as compared to all time scales. The relation above shows that maximizing the Kemeny constant of the clustered system, for a fixed number of clusters, leads to a variationally optimal partitioning, aimed at identifying the clusters with fastest intra-cluster (and the slowest inter-cluster) dynamics.

In conclusion, the Kemeny constant appears to be an objective function that quantifies the metastability of the coarse-grained system, similarly to the modularity, stability, and PCCA+W, however, it has a few advantages when compared to the other measures. One advantage is that it accounts for the information about the system dynamics at all lagtimes, via the integrated correlation function  $\hat{\mathbf{C}}$ , so that it does not rely on the choice of a particular lagtime  $\tau$ . In addition, it is variationally optimal and it has a simple interpretations in terms of other kinetic quantities, such as the system timescales and the mean first passage times. Such relations allow for the derivation of explicit formulae for the optimal position of the cluster boundaries that maximize the Kemeny constant in simple systems, e.g., diffusive processes on 1D potential, as shown in Sec. 4.3.5. These features make the Kemeny constant an attractive quantity for the detection of metastable clusters in both, kinetic and unweighted networks.

### 4.3 RESULT 1 (Published) THEORY - Timescales as variational parameter for kinetic network clustering

Efficiently identifying the most important communities and key transition nodes in weighted and unweighted networks is a prevalent problem in a wide range of disciplines.

Here we focus on the optimal clustering using variational kinetic parameters, linked to Markov processes defined on the underlying networks, namely the Mean First Passage Times, the slowest relaxation time and the Kemeny constant. We provide a protocol to reduce the dimensionality of kinetic networks, based on specific requirements that the MFPTs in the coarse-grained system should satisfy. We prove that this protocol coincides with the one proposed by Hummer and Szabo in [62] and it leads to a variational principle for the Kemeny constant. Finally, we introduce a modification of this protocol which preserves the Kemeny constant. We derive novel relations in terms of mean first passage times for optimizing clustering via the Kemeny constant, and show that the optimal clustering boundaries have equal round-trip times to the clusters they separate.

### 4.3.1 Introduction

MSM of many biological and physical systems have typically a large dimensionality, which makes them prohibitively expensive to work with. Coarse graining methods have been introduced to reduce the dimensionality of these systems, capturing in particular their slowest kinetic processes [8, 19, 20, 61–63]. In this work we address the definition of computationally efficient coarse-graining protocols, based on enforcing specific relations between MFPTs in the original and in the coarse-grained system.

A quantity which has attracted a large interest, over the years, since its introduction in 1960 by Kemeny and Snell [27], is the so-called Kemeny constant, which represents the sum of relaxation timescales in a kinetic network or Markov chain. Remarkably, the Kemeny constant is also equivalent with the weighted sum of all mean first passage times (MFPTs) from a selected state  $i$  to all other states  $j$ , where the weights are the equilibrium populations of states  $j$ . Surprisingly, the Kemeny constant is independent on the starting state  $i$ . This intriguing constancy has been the subject of several studies [28, 40]. The Kemeny constant has also attracted considerable interest in the field of graph theory and networks science. In particular, it has been used to calculate the Kirchoff index of a graph [94] and it has been proposed as an objective function to optimize in graph clustering algorithms [95]. In the context of graphs, a low Kemeny constant means, loosely speaking, that the time to travel between the nodes is on average small, so this is interpreted to mean that the graph is well-connected [96].

Taking advantage of the relations between the MFPT and transition rates derived in Sec. (2), we propose a protocol to reduce the dimensionality of kinetic networks, based on the requirement that a certain relation between the MFPTs of the original and the coarse-grained system is satisfied. We show that this protocol coincides with the coarse-graining proposed recently by Hummer and Szabo in [62], and it leads to a variational principle for the Kemeny constant, which can be useful to optimise the dynamical coarse-graining of kinetic networks. Finally, we show that by suitably modifying this protocol, one can define a coarse-graining which ensures that the Kemeny constant is preserved.

In addition to clusters, the notion of transition states is of particular importance in the context of MSMs: these are bottleneck states which the system passes through while moving between the metastable clusters. A related notion in network science is the one of node centrality. Different measures of node centrality have been introduced to characterise nodes linking between two or more clusters, e.g., betweenness[97] and closeness[98] centrality, however these are based on topological features of the network and disregard kinetics information, hence their use to automatically identifying transition states in kinetic and, more in general, weighted networks, remains unclear. In earlier work, it was demonstrated that the slowest timescale in kinetic networks can be used as a variational parameter for finding transition states effectively[17, 19], however the algorithm developed in Ref. [17] was inefficient when scaled to high-dimensional systems.

In this work, building on the framework developed in Refs. [17, 86], we propose a new clustering method based on the optimization of the Kemeny constant of the coarse-grained system, which satisfies a variational principle with respect to the original dynamics and is effective in finding metastable clusters. We derive properties of the optimal boundary positions in terms of mean first passage times. We find that in the optimal clustering of large complex networks each node belongs to its nearest cluster measured via round-trip times distance.

### 4.3.2 Coarse Graining of Rate Matrices based on MFPTs

The unified framework set up above is deeply useful for investigating new relations and interpreting the results physically. As an example we use this framework to derive a coarse graining protocol which preserves the MFPTs of the system. Coarse graining involves projecting a high dimensional dynamics on to some coarse lower dimensional space. This involves grouping together microstates (labeled by lower case indices  $i, j$ ) in to macrostates (labeled by upper case indices  $I, J$ ). In what follows, we will denote with  $\hat{p}_I(t)$  the occupation probability of the macrostates  $I = 1, \dots, N$ , with  $N < n$ . Clearly, this must be equal to the sum of the probabilities of all microstates  $i$  in the macrostate  $I$ , i.e.  $\hat{p}_I(t) = \sum_{i \in I} p_i(t)$ . The sum can be encoded in to an  $n \times N$  aggregation matrix  $\mathbf{A}$ , with elements  $A_{iI} = 1$  if  $i \in I$  and zero otherwise, which defines the clustering. Hence,  $\hat{\mathbf{p}}(t) = \mathbf{A}^T \mathbf{p}(t)$ .

There has been much recent research in to how best to perform a kinetic coarse graining [17, 19, 62], in particular on how to optimally define a coarse-grained rate matrix, for a given choice of the clustering  $\mathbf{A}$ , and how to optimally choose the latter. In [17, 19] it has been shown that meaningful grouping of states can be achieved by matching the relaxation time of the original rate matrix and the coarse-grained rate matrix proposed in [62], for which the second largest eigenvalue satisfies a variational principle. Here, we focus on the question of how to optimally choose a coarse-grained rate matrix for a given clustering  $\mathbf{A}$  of states. We propose that the link between MFPTs and rate matrices is used to define a coarse grained rate matrix which enforces a particular condition on the MFPTs.

In the previous sections we have shown that MFPTs fully determine the equilibrium and kinetic properties of a system. Hence, given the  $N \times N$  matrix  $\hat{\mathbf{t}}$  of MFPTs between the macrostates, the coarse-grained  $N \times N$

rate matrix  $\hat{\mathbf{K}}$  follows from (64) as

$$\hat{\mathbf{K}} = \hat{\mathbf{t}}^{-1}(\mathbf{D}_N^{-1} - \mathbf{1}_N \mathbf{1}_N^T), \quad (289)$$

where  $\mathbf{D}_N$  is the  $N \times N$  diagonal matrix with  $\mathbf{P}^{\text{eq}}$  along its diagonal and  $\mathbf{1}_N$  is the  $N$ -dimensional vector with all the entries equal to 1. A question that immediately arises, however, is: how should the MFPTs in the coarse grained system be defined or measured?

A minimal condition is that the coarse grained dynamics converges to the equilibrium distribution  $\hat{\mathbf{p}}^{\text{eq}} = \mathbf{A}^T \mathbf{p}^{\text{eq}}$ . Using (63), this results in the condition

$$\hat{\mathbf{p}}^{\text{eq}} = \frac{\mathbf{1}_N^T \hat{\mathbf{t}}^{-1}}{\mathbf{1}_N^T \hat{\mathbf{t}}^{-1} \mathbf{1}_N} \quad (290)$$

A second condition is that  $\hat{t}_{II} = 0 \forall I$ . In Appendix (C.1), we show that there is a whole family of vectors  $\hat{\mathbf{t}}$  which satisfies these requirements, so further conditions have to be imposed.

One possibility is to require that if we choose two (different) macrostates with equilibrium probability, then the MFPT between them is the same as if we choose two microstates from within the macrostates with equilibrium probability, i.e.

$$\hat{t}_{JI} = \frac{1}{\hat{p}_I^{\text{eq}} \hat{p}_J^{\text{eq}}} \sum_{i \in I, j \in J} p_j^{\text{eq}} p_i^{\text{eq}} t_{ji} - \frac{1}{(\hat{p}_J^{\text{eq}})^2} \sum_{i, j \in J} p_j^{\text{eq}} p_i^{\text{eq}} t_{ji}, \quad (291)$$

where the second term on the r.h.s. removes the contribution from microstates belonging to the same macrostate and ensures that  $\hat{t}_{II} = 0 \forall I$ . Obviously, we could have subtracted the contribution from  $I = J$  in a number of different ways, e.g.

$$\hat{t}_{JI} \hat{p}_I^{\text{eq}} \hat{p}_J^{\text{eq}} = (1 - \delta_{IJ}) \sum_{i \in I} \sum_{j \in J} p_i^{\text{eq}} p_j^{\text{eq}} t_{ji}, \quad (292)$$

however, one can show that this latter choice does not lead to the right equilibrium distribution (see Fig. (59)).

Below we derive analytically the rate matrix  $\hat{\mathbf{K}}$  resulting from the MFPTs choice (291), and show that it leads to the steady state  $\hat{\mathbf{p}}^{\text{eq}}$ . Rewriting the summation on the r.h.s. of (291) in terms of  $\mathbf{A}$ , we have

$$\hat{t}_{JI} = (\mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JI} - z_J \quad (293)$$

where we have defined  $z_J = (\mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JJ}$ . This translates to the matrix relation

$$\hat{\mathbf{t}} = \mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} - \mathbf{z} \mathbf{1}_N^T. \quad (294)$$

Multiplying times  $\hat{\mathbf{K}}$  from right, using  $\mathbf{1}_N^T \hat{\mathbf{K}} = 0$  and (289), we obtain

$$\hat{\mathbf{K}} = (\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})^{-1} (\mathbf{I} - \hat{\mathbf{p}}^{\text{eq}} \mathbf{1}_N^T) \quad (295)$$

which clearly satisfies  $\hat{\mathbf{K}} \hat{\mathbf{p}}^{\text{eq}} = 0$ . An alternative expression, which is useful for later,

$$\begin{aligned} \mathbf{R} &= \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - [\mathbf{P}^{\text{eq}} \mathbf{1}_N^T + \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{p}^{\text{eq}} \mathbf{1}_N^T \\ &\quad - \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}]^{-1}, \end{aligned} \quad (296)$$

is derived in appendix (C.2).

In Appendix (C.3) we show that this coarse-graining protocol coincides with the one proposed by Hummer and Szabo in [62]. An advantage of using (295) to calculate the rate matrix of the coarse-grained system, when compared to its formulation in [62], is that it only requires the inversion of a matrix with low dimensionality  $N < n$ , and is thus computationally highly efficient, provided the MFPTs and the equilibrium distribution of the original system are known.

In Section (4.3.3), we show that a variational principle on the Kemeny constant exists for this coarse-graining protocol, namely the Kemeny constant of the coarse-grained system is bounded from above by the Kemeny constant of the original system.

As we have discussed above, there are other valid choices one could make for the coarse-grained MFPTs, which lead to alternative definitions of the coarse-grained rate matrices. In Appendix (C.1), we derive an alternative choice, which ensures that the Kemeny constant of the original and the clustered systems are identical, in contrast to (295) and [62]:

$$\hat{\mathbf{t}} = \frac{1}{1 - \frac{1}{\zeta} [\hat{\mathbf{p}}^{\text{eq}}]^T \mathbf{z}} [\mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} - \mathbf{z} \mathbf{1}_N^T], \quad (297)$$

where  $\zeta$  is the Kemeny constant of the original system.

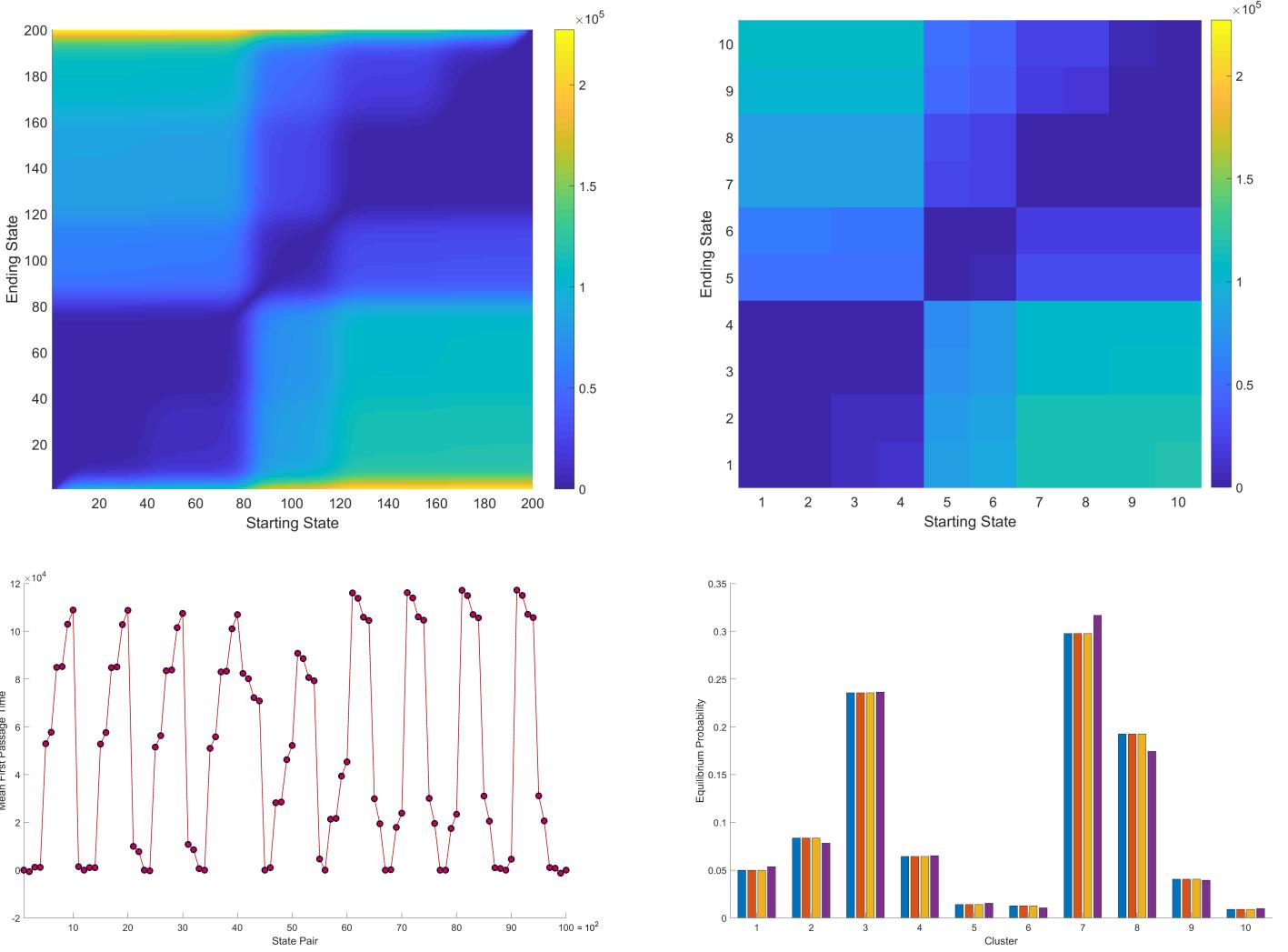


Figure 59: MFPTs and equilibrium distribution of the system diffusing on the potential  $v(x) = \sin((x - \pi)/2) + \sin((x - \pi)/5)$ , coarse-grained from initially 200 into 10 bins, equally sized, in the interval  $[-6\pi, 14\pi]$ . Top left: MFPTs  $t_{ij}$  of the original system, as a function of  $i, j \in \{1, \dots, 200\}$ . Top right: coarse-grained MFPTs  $\hat{t}_{IJ}$ , as given in (291), as a function of the macrostates  $I, J \in \{1, \dots, N\}$ . Bottom left: comparison between the coarse-grained MFPTs  $\hat{\mathbf{t}}$ , computed via formula (291) from the MFPTs  $\mathbf{t}$  of the original system (green line), and those computed via formula (48), from the coarse-grained rate matrices, given by the two equivalent expressions (295) (blue line) and (296) (pink line). For simplicity,  $\hat{t}_{IJ}$  is plotted here as a single index quantity  $\hat{t}_{(J-1)N+I}$ , versus its index. Bottom right: targeted equilibrium distribution  $\mathbf{A}^T \mathbf{p}^{\text{eq}}$  (blue), and equilibrium distribution  $\hat{\mathbf{p}}^{\text{eq}}$  resulting from (290) with  $\hat{\mathbf{t}}$  defined in (291) (orange) and (292) (purple), respectively. The former leads to the targeted distribution, while the latter does not. The yellow columns show results for a new protocol, defined by equation (297).

We test the above relations on the system defined in (315), diffusing on the one-dimensional potential plotted in Fig. (1). In the top panels of Figure (59) MFPTs are shown as heat maps for the original and the clustered system. In the bottom left panel, we plot the MFPTs computed via formula (48), for the rate matrices (295) and (296). As expected, these are identical, and coincide with the MFPTs calculated via formula (291). In the bottom right panel, we plot the equilibrium distribution resulting from (290) with  $\hat{\mathbf{t}}$  defined in (291). As expected, this is identical to the targeted distribution  $\mathbf{A}^T \mathbf{p}^{\text{eq}}$ . For comparison, we also plot the distribution that would have resulted from the choice (292), which deviates from the targeted distribution  $\mathbf{A}^T \mathbf{p}^{\text{eq}}$ , as anticipated. Finally, we show the equilibrium distribution for the alternative clustering protocol given in (297), which preserves the Kemeny constant. This is seen to match the targeted distribution, in addition, it is verified numerically that it leads to the same Kemeny constant as the original system.

#### 4.3.3 Variational principle for Kemeny Constant in Hummer-Szabo Coarse Graining

In [19] we have shown that a variational principle holds for the second largest eigenvalue of the rate matrix in the system coarse-grained according to the Hummer-Szabo prescription, namely its inverse (corresponding to the relaxation time in the coarse-grained system) is smaller than or equal to the inverse second largest eigenvalue of the rate matrix of the original system (giving the relaxation time of the original system). This variational principle has been used in [17] to identify optimal clustering protocols. In this section we show that a similar variational principle holds for the Kemeny constant itself, for the coarse-graining protocol based on (291), that we prove to be equivalent to the one proposed by Hummer-Szabo, in appendix (C.3). Summing (291) over  $J$  and rewriting  $\sum_J \sum_{j \in J} = \sum_j$

$$\sum_j \sum_{i \in I} p_j^{\text{eq}} p_i^{\text{eq}} t_{ji} - \sum_J \frac{\hat{p}_J^{\text{eq}}}{\hat{p}_J^{\text{eq}}} \sum_{i,j \in J} p_j^{\text{eq}} p_i^{\text{eq}} t_{ji} = \sum_J \hat{p}_J^{\text{eq}} \hat{p}_I^{\text{eq}} \hat{t}_{JI} \quad (298)$$

and finally using (35) we obtain

$$\zeta = \sum_J \frac{1}{\hat{p}_J^{\text{eq}}} \sum_{i,j \in J} p_j^{\text{eq}} p_i^{\text{eq}} t_{ji} + \zeta^{\text{CG}} \quad (299)$$

where  $\zeta^{\text{CG}}$  is the Kemeny constant in the coarse-grained system. Since the first term on the RHS of the equation above is non-negative, we have

$$\zeta^{\text{CG}} \leq \zeta. \quad (300)$$

This extends the variational principle previously found for the relaxation time, to the sum of all the timescales in the system. We intend to explore in future work variational clusterings based on Kemeny constants.

#### 4.3.4 Coarse Graining of Transition Matrices based on MFPTs

Similarly, a relation for the coarse-grained transition matrix  $\hat{\mathbf{Q}}$ , in discrete time dynamics, would follow from (70) as

$$\hat{\mathbf{Q}} = (\mathbf{I} - \mathbf{D}_N \hat{\mathbf{t}}')^{-1} (\mathbf{p}^{\text{eq}} \mathbf{1}_N^T - \mathbf{D}_N \hat{\mathbf{t}}'), \quad (301)$$

where  $\hat{\mathbf{t}}'$  is the matrix of discrete-time MFPTs in the coarse-grained system. A definition of the latter that would seem physically meaningful would be

$$\hat{t}'_{JI} \hat{p}_I^{\text{eq}} = \sum_{i \in I} p_i^{\text{eq}} \bar{t}'_{Ji} \quad (302)$$

where  $\bar{t}'_{Ji}$  denotes the MFPT from a microstate  $i$  to a macrostate  $J$ , which can be calculated via a recursive equation, analogous to (18),

$$\bar{t}'_{Ji} = \sum_{j \in J} Q_{ji} + \sum_{k \notin J} (\bar{t}'_{jk} + 1) Q_{ki} = 1 + \sum_{k \notin J} \bar{t}'_{jk} Q_{ki}. \quad (303)$$

This is rewritten more conveniently by defining  $M_{ki}^{(J)} = Q_{ki}$  for  $k \notin J$  and  $M_{ki}^{(J)} = 0$  for  $k \in J$ ,

$$\sum_k \bar{t}'_{jk} (\delta_{ki} - M_{ki}^{(J)}) = 1 \quad (304)$$

which allows to compute each row of the rectangular  $N \times n$  matrix  $\bar{\mathbf{t}}'$ ,  $[\bar{\mathbf{t}}'_J]^T = (\bar{t}'_{J1}, \dots, \bar{t}'_{Jn})$ , simply by inverting matrix  $\mathbf{B}^{(J)}$ , with elements  $B_{ki}^{(J)} = \delta_{ki} - M_{ki}^{(J)}$

$$[\bar{\mathbf{t}}'_J]^T = \mathbf{1}_n^T [\mathbf{B}^{(J)}]^{-1} \quad (305)$$

and to finally compute  $\hat{\mathbf{t}}'$  using (302).

Intriguingly, however, the equilibrium distribution computed from (68) for this natural choice of MFPTs, does not lead to the right equilibrium distribution  $\mathbf{A}^T \mathbf{p}^{\text{eq}}$ , for arbitrary choices of  $\mathbf{A}$ . Note that, nevertheless, the diagonal terms of the MFPTs matrix so defined, trivially retrieve the equilibrium probabilities: upon multiplying equation (303) times  $p_i^{\text{eq}}$ , summing over  $i$  and using  $\sum_i Q_{ki} p_i^{\text{eq}} = p_k^{\text{eq}}$ , we obtain the relation

$$\sum_{k \in J} \bar{t}'_{jk} p_k^{\text{eq}} = 1, \quad (306)$$

which combined with (302), leads to  $\hat{t}'_{II} = 1/\hat{p}_I^{\text{eq}} \forall I$ . Given this surprising outcome, the definition of physically meaningful coarse-grained MFPTs in discrete-time dynamics remains an interesting pathway for future work.

#### 4.3.5 Analytic maximization of the Kemeny constant

In this work we propose to use the Kemeny constant of the coarse-grained dynamics as the objective function to maximize, in order to identify the optimal clustering. Given that the Kemeny constant is related to MFPTs by a simple relation, it turns out that for systems diffusing in a 1D potential, one can calculate analytically the position of the optimal boundaries between clusters.

##### 4.3.5.1 Diffusion in 1D symmetric potential: 3-state clustering

In this section we consider a 1D system diffusing in a symmetric potential  $U(x)$ . We first consider clustering the system into three clusters. Since the potential is symmetric, we assume that the position of the boundaries separating the clusters are also symmetric, and positioned at  $-a$  and  $a$ , where 0 is the center of the 1D space. It is the aim of the clustering to determine the most suitable cluster boundary  $a$ .

Considering the definition of the Kemeny constant in Eq. (285), taking the central cluster, denoted "2" and containing the region  $(-a, a)$ , as the starting state, and using the symmetry of the boundaries, we obtain

$$\zeta^{CG} = 2\Pi_1 t_{12}^{CG} \quad (307)$$

where  $\Pi_1$  is the equilibrium occupation probability of cluster 1, which is equal to  $\Pi_3$  by symmetry, and  $t_{12}^{CG}$  is the MFPT from  $I = 2$  to  $J = 1$  in the coarse-grained system. In Ref. [86] we have derived an expression for  $t_{JI}^{CG}$  for discrete-state Markov processes. By replacing summations with integrals in that expression, we can similarly write, for Markov processes in continuous space,

$$t_{JI}^{CG} = \frac{1}{\Pi_I \Pi_J} \int_I dx \int_J dy \pi(y) t_{yx} \pi(x) - \frac{1}{\Pi_J^2} \int_J dx \int_J dy \pi(y) t_{yx} \pi(x) \quad (308)$$

where  $\pi(x) = e^{-U(x)/kT}/Z$  is the Boltzmann distribution and  $Z = \int dx e^{-U(x)/kT}$  is the partition function.

Both  $\Pi_1$  and  $t_{12}^{CG}$  are dependent on  $a$ , and one can maximize  $\zeta^{CG}$  with respect to the position of the boundary  $a$ , by equating  $\partial \zeta^{CG} / \partial a$  to zero, which results in the following relation (see Appendix C.4)

$$\Pi_1 t_{-aa} - \bar{t}_{a2} + \frac{\Pi_2}{\Pi_1} \bar{t}_{-a1} = 0 \quad (309)$$

where

$$\bar{t}_{xI} = \int_I dy \frac{t_{xy} \pi(y)}{\Pi_I} \quad (310)$$

denotes the MFPT from cluster  $I$  to a single position  $x$ .

Moreover, we can write (see derivation in Appendix C.4)

$$\frac{\partial \zeta^{CG}}{\partial a} = 2\pi(a)[t_{3a}^{RT} - t_{2a}^{RT}] \quad (311)$$

where

$$t_{J\alpha}^{RT} = \bar{t}_{\alpha J} + \hat{t}_{J\alpha} \quad (312)$$

represents the round-trip time[99, 100] between the single position of the boundary  $\alpha$  and the cluster  $J$ , with  $\hat{t}_{J\alpha}$  defined in Eq. (388). This quantity naturally provides a distance metric based on mean first passage times, as it satisfies the symmetry property  $t_{IJ}^{RT} = t_{JI}^{RT}$  and the triangle inequality. Interestingly, the round-trip times for two-state systems have been related to the flux between the two states[101]. Moreover, round-trip

times, also called commute times, have been previously considered for clustering[102, 103] and applied for image classification [104].

From Eq. (311), it is clear that the optimal position of the boundary  $a$  is such that the round-trip times from the clusters that it separates are equal:

$$\frac{\partial \zeta^{CG}}{\partial a} = 0 \Leftrightarrow t_{3a}^{RT} = t_{2a}^{RT} \quad (313)$$

By symmetry,  $-a$  makes the round-trip times  $t_{1-a}^{RT}$  and  $t_{2-a}^{RT}$  equal.

To test the result in Eq. (309) on numerical examples, we consider systems diffusing in 1D double- and triple-well potentials of mean force (PMF), given by

$$U(X) = \cos(\alpha(X + \pi)) + e^{-X^2} \quad X \in [-\pi; \pi] \quad (314)$$

where  $\alpha = 2, 3$  is the number of wells (local minima).

To map the continuous motion in 1D to a Markov process on a linear chain, we discretize the configuration space  $X$  in  $N$  bins, that we label with  $i$ . We define the transition rates governing the motion between the bins as  $K_{ji} = 0$  for  $j \neq i \pm 1$  and

$$K_{(i\pm 1)i} = A e^{-[U(i\pm 1) - U(i)]/2k_B T} \quad (315)$$

where  $A$  is a constant set to 1,  $T$  is the temperature of the system, set to 298 Kelvin in our examples.

Then the MFPT matrix  $\mathbf{t}$  is obtained from the transition rates matrix  $\mathbf{K}$  using the formula [86]

$$t_{ji} = \frac{1}{\pi_j} [(\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})_{jj}^{-1} - (\boldsymbol{\pi} \mathbf{1}_n^T - \mathbf{K})_{ji}^{-1}] . \quad (316)$$

From these,  $\zeta^{CG}$  is computed by using the definition of the Kemeny constant  $\zeta^{CG} = \sum_{J \neq I} \Pi_{JI} t_{JI}^{CG}$  and Eq. (308) for the coarse-grained MFPTs. In addition, the LHS of Eq. (309), denoted as  $\Delta\zeta$ , is computed for each value of the boundary position  $a$ , using Eq. (316).

Fig. 61 shows  $\Delta\zeta$  as a function of the boundary position  $a$  and the position of the boundaries for which  $\zeta^{CG}$  is maximized, for  $\alpha = 2$  panel (a) and  $\alpha = 3$  panel (b) respectively. At these positions,  $\Delta\zeta$  vanishes as expected. For comparison, the optimal boundary position resulting from the maximization of the slowest timescale of the clustered system,  $\tau_2^{CG}$ , are shown in the same plot. We compute  $\tau_2^{CG}$  for each value of  $a$  using the relation [85]  $\tau_2^{CG} = \Pi_1 t_{-aa} + \bar{t}_{-a1}$ . The boundary position that maximizes  $\tau_2^{CG}$  is known to satisfy [85]:

$$\Pi_1 t_{-aa} - \bar{t}_{-a1} = 0 \quad (317)$$

The left hand side of Eq. (317), denoted with  $\Delta\tau_2$ , is seen to vanish at the positions that maximize  $\tau_2^{CG}$ , as expected.

When the number of potential wells (i.e. metastable states) is lower than the number of clusters, both methods identify a transition cluster, which is broader when the Kemeny constant is maximized. Conversely, when the number of potential wells is greater or equal than the number of clusters, the two methods lead to very similar boundaries positions, which are found around the top of the potential barriers (Fig. 64). The approach presented here assumes a symmetric potential and symmetrically placed optimal boundaries. In the next section, we relax these assumptions and develop a more general method, applicable to an arbitrary number of boundaries and clusters.

#### 4.3.5.2 Diffusion in 1D potential: $m$ -state clustering

Next, we consider coarse-graining a system diffusing in a 1D asymmetric potential into  $m$  states. When considering motion in a continuous one-dimensional space, with clusters defined by the positions of the separating barriers  $b_L$ , we can compute the derivative of the coarse-grained Kemeny constant with respect to any barrier position  $b_L$ :

$$\frac{\partial \zeta^{CG}}{\partial b_L} = \pi(b_L) \left[ t_{b_L(L+1)}^{RT} - t_{b_L L}^{RT} \right] \quad (318)$$

Where  $\pi(x)$  is the probability density function of the position in space, and  $t_{J\alpha}^{RT}$  is the round-trip time given in Eq. (312). See the full derivation of the result in Appendix C.5.

From Eq. (318), it is clear that the position of the barrier  $b_L$  maximises the coarse-grained Kemeny constant when  $b_L$  is at equal round-trip distance from cluster  $L$  and cluster  $L + 1$ .

#### 4.3.5.3 Random walk on linear chain: $m$ -state clustering

In the case of a 1D lattice, or linear chair, we can obtain a computationally efficient formula for the finite difference  $\Delta\zeta^{CG}$  when moving each boundary position  $b_J, J = 1, \dots, m - 1$  from cluster  $J + 1$  to  $J$  (that is the discrete-space analogue of the partial derivatives of the Kemeny constant with respect to the barrier positions). This is found to be (see Appendix C.6 for details):

$$\begin{aligned} \Delta\zeta^{CG}(b_J, J + 1 \rightarrow J) = \\ \pi_{b_J} \times \left[ \frac{\sum_{S < J} \Pi_S}{\Pi_J^2} \sum_{j \in J} \pi_j t_{jb_J} - \frac{\sum_{S \geq J} \Pi_S}{\Pi_J^2} \sum_{i \in J} \pi_i t_{b_J i} \right. \\ \left. + \frac{\sum_{S \leq J+1} \Pi_S}{\Pi_{J+1}^2} \sum_{i \in J+1} \pi_i t_{b_J i} - \frac{\sum_{S > J+1} \Pi_S}{\Pi_{J+1}^2} \sum_{j \in J+1} \pi_j t_{jb_J} \right] \end{aligned} \quad (319)$$

The optimal boundary positions correspond to the set of boundaries where  $\Delta\zeta^{CG}(b_J, J + 1 \rightarrow J)$  is closest to 0.

To test the results of Eq. (319) on analytical potentials, we first identify the optimal solution by exhaustive search, represented by the vector  $\mathbf{b}^* = (b_1^*, b_2^*, \dots, b_{m-1}^*)$ . We then compute the values of  $\Delta\zeta^{CG}(b_J, J + 1 \rightarrow J)$  for all possible positions of  $b_J$  between the optimal boundaries  $b_{J-1}^*$  and  $b_{J+1}^*$ , where the remaining boundaries  $\mathbf{b}^{*(J)} = (b_1^* \dots b_{J-1}^*, b_{J+1}^* \dots b_{m-1}^*)$  are kept constant. To this purpose, we considered the symmetric three-well potential defined by Eq. (314) with  $\alpha = 3$ , and an asymmetric four-well potential with barriers of varying height, defined by

$$U(X) = 0.1X(\cos X - 1) \quad \forall X \in [0, 8\pi] \quad (320)$$

For the symmetric three-well potential (Fig. 60a) the boundary positions that maximize  $\zeta$  are found at the two local maxima of the potential, consistently with results from Fig. 61a. This is also the case in the asymmetric four-well potential (Fig. 60b), where the optimal boundary is identified at the top of the barrier. In both examples, the optimal positions coincide with the locations where the numerical derivative given in Eq. (319) is theoretically predicted to vanish (Fig. 60 colored curves with symbols).

#### 4.3.5.4 Random walks on complex networks: $m$ -state clustering

The result provided in Eq. (318) extends to higher dimensional lattices and complex networks. Several important processes can be modelled as random walks on complex networks. These include diffusive processes in higher dimensions: upon discretizing the configuration space of such systems in  $N$  states, there are in general multiple paths between two states  $i$  and  $j$ , as described by the links of a networks.

Using the assumption that the population of the selected node is small compared to the cluster populations,  $\pi_\alpha \ll \Pi_A$  and  $\pi_\alpha \ll \Pi_B$ , we find that the change in the coarse-grained Kemeny constant upon moving node  $\alpha$  from cluster  $A$  to  $B$  is given as:

$$\Delta\zeta^{CG}(\alpha, A \rightarrow B) = \pi_\alpha [t_{\alpha A}^{RT} - t_{\alpha B}^{RT}] \quad (321)$$

where  $\pi_\alpha$  is the equilibrium probability of node  $\alpha$  (see derivation in Appendix C.7). We note that in contrast to linear chains, in complex networks there is no natural 1D ordering of nodes, hence the analytical expressions given by Eq. (319) do not hold as they rely on relations of the type  $t_{ba} = t_{bj} + t_{ja} \forall j \in (a, b)$ , which are valid only when all transitions between  $a$  and  $b$  go via the intermediate state  $j$ .

Based on our final results from Eq. (321), the optimal assignment of node  $\alpha$  will be the closest cluster as measured via the round trip time distance (assuming that the individual population of  $\alpha$  is small compared to those of the clusters). Otherwise, we could increase the Kemeny constant by moving  $\alpha$  from its current cluster to another one, to which its round trip time is smaller. Therefore, in an optimal clustering of a large complex network that maximizes the Kemeny constant, each node belongs to its nearest cluster measured by the round trip time distance.

#### 4.3.6 Conclusion

We have shown how the relations between MFPTs and rate matrices can be used to introduce clustering protocols that preserve MFPTs. One such protocol leads to an expression for the coarse-grained rate matrix which coincides with the one derived by Hummer-Szabo, and can be computed at low computational cost when information about MFPTs and equilibrium distribution in the original system is available. We have shown that such coarse-graining leads to a variational principle for the Kemeny constant, which may be used to optimise the coarse-graining protocol. in addition, we have shown that an alternative definition of the coarse-graining protocol exists, which preserves the Kemeny constant. The identification of optimal ways to cluster states, based on this latter definition of the coarse-grained rate matrix, constitutes another interesting pathway for

future work. Additionally, it remains to be tested if our formalism using mean first passage times might help overcome statistical uncertainties [105] often arising in coarse-graining of MSMs using spectral decomposition methods.

We introduced variational clustering protocols using the Kemeny constant and the slowest relaxation time. Building upon our earlier computational and theoretical work on the Kemeny constant[17, 19, 86], we derived a novel theoretical expression for the gradient of the Kemeny constant with respect to clustering, and provided an analytical solution for the optimal clustering of a 1D potential maximizing the Kemeny constant, for an arbitrary number of clusters. We showed that boundaries in optimal clustering correspond to equal round-trip times between clusters they separate. We further extended this result to complex networks in general, and showed that in the optimal clustering maximizing the Kemeny constant each node belongs to the cluster that is the closest as measured by the round-trip time distance measure.

The optimization using the Kemeny constant appears to successfully identify the key metastable states (See next Section (??)), in both weighted and unweighted networks, illustrating the proof of the variational result[86]. Moreover, maximizing  $\tau_2$  has shown to consistently identify less stable states, which can be considered as transition clusters, in addition to the metastable states. These transition clusters appear to contain nodes with high closeness and betweenness centrality. While the vast majority of clustering methods focus on detecting communities (metastable states), we present results suggesting that the use of specific timescales  $\tau_i$  can be used to also efficiently identifying key transition states in complex systems.

This new approach of maximizing specific timescales ( $\tau_i$ ) or their sum (Kemeny) opens up new and exciting potential research avenues and applications. For one, this method can effectively identify key transition states from a Markov model. In the context of molecular simulations, this can be incorporated to analyse and enhance the sampling of the system's configuration space[106].

Furthermore, the novel analytical expression for the gradient of the Kemeny constant,  $\partial\zeta^{CG}/\partial b_J$  and its discrete form using round trip times, derived in this work, also opens the possibilities to employ alternative implementations of our clustering method, based on e.g., gradient descent techniques, thus avoiding an exhaustive search over the space of possible boundary positions. This will be explored in future work.

## 4.4 RESULT 3 (Published) ALGO - Parralel Tempering Variational Clustering

### 4.4.1 Introduction

To more effectively manage large complex networks, an intuitive interpretation is crucial, which often requires the identification of clusters (or communities)[107–109]. Algorithms for cluster detection in unweighted networks typically rely on topological features of the network, i.e., they depend only on the usually static connection patterns and not on the processes which are taking place in the networks. Exact approaches consider all the different partitions into sub networks and optimize suitable quantities such as the so-called modularity [73, 110]. However, these scale poorly with the size of the system: when clustering a network with  $N$  nodes into a coarse-grained one with  $m$  nodes, there are  $m^N$  possible partitions. Hence, most clustering problems are known to belong [111] to the complexity class of non deterministic polynomial-time (NP) hard problems [112].

Such complexity also arises when performing kinetic clusterings of Markov State Models (MSMs). These are memoryless dynamical processes evolving on a finite set of states, which can be seen as weighted networks, where the nodes represent the Markov states and the edge weights are given by the transition rates[10, 13–15]. MSMs, or kinetic transition networks, have recently become a prominent tool for modelling and interpreting large simulation data-sets of complex kinetic systems in many domains of academia and industry[113–115].

In the context of coarse-graining MSMs, microstates of the system (nodes) are grouped together into macrostates (clusters) which are normally taken as the metastable states of the system, i.e., sets of microstates between which the system is slow to move. Hence many algorithms for kinetic clustering are related to maximizing the metastability of the clusters[79, 80, 116–124].

These clusters are further used to define coarse-grained versions of the original system, where the nodes (or states) are the clusters and the original kinetics is approximated by effective transitions between clusters[125]. This leads to a transition rate matrix in coarse-grained space, whose eigenvalues and eigenvectors will differ from those of the original transition matrix, resulting in different kinetics. An alternative to maximizing the metastability of the clusters is to maximize the timescales of the eigenprocesses in the clustered system, aiming to variationally preserve the slowest relaxation times.

Our novel implementation aims at accelerating the search for optimal clustering. Our approach leverages *parallel tempering* [49–51], which is an efficient physics-inspired optimisation algorithm [126, 127] based on interpreting the target function to optimize as a physical energy and coupling the optimization process to several artificial heat baths in parallel. Our algorithm can similarly be applied to optimize other objective functions, including the slowest timescale used in Ref. [17] to detect transition states. In this way, we provide a computationally efficient way to automatically detect communities as well as transition clusters in complex networks, based on kinetic properties of network processes.

In section (4.4.3) we will present a clustering algorithm, that we name the "parallel tempering variational clustering" (PTVC), for optimizing clusters using arbitrary objective functions on complex networks, based on the parallel tempering method used in statistical physics. Then in section 4.4.4 we compare clustering resulting from optimizing different objective functions i.e. Modularity, Stability, and Kemeny constant, as well as using the Perron cluster analysis on synthetic and real-world networks.

#### 4.4.2 Original Parallel Tempering

Considering the case of clustering a network with  $N$  nodes into a coarse-grained one with  $m$  nodes, results in  $m^N$  possible partitionings. Hence exact clustering algorithms scale poorly with the size of the system, with most clustering problems known to be NP-hard [111], therefore one often uses approximate algorithms to obtain a solution efficiently.

One approach, known as the Louvain method,[74, 128] is to iteratively join clusters together until the objective function (modularity) is maximized. Although this approach is fast to execute, it is not readily applicable to many objective functions, such as the Kemeny constant, which always decreases when joining clusters together, following Eq. (288).

Another approach to reducing the complexity of the cluster assignment, consists in defining an ordering for the nodes in the network, *i.e.*, in projecting the nodes onto a one-dimensional space. Subsequently, the task of finding  $m$  optimally selected subsets in a network of  $N$  nodes, is reduced to finding  $m - 1$  separating boundaries. Two spaces that have been used for the projections are the eigenvector associated to the slowest process, and the committor probability with respect to the two most "distant" nodes in the network, often defined as the pair  $(i, j)$  with the largest MFPT,  $t_{ij}$ . This however requires the computation of the MFPTs for all the nodes in the network, which has a large computational cost for large networks.

The problem of extremizing a function dependent on the cluster assignment of the nodes in a network is very common in statistical physics. In particular, assigning a cluster to a node is equivalent to assigning a spin value to a node in the Potts model [59, 129, 130], hence one can apply methods devised in statistical physics to simulate large physical systems for the task at hand.

Parallel tempering[49, 50, 131–133] is a simulation method [134] that was developed for the purpose of studying physical systems coupled to a heat bath over extended temperature intervals, in particular down to small temperatures. It consists in running  $s$  parallel simulations of the same system, *i.e.*, different replicas, each at a different temperature [135] and allows replicas to be exchanged between neighbouring temperatures. This results in the high temperature replicas exploring the configuration space freely while the low temperature replicas exploring the low energy regions more extensively, hence overall reducing the time to escape from a local minimum of the energy. When concentrating on the sampled configurations of replicas while being at the very low temperatures, the approach was in particular used to find the global minimum energy configurations of complex physical systems, for example for spin glasses, which is NP-hard. [136] Clearly, any objective function of a system to be optimized can be considered as an "energy" which allows one to use parallel tempering as a general-purpose optimization method [126, 127].

Here, we use the Parallel Tempering algorithm to obtain clusterings, therefor we use a corresponding notation. In detail, every simulation of a replica  $\alpha$  evolves according to a standard Metropolis-Hastings algorithm. This means, one generates a Markov chain  $\mathbf{S}^{(0)} \rightarrow \mathbf{S}^{(1)} \rightarrow \dots$  of clustering configurations (we omit the  $\alpha$  here for brevity), which is not to be confused with the Markov chains of the MSM. For a given clustering  $\mathbf{S} = \mathbf{S}^{(l)}$  at "time"  $l$ , exhibiting energy  $E = E(\mathbf{S}^{(l)})$ , a trial configuration  $\mathbf{S}'$  is randomly constructed, where the corresponding energy is denoted as  $E'$ . Typically  $\mathbf{S}'$  is obtained by a random small change to  $\mathbf{S}$ . The construction probability that  $\mathbf{S}'$  is obtained from  $\mathbf{S}$  is denoted as  $C(\mathbf{S} \rightarrow \mathbf{S}')$ . Often this is constant, but not always. Now, the trial configuration will become the next configuration in the Markov chain, *i.e.*,  $\mathbf{S}^{(l+1)} = \mathbf{S}'$  with the acceptance probability  $p_{\text{acc}}^{(\alpha)}$  that depends on the difference between the energies  $E$  and  $E'$  and on the temperature  $T^{(\alpha)}$  of the replica  $\alpha$

$$p_{\text{acc}}^{(\alpha)} = \min \left( 1, e^{(E - E') / k_B T^{(\alpha)}} \frac{C(\mathbf{S}' \rightarrow \mathbf{S})}{C(\mathbf{S} \rightarrow \mathbf{S}')} \right) \quad (322)$$

where  $k_B$  is the Boltzmann constant. Otherwise, with probability  $1 - p_{\text{acc}}^{(\alpha)}$ , the current configuration will be kept, *i.e.*  $\mathbf{S}^{(l+1)} = \mathbf{S}^{(l)}$ . Note that the fraction is one if the construction probability is constant. This choice for the acceptance probability ensures detailed balance between neighboring configurations at the same temperature, hence leads to convergence to the Boltzmann equilibrium measure. A standard time unit in this standard simulation is a sweep. Within a sweep, each degree of freedom is allowed to be changed on average once within a trial configuration.

At regular time intervals, typically after a sweep for each replica is performed, configurations at different

temperatures (in different replicas) are interchanged with a probability

$$p_{\alpha\beta} = \min \left( 1, e^{\left( E^{(\alpha)} - E^{(\beta)} \right) \left( \frac{1}{k_B T^{(\alpha)}} - \frac{1}{k_B T^{(\beta)}} \right)} \right) \quad (323)$$

where  $\alpha$  and  $\beta$  are the indices of the simulations being interchanged and  $E^{(\alpha,\beta)}$  are the energies of the configurations in replicas  $\alpha$  and  $\beta$ , respectively. Here again detailed balance is guaranteed, hence the simulation will equilibrate with respect to the product measure of two neighboring replicas. In practice, one usually only considers moves between replicas where the temperatures are direct neighbours in the ordered list of temperatures. This applies to all pairs of (neighboring) temperatures, i.e., for the full system.

The temperature set  $\{T^\alpha\}$  needs to be specified. A standard rule of thumb is that the empirical acceptance frequency of an exchange between neighbouring temperatures should be roughly 0.5. Usually one performs short test simulations for various sets of temperatures. Thus, if the observed frequency is too small, one would move the respective two temperatures closer to each other, possibly one has to increase the number  $s$  of temperatures. For very high acceptance frequencies, it is opposite. Typically, this results in sets where at low temperatures the differences between neighbouring temperatures are small, while at high temperatures, the differences are large. In the present work, we follow the same main idea, but use a different protocol as explained in Sec. 4.4.3.

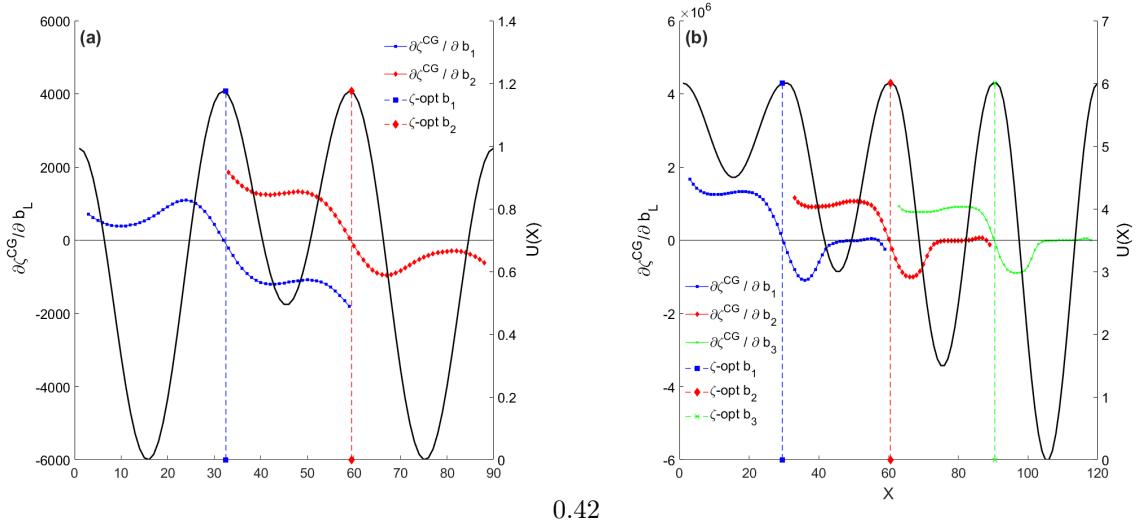


Figure 60: Three-state **(a)** and four-state **(b)** clustering of system diffusing on symmetric **(a)** and asymmetric **(b)** 1D potentials (black line, y axes on right in kcal/mol units), respectively. Colored dashed lines show the boundary positions corresponding to the global maximum of  $\zeta^{CG}$ , found by exhaustive search. Dashed lines show  $\partial \zeta^{CG} / \partial b_J$  with each color (blue, red or green) and associated symbol (square, diamond and x) corresponding to a different value of  $J$  (1, 2, 3, respectively), were computed using Eq. (319) for all possible positions of  $b_J \in (b_{J-1}, b_{J+1})$  when the optimal positions  $b_I$  for all  $I \neq J$  are kept fixed.

#### 4.4.3 Parallel Tempering Variational Clustering

Here we propose to use a parallel tempering approach for finding the clustering that optimizes an arbitrary objective function.

In the context of network clustering, we replace the energy term  $E^\alpha$  in Eq. (322) and Eq. (323) by the objective function one aims to optimize. The temperature  $T_i^\alpha$  is a parameter that governs the acceptance probability of a change in clustering assignments, such that the larger  $T_i^\alpha$ , the higher the probability of accepting a move that does not improve the objective function.

To generate an initial starting configuration for the parallel tempering algorithm, we generate an easy to obtain clustering. Therefore, we first find a one-dimensional ordering of the nodes of the network. For this purpose, we consider a kinetic process on the network, characterized by the rate matrix  $\mathbf{K}$ , and we determine the first and the last nodes of the ordering,  $i$  and  $j$  respectively, as those with the largest MFPT  $t_{ij}$ . Although the MFPTs are not symmetric, this definition of first and last nodes, that we will refer to as "extreme nodes" (red nodes in Fig. 62), allows for a meaningful projection as a starting point of our simulation, while remaining applicable to any system. The remaining nodes are then ordered based on their committor probabilities, i.e., the probability to first reach one state before the others. With this 1D ordering of the states, different clusterings are generated randomly, by placing boundaries randomly along the 1D coordinate. The clustering with the highest

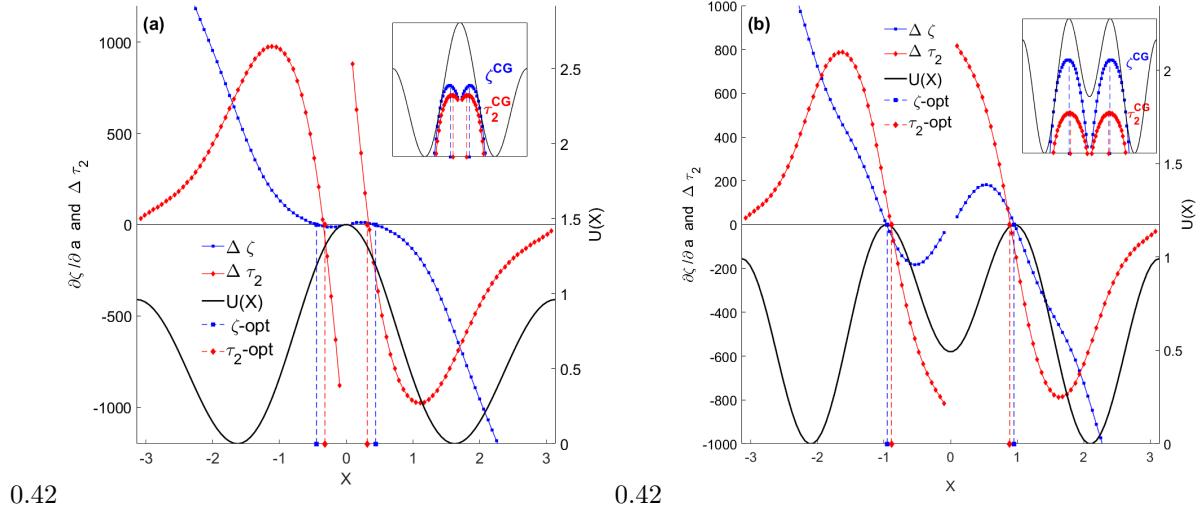


Figure 61: Three-state clustering of system diffusing on symmetric 1D double well (a) and triple well (b) potentials (black line, y axes on right in kcal/mol units). The dashed lines show the boundary positions corresponding to maxima of the relaxation time  $\tau_2^{\text{CG}}$  (red diamonds, dashed red line) and Kemeny constant  $\zeta^{\text{CG}}$  (blue squares, dashed blue line) in the clustered system. Symbols highlight the end points of these lines.  $\Delta\zeta$  (blue curve with squares) defined via the relation in Eq. (309) and  $\Delta\tau_2$  (red curve with diamonds) defined via Eq. (317) are plotted against the boundary position. The zeros of  $\Delta\zeta$  and of  $\Delta\tau_2$  yield the corresponding optimum boundary positions, respectively. The inset shows  $\zeta^{\text{CG}}$  (blue squares) and  $\tau_2^{\text{CG}}$  (red diamonds).

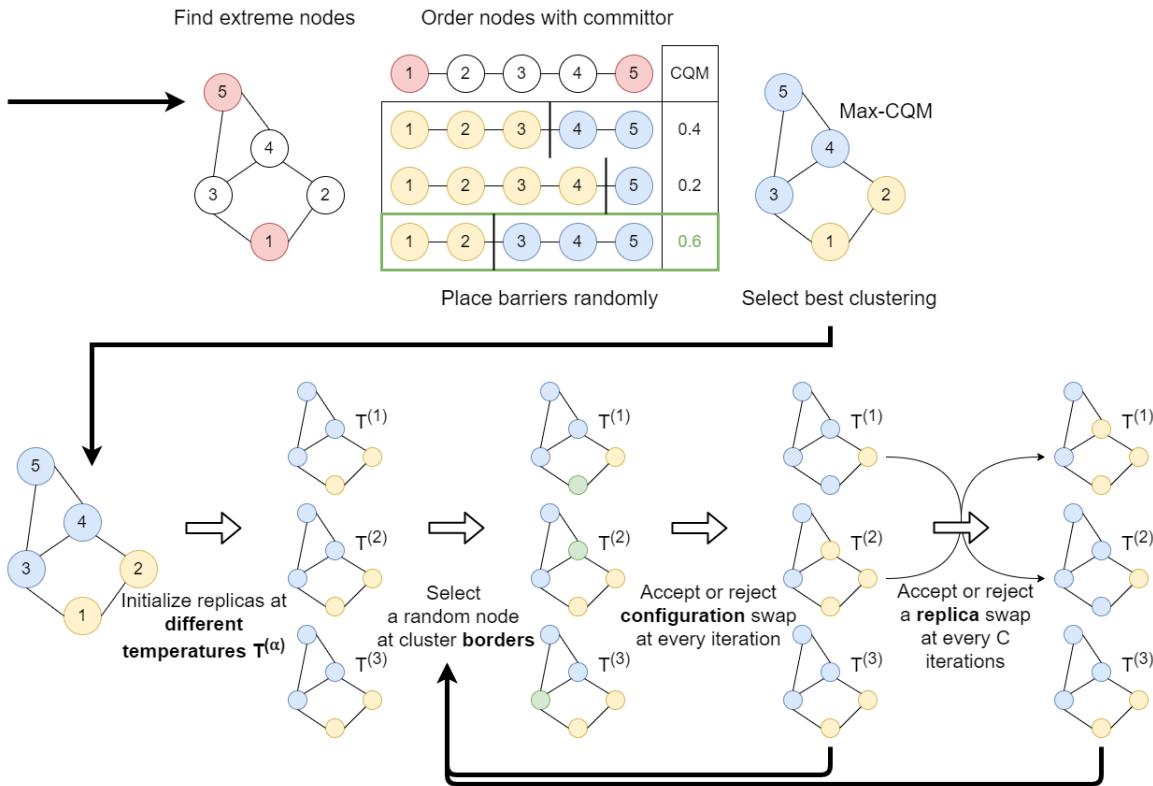


Figure 62: Illustration of PTVC algorithm. Extreme nodes (having maximal MFPT values) are indicated with red. Clusters are indicated with blue and yellow colors, selected bordering nodes indicated with green.

value of the objective function is selected, among those randomly generated, as the starting configuration for the PTVC algorithm.

After initialization of replicas at different temperatures, the PTVC algorithm will no longer be constrained to the 1D projection. Then for each replica, at each time-step of the simulation, we find all the border nodes in the networks and we sample uniformly at random one node  $i^*$  from this set, and then among the neighbouring

clusters of  $i^*$ , a cluster  $J^*$  is sampled uniformly again. Finally assigning node  $i^*$  to cluster  $J^*$  is proposed as a trail clustering. We define a node  $i \notin J$  as border note of cluster  $J$  if there exist a node  $j$  such that  $A_{ij}S_{jJ} = 1$ . We define a cluster  $I$  to be neighbour of a cluster  $J$  if there are two nodes  $i, j$  such that  $A_{ij}S_{jJ}S_{iI} = 1$ .

The value of the objective function for the clustering after assigning node  $i^*$  to cluster  $J^*$  is calculated, and the move is accepted or rejected following the acceptance probability in Eq. (322).

The method outlined above (and summarised in Fig. 62) can be applied to arbitrary parameters considered as objective function or energy  $E$ . We will use the Kemeny constant and the slowest timescale of the coarse-grained system, respectively, with the purpose of identifying both stable clusters and key transition states.

Other methods in the field of network cluster identification have employed the idea of introducing the concept of temperature to accelerate a variational search through conformations [65], 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.

Our method employs the kinetic timescales of the system as the variational parameter to identify transition states, as opposed to modularity, and we introduce a kinetically motivated initial ordering of the states to enhance the quality of the starting clustering.

#### 4.4.3.1 PTVC algorithm implementation details

Following Eq. (322), the Metropolis-Hastings acceptance probability shall be  $p_{\text{acc}}^\alpha = \min(1, e^{(E-E')/kT^{(\alpha)}} \frac{C(\mathbf{S}' \rightarrow \mathbf{S})}{C(\mathbf{S} \rightarrow \mathbf{S}')})$  to ensure detailed balance, i.e., equilibrium sampling over all temperatures  $T^{(\alpha)}$ . Usually, the number of border nodes and neighbour clusters will differ before and after the proposed move, hence the term  $\frac{C(\mathbf{S}' \rightarrow \mathbf{S})}{C(\mathbf{S} \rightarrow \mathbf{S}')}$  is not constant. Nevertheless we set this term to 1 in our applications as the preservation of detailed balance is not required for the optimization task, hence lowering the computational cost.

One practical consideration that requires discussion is the choice of temperature to be used, which has no physical motivation here and exists only to control the probability of accepting proposed switches for different simulations. To determine the temperature to use, we propose to initialise the  $s$  simulations at equally spaced increasing temperatures values  $T = T^{(i)}$  where  $i \in \{1, \dots, s\}$ , with  $T^{(1)} < T^{(2)} < \dots < T^{(s)}$ . In our applications presented here, we have used  $s = 50$  simulations, and initial temperatures bounded by  $T^{(1)} = 0.001$  and  $T^{(s)} = 1$ .

This effectively enforces that at first mostly the proposed moves which optimize the parameter are accepted. From here, we can then increase the temperatures until the average acceptance probability reaches a desired value, that we set to 50%. We achieve this by computing the proportion of accepted moves since the latest temperature update ( $p^a$ ) every 10 sweeps, and then update the temperatures using the formula  $T^{new} = T \times \frac{\log(p^a)}{\log(0.5)}$ . This ensures that temperatures are high enough for the system to explore the whole configuration space, but not high enough to cause the algorithm to accept every proposed move.

The value of the objective functions was monitored over simulations of 20000 sweeps, with  $s = 50$  different temperatures, and the clustering with the highest objective function was chosen.

#### 4.4.4 Clustering results

We use the PTVC algorithm in five test systems to maximize three different parameters: (i) the modularity (Eq. (266)), (ii) the Kemeny constant (Eq. (288)), and (iii) the slowest timescale  $\tau_2$  (Sec. 4.2.3.4). We compare the resulting clusterings with that of PCCA+. Our test systems consist of a diffusion process in a 1D potential, a synthetic network generated from the stochastic block model [137] and three real-world networks.

##### 4.4.4.1 One-dimensional energy profile

Before applying the method to complex networks, we test the PTVC method on a simple 1D model, where results can be easily interpreted and compared to an exhaustive search for the global maximum.

We consider a potential with four wells of varying depth, defined by Eq. (422). When clustering into three states, we obtain identical clusters for PCCA+, the Kemeny constant and  $\tau_2$  clustering (Fig. 63). These correspond to the three most stable local minima on the potential energy profile separated by the top of the barriers. However, modularity clusters the coordinate space into three equal clusters, independently on the underlying free energy profile. This is expected as all networks corresponding to a 1D profile have the same tridiagonal connectivity. The optimal 4-state clustering on the same potential (Fig. 64 and Table 1) also results in the same clusters for PCCA+ and the Kemeny constant, identifying all four local minima. Analogously as for 3-state clustering, modularity identifies an equal spacing of the reaction coordinate into 4 parts, which does not take into account the free energy profile. Interestingly, however,  $\tau_2$  clustering results in a fourth state that is a transition state (TS), suggesting that the highest energy local minimum is less important for the slowest

timescale of the process, and a cluster with the TS node is more optimal to maximize  $\tau_2$ . Therefore, the number of metastable clusters can be determined by observing the first time the maximization of  $\zeta^{CG}$  and  $\tau_2^{CG}$  yield different results. Analogously, the eigenvalue spectrum also suggests three metastable states for this system (Fig. 72a). In general, once we move beyond the number of metastable states, the Kemeny constant identifies smaller metastable states, and  $\tau_2$  finds a cluster with very small population around the dominant barrier.

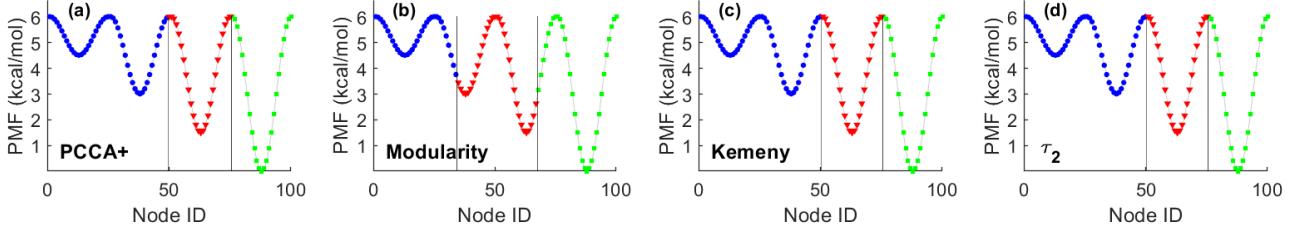


Figure 63: Clustering a 1D multi-well potential into three clusters (blue circles, red triangles, and green squares symbols with black vertical lines showing the boundaries) using PTVC with the three different target functions: Modularity (b), the Kemeny constant (c) and  $\tau_2$  (d), respectively. PCCA+ clustering (a) also shown for comparison.

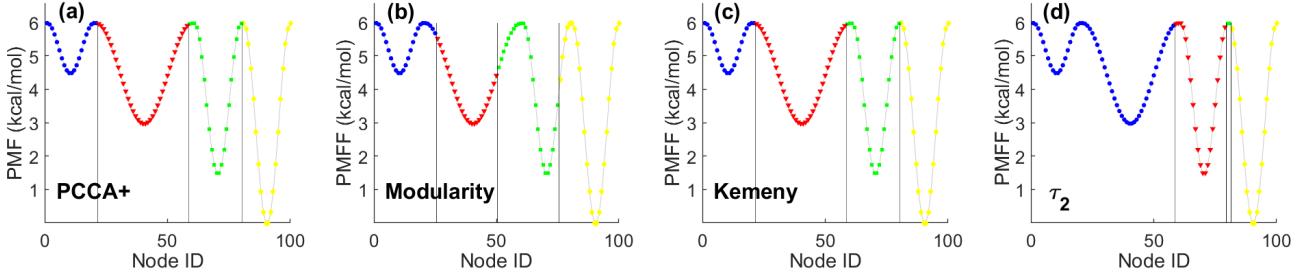


Figure 64: Clustering a 1D multi-well potential into four clusters (blue circles, red triangles, and green squares symbols with black vertical lines showing the boundaries) using PTVC with the three different target functions: Modularity (b), the Kemeny constant (c) and  $\tau_2$  (d), respectively. PCCA+ clustering (a) also shown for comparison.

Table 1: One-dimensional Potential Energy Surface reduced to four clusters as shown in Fig. (64). Values of each objective function: the Kemeny constant,  $\tau_2$  and Modularity (columns) for each clustering method PCCA+, objective functions Kemeny constant,  $\tau_2$  and Modularity (rows) are shown. The largest value for each objective function is highlighted in bold.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	<b>53185</b>	41780.6	0.41716
Kemeny	<b>53185</b>	41780.6	0.41716
$\tau_2$	52308	<b>41781.3</b>	0.38043
Modularity	52865	41660.2	<b>0.42234</b>

#### 4.4.4.2 Stochastic block model

To analyze more complex networks beyond the one-dimensional connectivity, we generated 3-state random networks using the stochastic block model (SBM) (Fig. 65). The adjacency matrix of a random SBM network is constructed according to Eq. (324).

$$P(A_{ij} = 1) = \frac{cW(X_i, X_j)}{NP(X_i)P(X_j)} \quad (324)$$

where  $X_i \in 1, \dots, m$  indicates the cluster of node  $i$ ,  $P(X)$  is the probability of a node being in cluster  $X$ ,  $W(X, X')$  is the probability of a link existing between cluster  $X$  and  $X'$ ,  $N$  is the total number of nodes in the network, and  $c = 4$  is the average connectivity of the network. In our applications we have used the uniform distribution for  $P(X) = 1/m$ , with  $m = 3$  clusters,  $N = 99$  nodes in total, and intra-cluster and inter-cluster edge existence probability of  $W(X, X) = 0.7$  for all  $X$  and  $W(X, X') = 0.005$  for all  $X \neq X'$ , respectively.

For SBM networks there is no notion of dynamics, hence we consider a random walk on the network, with the Laplacian introduced in Eq. (264). Therefore, the dynamics of an unbiased walker is characterized by frequent intra-cluster transitions and rare inter-cluster transitions, so that the network communities correspond to the clusters as metastable states.

We used the PTVC algorithm to identify three clusters within a 3-state SBM network for all three variational parameters. As expected, we obtain the correct clustering using all three methods as well as using PCCA+ (Fig. 73 and Table 7).

The results for four clusters (Fig. 65 and Table 2) show a consistent picture with the 1D example. PCCA+ and Kemeny subdivides one of the clusters in two approximately equal-sized clusters, whereas  $\tau_2$ -optimal clustering identifies a transition cluster separating one small metastable cluster from the others. Modularity identifies the three expected clusters and assigns a single node to an additional 4th cluster.

Moreover, we find that the identified TS contains both the nodes with highest closeness and betweenness centrality (Fig. 74), which indeed reflects the nature of transition states: they are positioned *between* metastable states, most of the paths go through them, and they are *close* to all the other nodes.

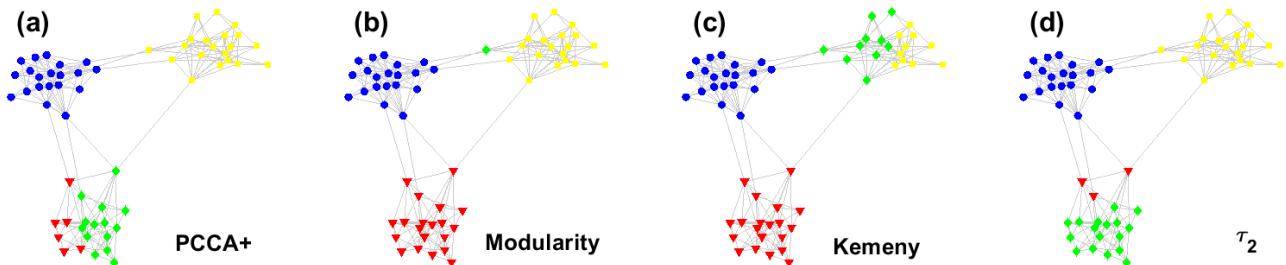


Figure 65: Clustering of a three-state stochastic block model network into four clusters using PTVC with the three different target functions: Modularity (b), the Kemeny constant (c) and  $\tau_2$  (d). PCCA+ clustering (a) also shown for comparison.

Table 2: Values of the Kemeny constant,  $\tau_2$  and Modularity for a three-state stochastic block model network reduced to four clusters using the PCCA+ method, and the optimization for the Kemeny constant,  $\tau_2$  and Modularity. Bolded numbers indicate the largest value of the respective objective function.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	65.342	40.774	0.3645
Kemeny	<b>65.574</b>	40.807	0.35908
$\tau_2$	65.373	<b>40.942</b>	0.37832
Modularity	65.091	40.796	<b>0.39652</b>

#### 4.4.4.3 Santa Fe Collaboration Network

As a first example of a real world network which is small enough for clear visualisation, we consider the Santa Fe collaboration network (Fig. 66)[138]. In this network, links are drawn between 118 researchers at the Santa Fe institute who appeared as co-authors on at least one publication. The nodes then form clusters corresponding to three main research groups, linked by cross-disciplinary researchers.

We apply random-walk dynamics on this network and cluster it into three states (Fig. 75 and Table 8), which corresponds to the number of metastable clusters predicted by the spectral gap (Fig. 72), we find that the resulting clusterings are identical for all four methods, as in the previously considered networks.

When clustering into four states (Fig. 75 and Table 9), we find that the  $\tau_2$ -optimal clustering still finds a transition cluster separating two of the metastable regions as expected. However the Kemeny-optimal clustering no longer behaves similarly to PCCA+ clustering, and is instead identical to the  $\tau_2$ -optimal one.

The fact that the Kemeny and  $\tau_2$ -optimal clusterings are identical can be seen as a consequence of the fact that the additional cluster is also metastable but too small to be detected via spectral gap, hence it is insightful to look for one more cluster. We find indeed that when optimizing for five clusters (Fig. 66 and Table 3), the Kemeny-optimal and  $\tau_2$ -optimal clusterings are no longer identical, and the latter successfully identifies the two regions separating the three initial metastable states into distinct clusters. Again we find that the two additional clusters contain nodes with high closeness and betweenness centrality (Fig. 76).

In conclusion, our method retrieves all the communities, when the number of clusters used coincide with the number of communities detected via spectral gap analysis, and it allows to identify transition clusters containing the nodes with the highest centrality when used with a larger number of clusters.

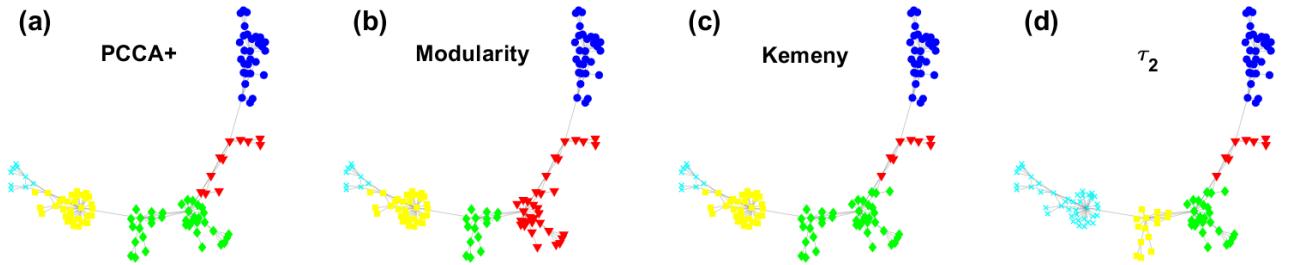


Figure 66: Clustering of the Sante Fe Institute collaboration network into five clusters using PTVC with the three different target functions: Modularity (b), the Kemeny constant (c) and  $\tau_2$  (d). PCCA+ clustering (a) also shown for comparison.

Table 3: Values of the Kemeny constant,  $\tau_2$  and Modularity for the Santa Fe Collaboration network reduced to five clusters using the PCCA+ method, and the optimization for the Kemeny constant,  $\tau_2$  and Modularity. Bolded numbers indicate the largest value of the respective objective function.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	335.58	238.51	0.41749
Kemeny	<b>336.2</b>	238.8	0.41504
$\tau_2$	333.11	<b>239.9</b>	0.41652
Modularity	334.53	236.52	<b>0.41999</b>

#### 4.4.4.4 Political Books co-purchasing Network

Our next real-world network is one of political books: each node represents a book, and an edge is present between two books if they appear as being often co-purchased on Amazon. Thus, for our clustering approach, we also use the random-walk dynamics. The data have been compiled by Valdis Krebs. [139] Based on a reading of the descriptions and reviews of the books on Amazon, the nodes have been labelled manually by Mark Newman into one of 3 categories: "liberal" (blue), "neutral" (red), and "conservative" (green) [140] (Fig. 77), providing a comparison to our clusterings.

The spectral gap analysis detects two metastable states (Fig. 72), corresponding to the categories "liberal" and "conservative" as we could expect, while the "neutral" category will be considered as the transition state between the two. We observe that partitioning into two clusters yields identical clusterings for all dynamical clustering methods and a very similar clustering for modularity (Fig. 78 and Table 10). When optimizing for three clusters yields results consistent with our previous observations (Fig. 67 and Table 4). On one hand the Kemeny constant and PCCA+ converge to a similar clustering. On the other hand the  $\tau_2$ -optimal clustering is seen to separate the two stable states (Fig. 67), while the small transition cluster contains nodes with high closeness and betweenness centrality (Fig. 79). Modularity does not separate the two stable states fully and identifies a third cluster somewhat different from Kemeny and PCCA+.

The manual labelling (Fig. 77) is not retrieved by any of the kinetic clusterings we have tried. This can be explained by multiple factors: the edges here are binary, *i.e.*, they do not measure *how strongly* connected (co-purchased) two books are, hence all the information about the strength of the connection is lost. Given that our method is based on the dynamics of the network, which is highly dependent on the weights of the edges (transition rates), we can expect it to lead to a different clustering in the presence of weights, which may be closer to the manual clustering with a weighted network. Additionally, the manual labelling have been made by a single person and therefore it is prone to the subjectivity of this person's judgment. The discrepancy between the  $\tau_2$ -optimal clustering and the labelling can be used to suggest reconsideration of the labelling, or to gather a more robust dataset for the task at hand. Furthermore, it also suggests that there is no completely neutral opinion, but some bias exists towards liberal or conservative orientation.

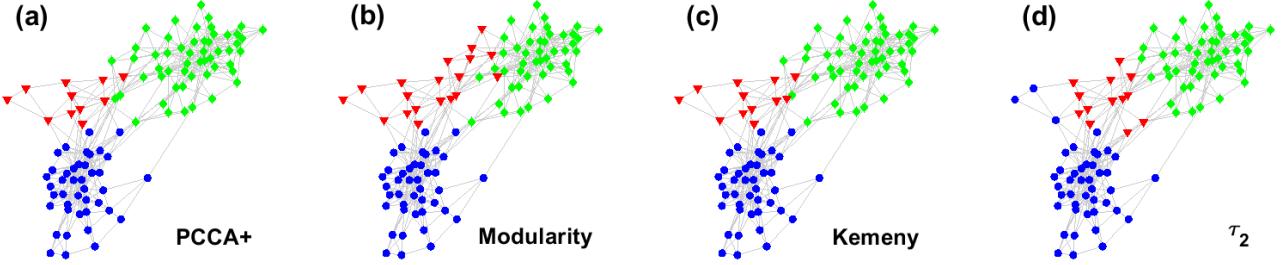


Figure 67: Clustering of the Political Books co-purchase network into three clusters using PTVC with the three different target functions: Modularity (b), the Kemeny constant (c) and  $\tau_2$  (d). PCCA+ clustering (a) also shown for comparison. The "Conservative" and "Liberal" books clusters (blue and green) are similarly detected by all parameters, with differences mainly on the "Transition" cluster (red).

Table 4: Values of the Kemeny constant,  $\tau_2$  and Modularity for the political book network reduced to three clusters using the PCCA+ method, and the optimization for the Kemeny constant,  $\tau_2$  and Modularity. Bolded numbers indicate the largest value of the respective objective function.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	29.944	25.041	0.35879
Kemeny	<b>29.954</b>	25.237	0.35843
$\tau_2$	29.023	<b>25.653</b>	0.34739
Modularity	29.524	25.062	<b>0.35954</b>

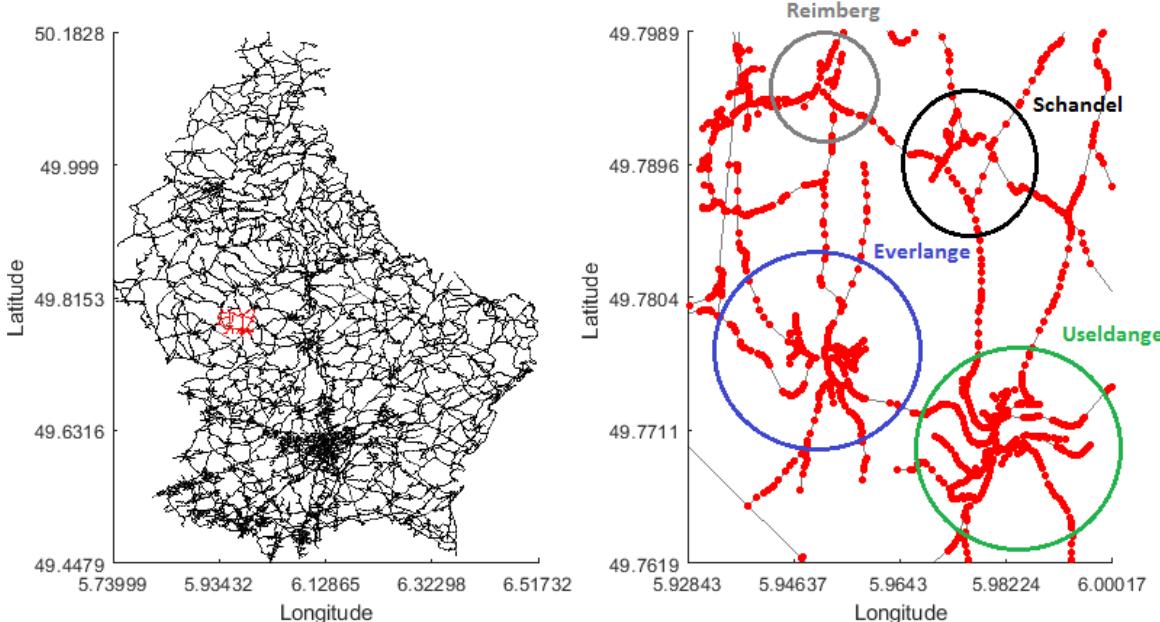


Figure 68: Street Network of Luxembourg from DIMACS10 [141, 142], and the subset we consider. Latitude and Longitude in Decimal Degrees, the colored circles highlight the main cities in this geographic area.

#### 4.4.4.5 Streets Network

Finally, we consider a subset of the Open Street Map road networks of Luxembourg (Fig. 68). The graph is an undirected and unweighted version of the largest strongly connected component of the corresponding Open Street Map road networks[142]. Due to the large size of the network (114599 nodes), we have taken a subset of the initial graph (Fig. 68, right), the subset contains two medium-sized cities (Everlange and Useldange), as well as a few smaller cities (Reimberg and Shandel).

In the absence of data about the dynamics in this network, we applied the Laplacian method described in section 4.2.1.2 to obtain the random walk dynamics. We find that the spectral gap analysis determines that there are two metastable states.

Similarly to what we observe in the previous networks, when optimizing for two clusters the partitions from all four methods provide similar results (Fig. 71, top and Table 5). Each of the two clusters contains one of the two largest cities of the network, Everlange and Useldange (Fig. 68, right).

When clustering into three states (Fig. 71, bottom and Table 6), we find that PCCA+ provides a sensible clustering, identifying the third largest city, Schandel, into the third cluster, similarly to the Kemeny-optimal clustering. The Modularity-optimal clustering splits the left-hand cluster, we can see that the additional cluster corresponds to a part of one of the largest cities (Everlange). The  $\tau_2$ -optimal clustering maintains the two main cities unseparated, and instead assigns the third largest city as well as the main road connecting the two main cities into the third, transition cluster, thus completely separating the two main cities.

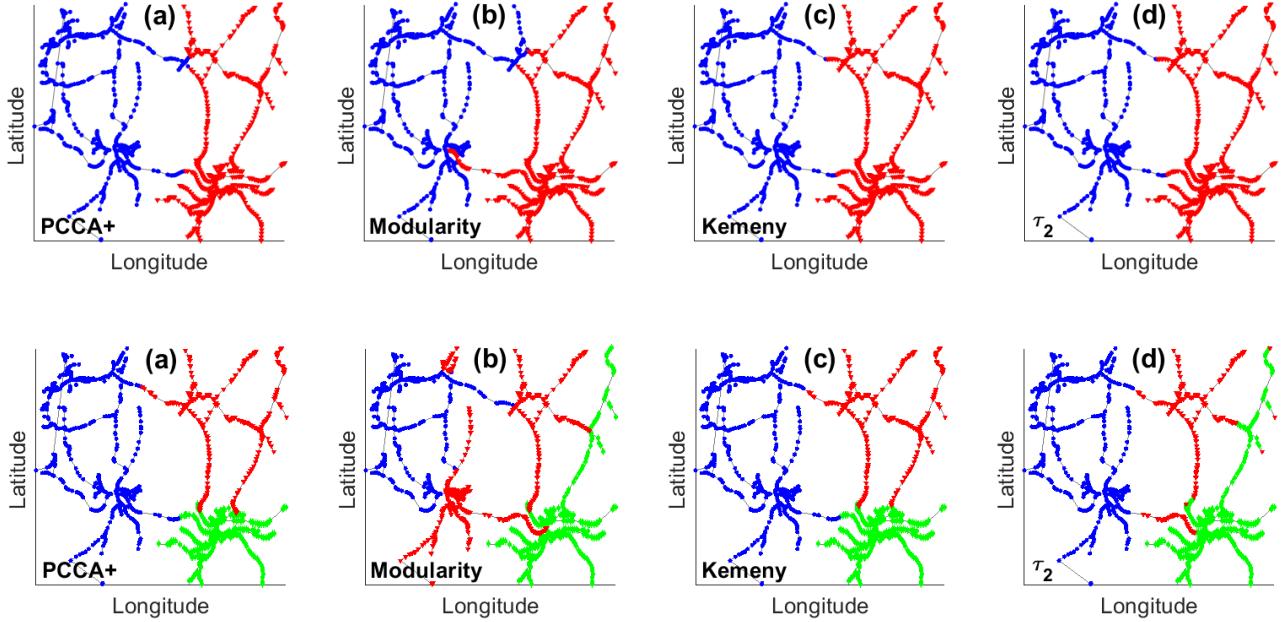


Figure 69: Clustering of the subset of Luxembourg streets network (Fig. 8, left) into two (top) and three (bottom) states.

Table 5: Values of the Kemeny constant,  $\tau_2$  and Modularity for the Luxembourg roads network reduced to two clusters using the PCCA+ method, and the optimization for the Kemeny constant,  $\tau_2$  and Modularity. Bolded numbers indicate the largest value of the respective objective function.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	14327	14327	0.37365
Kemeny	<b>14593</b>	<b>14593</b>	0.37404
$\tau_2$	<b>14593</b>	<b>14593</b>	0.37404
Modularity	13280	13280	<b>0.37367</b>

Table 6: Values of the Kemeny constant,  $\tau_2$  and Modularity for the Luxembourg roads network reduced to three clusters using the PCCA+ method, and the optimization for the Kemeny constant,  $\tau_2$  and Modularity. Bolded numbers indicate the largest value of the respective objective function.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	19973	15007	0.4057
Kemeny	<b>20007</b>	15035	0.40499
$\tau_2$	18167	<b>15374</b>	0.3999
Modularity	15260	13144	<b>0.41305</b>

#### 4.4.5 Conclusion

To enable more efficient clustering in complex systems, we described in the present work an algorithmic protocol, PTVC, allowing for the identification of clusters in large and complex networks by using parallel tempering to

maximize any variational parameter or objective function. Using this algorithm, we compared different objective functions (Modularity, Kemeny constant, and slowest timescale  $\tau_2$ ) as well as PCCA+ on the examples of two model networks and three real data-derived networks.

## 4.5 RESULT 4 (Unpublished) ALGO - (Faster algorithm inspired from Louvain Modularity maximization )

### 4.5.1 Introduction

The PTVC algorithm presented in our earlier work is not fast enough.

The Louvain Method of Modularity maximization have been shown to be very efficient [\cite{blondel2008fast, de2011generalized}], we will consider the application of its idea to the optimization of the Kemeny constant.

### 4.5.2 Original Louvain Method

The orginal Louvain method of modularity maximization have been shown to be very fast, and able to handle networks with 118 Million nodes.

The Louvain algorithm describes as follows:

Take node(/cluster) i,

compute the modularity gain if moving  $i$  to another community  $j$  :  $\Delta \text{Modularity}_{(i \rightarrow j)}$ ,  
add  $i$  to the cluster  $j$  with the highest modularity increase.

Note that  $\Delta \text{Modularity}_{(i \rightarrow j)}$  can be written as a matrix.

### 4.5.3 Louvain for Kemeny

We know that when using the Hummer Szabo reduction protocol, the Kemeny constant of a coarse-grained system  $\zeta^{CG}$  is always smaller than the one of the original system  $\zeta^{orig}$ , from the following equation:

$$\begin{aligned} \zeta^{CG} &= \zeta^{orig} - \sum_J \frac{1}{\Pi_J} \sum_{i \in J} \sum_{j \in J} \pi_j t_{ji} \pi_i \\ &= \zeta^{orig} - \delta(S) \end{aligned} \quad (325)$$

Hence, maximizing  $\zeta^{CG}$  is equivalent to minimizing the second term  $\delta(S)$ :

$$\delta(S) \equiv \sum_J \frac{F_{JJ}}{\Pi_J} \quad \text{where} \quad F_{JI} \equiv \sum_{i \in I} \sum_{j \in J} \pi_j t_{ji} \pi_i \quad (326)$$

Although the Kemeny constant generally decreases when two nodes are combined into one, we can think of combining two clusters iteratively, so that at each step the Kemeny is reduced as little as possible.

#### 4.5.3.1 Kemeny loss from combining clusters

Consider combining two clusters  $A$  and  $B$  into one, called  $\Gamma$ . The Kemeny constant decreases by an amount depending on the choice of clusters to combine.

The "Kemeny loss" can be computed as the difference in  $\zeta^{CG}$  after and before merging the clusters:

$$\begin{aligned} \delta(S_n) &= \frac{F_{AA}}{\Pi_A} + \frac{F_{BB}}{\Pi_B} + \sum_{J \neq A, B} \frac{F_{JJ}}{\Pi_J} \\ \delta(S_{n+1}) &= \frac{F_{\Gamma\Gamma}}{\Pi_A + \Pi_B} + \sum_{J \neq \Gamma} \frac{F_{JJ}}{\Pi_J} \end{aligned} \quad (327)$$

Now, we can compute the loss of Kemeny constant resulting from combining clusters A and B:

$$\zeta^{CG}(n) - \zeta^{CG}(n+1) = \delta(S^{(n)}) - \delta(S^{(n+1)}) \quad (328)$$

Since  $\Gamma = \{A \cup B\}$ , the second terms in Eq. (327) are equal, it simplifies to:

$$\zeta^{CG}(n) - \zeta^{CG}(n+1) = \frac{F_{AA}}{\Pi_A} + \frac{F_{BB}}{\Pi_B} - \frac{F_{\Gamma\Gamma}}{\Pi_A + \Pi_B} \quad (329)$$

Expanding  $F_{\Gamma\Gamma}$  using  $\Gamma = \{A \cup B\}$ :

$$F_{\Gamma\Gamma} = F_{AA} + F_{BA} + F_{AB} + F_{BB} \quad (330)$$

And rearranging, we can rewrite Eq.(329))

$$\zeta^{CG(n)} - \zeta^{CG(n+1)} = \frac{1}{\Pi_A + \Pi_B} \left[ \left( \frac{\Pi_B}{\Pi_A} F_{AA} + \frac{\Pi_A}{\Pi_B} F_{BB} \right) - (F_{BA} + F_{AB}) \right] \quad (331)$$

One could think of starting with no clustering (each node in its own cluster), then compute this "Kemeny loss" for each pair of clusters and combine the two cluster for which this loss is minimal At each iteration we can look for the pair of clusters ( $A, B$ ) for which this quantity is minimal, and then combine them until a target number of clusters is reached.

#### 4.5.3.2 Kemeny gain from moving 1 node from cluster A to cluster B

If we move a single node  $\alpha$  from cluster  $A$  to  $B$ , what will be the Kemeny gain?

Using the formalism introduced in the previous section, we can rewrite Eq. (325) to compute the Kemeny gain from moving (re-assigning) a node  $\alpha$  from the cluster  $A$  to  $B$ .

$$\zeta^{CG} = \zeta^{orig} - \sum_J \frac{F_{JJ}}{\Pi_J} \quad (332)$$

Writing  $A$  and  $B$  for the clusters without node  $\alpha$ , and  $A^+ = \{A \cup \alpha\}$  ( $B^+ = \{B \cup \alpha\}$ ) for the set of states containing the cluster  $A$  ( $B$ ) and the node  $\alpha$ , hence we have:

$$\begin{cases} \Pi_{A^+} = \Pi_A + \pi_\alpha \\ F_{A^+A^+} = \sum_{i,j \in A^+} \pi_j t_{ji} \pi_i \end{cases} \quad (333)$$

Only the contributions of cluster  $J = A$  and  $B$  to the Kemeny constant are relevant, as only these terms change when node  $\alpha$  is moved (re-assigned) from cluster  $A$  to cluster  $B$ .

$$\begin{cases} \delta(S^{(\alpha \in A)}) = C + \frac{F_{BB}}{\Pi_B} + \frac{F_{A^+A^+}}{\Pi_{A^+}} \\ \delta(S^{(\alpha \in B)}) = C + \frac{F_{AA}}{\Pi_A} + \frac{F_{B^+B^+}}{\Pi_{B^+}} \end{cases} \quad (334)$$

The difference in Kemeny between the two configurations is given by the difference between the  $\delta(S)$ :

$$\zeta^{CG}(\alpha \in B) - \zeta^{CG}(\alpha \in A) = \delta(S^{(\alpha \in A)}) - \delta(S^{(\alpha \in B)}) \quad (335)$$

$$\begin{aligned} &= \left( \frac{F_{BB}}{\Pi_B} + \frac{F_{A^+A^+}}{\Pi_{A^+}} \right) - \left( \frac{F_{AA}}{\Pi_A} + \frac{F_{B^+B^+}}{\Pi_{B^+}} \right) \\ &= \left( \frac{F_{BB}}{\Pi_B} - \frac{F_{B^+B^+}}{\Pi_{B^+}} \right) - \left( \frac{F_{AA}}{\Pi_A} - \frac{F_{A^+A^+}}{\Pi_{A^+}} \right) \\ &= \frac{1}{\Pi_B \Pi_{B^+}} (\Pi_{B^+} F_{BB} - \Pi_B F_{B^+B^+}) - \frac{1}{\Pi_{A^+} \Pi_A} (\Pi_A F_{AA} - \Pi_{A^+} F_{A^+A^+}) \\ &= G_B - G_A \quad \text{Defining } G_X \equiv \frac{1}{\Pi_X \Pi_{X^+}} (\Pi_{X^+} F_{XX} - \Pi_X F_{X^+X^+}) \end{aligned} \quad (336)$$

Rewriting Eq. (330) using  $F_{\Gamma\Gamma}$  where  $\Gamma = A^+$ , we have:

$$F_{A^+A^+} = F_{AA} + F_{A\alpha} + F_{\alpha A} + F_{\alpha\alpha} \quad (337)$$

Considering  $G_B$  and expanding  $F_{A^+A^+}$  and  $\Pi_{A^+}$

$$G_B = \frac{1}{\Pi_B \Pi_{B^+}} ((\Pi_B + \pi_\alpha) F_{BB} - \Pi_B (F_{BB} + F_{B\alpha} + F_{\alpha B} + F_{\alpha\alpha})) \quad (338)$$

Simplifying:

$$G_B = \frac{1}{\Pi_B \Pi_{B^+}} (\pi_\alpha F_{BB} - \Pi_B (F_{B\alpha} + F_{\alpha B} + F_{\alpha\alpha}))$$

Rearranging:

$$G_B = \pi_\alpha \left( -\left( \frac{F_{B\alpha}}{\pi_\alpha \Pi_{B^+}} - \frac{F_{BB}}{\Pi_B \Pi_{B^+}} \right) - \left( \frac{F_{\alpha B}}{\pi_\alpha \Pi_{B^+}} + \frac{F_{\alpha\alpha}}{\pi_\alpha \Pi_{B^+}} \right) \right) \quad (339)$$

Using the following formula for the clustered mean first passage times:

$$t_{JI}^{CG} = \frac{1}{\Pi_I \Pi_J} \sum_{i \in I, j \in J} \pi_j \pi_i t_{ji} - \frac{1}{\Pi_J^2} \sum_{i, j \in J} \pi_j \pi_i t_{ji} = \frac{F_{JI}}{\Pi_J \Pi_I} - \frac{F_{JJ}}{\Pi_J^2} \quad (340)$$

and  $F_{\alpha\alpha} = 0$  (from  $t_{\alpha\alpha} = 0$ ), we can simplify Eq. (339) further:

$$G_B = \pi_\alpha \left( -\left( \frac{F_{B\alpha}}{\pi_\alpha \Pi_{B^+}} - \frac{F_{BB}}{\Pi_B \Pi_{B^+}} \right) - \left( \frac{F_{\alpha B}}{\pi_\alpha \Pi_{B^+}} + \frac{F_{\alpha\alpha}}{\pi_\alpha \Pi_{B^+}} \right) \right) \quad (341)$$

From  $t_{\alpha\alpha} = 0$  follows that  $F_{\alpha\alpha} = 0$ . Moreover, under the approximation  $\Pi_B + \pi_\alpha \sim \Pi_B$ , we can use Eq. (340) to simplify Eq. (341) further:

$$G_B = \pi_\alpha \left( -t_{B\alpha}^{CG} - \frac{F_{\alpha B}}{\pi_\alpha \Pi_{B^+}} \right) \quad (342)$$

Then we can notice that

$$t_{\alpha B}^{CG} = \frac{F_{\alpha B}}{\pi_\alpha \Pi_{B^+}} - \frac{F_{\alpha\alpha}}{\pi_\alpha^2} = \frac{F_{\alpha B}}{\pi_\alpha \Pi_{B^+}} \quad (343)$$

Again, using  $F_{aa} = 0$ , leads to

$$G_B = -\pi_\alpha (t_{B\alpha}^{CG} + t_{\alpha B}^{CG}) \quad (344)$$

Similarly, for  $G_A$  we can show that

$$G_A = \pi_\alpha \left( -\left( \frac{F_{A\alpha}}{\pi_\alpha \Pi_{A^+}} - \frac{F_{AA}}{\Pi_A \Pi_{A^+}} \right) - \left( \frac{F_{\alpha A}}{\pi_\alpha \Pi_{A^+}} + \frac{F_{\alpha\alpha}}{\pi_\alpha \Pi_{A^+}} \right) \right) \quad (345)$$

And using again  $F_{\alpha\alpha}$ ,  $\Pi_A + \pi_\alpha \sim \Pi_A$ , and the formula for  $t_{A\alpha}^{CG}$ , simplify to:

$$G_A = -\pi_\alpha (t_{A\alpha}^{CG} + t_{\alpha A}^{CG}) \quad (346)$$

Retrieving the same result as from the 1-D derivation:

$$\zeta^{CG}(\alpha \in B) - \zeta^{CG}(\alpha \in A) = G_B - G_A = \pi_\alpha ((t_{A\alpha}^{CG} + t_{\alpha A}^{CG}) - (t_{B\alpha}^{CG} + t_{\alpha B}^{CG})) \quad (347)$$

This result holds in the general case of any-dimensional networks, as long as  $\pi_\alpha \sim 0$  and  $t_{\alpha\alpha} = 0$   
Defining the mean first round-trip time  $t_{xy}^{RT} = t_{xy}^{CG} + t_{yx}^{CG}$

We find

$$\zeta^{CG}(\alpha \in B) - \zeta^{CG}(\alpha \in A) = \pi_\alpha (t_{A\alpha}^{RT} - t_{B\alpha}^{RT}) \quad (348)$$

NOTE: try sum to get round trip from A to B (should be constant / independent from  $\alpha$ )

$$(t_{Ba} + t_{aB}) + (t_{Aa} + t_{aA}) = (t_{Ba} + t_{aA}) + t_{Aa} + t_{aB} = t_{AB}^{RT} \quad (349)$$

#### 4.5.3.3 Algorithm

We propose the following algorithm:

We start from MFPT  $\mathbf{t}$  or obtain it from rates matrix  $\mathbf{K}$ , and set the assignment matrix  $\mathbf{S}$  to Identity. We use two consecutive loops as two separate steps. In the first step we reduce the number of clusters one by one: we iteratively combine (merge) two clusters, until the target number of clusters (obtained from spectral gap as in PTVC) is reached. In the second step we keep the number of clusters fixed, and try flipping nodes one by one to another cluster they are connected to. We iteratively find all edge (border) nodes (connected to a node belonging to a different cluster) and compute the Kemeny loss  $\delta(S)$  defined in Eq. (326) after trying to flip each edge node to the other cluster it is connected to, one edge node by one, and compare the best flip (for which  $\delta(S)$  is the smallest) to the current  $\delta(S)$ , and if the flip allows to increase the kemeny constant the flip is performed. This second step is necessary to fully "converge to the local minimum".

---

**Algorithm 1** First algorithm for Louvain-like Kemeny maximization

---

**Require:**  $\mathbf{t}$  and  $\boldsymbol{\pi}$  ▷  $\mathbf{t}$  can be obtained from  $\mathbf{K}$

$\mathbf{S} \leftarrow I_n$   
 $\mathbf{W} \leftarrow \boldsymbol{\pi}^T \mathbf{t} \boldsymbol{\pi}$   
 $N_{target} \leftarrow \text{Spectral decomp}(\mathbf{K})$   
 $N_{cluster} \leftarrow N_{nodes}(\mathbf{t})$   
**while**  $N_{cluster} > N_{target}$  **do** ▷ Step 1: Iteratively combining clusters  
     $\hat{\boldsymbol{\pi}} \leftarrow \mathbf{S} \boldsymbol{\pi}$   
     $\mathbf{F} \leftarrow \mathbf{S}^T \mathbf{W} \mathbf{S}$   
     $\mathbf{d}\zeta_{JI} \leftarrow \frac{1}{\hat{\pi}_I + \hat{\pi}_J} \left[ (\frac{\hat{\pi}_J}{\hat{\pi}_I} F_{II} + \frac{\hat{\pi}_I}{\hat{\pi}_J} F_{JJ}) - (F_{JI} + F_{IJ}) \right]$   $\forall (I, J)$  ▷ Compute Kemeny loss for combining  $(I, J)$   
     $(\mathbf{I}^*, \mathbf{J}^*) \leftarrow \min_{(I, J)} \mathbf{d}\zeta_{JI}$  ▷ Find pair  $(\mathbf{I}^*, \mathbf{J}^*)$  with the smallest value of  $\mathbf{d}\zeta_{JI}$   
     $\mathbf{S}^* \leftarrow \mathbf{S}$   
     $\mathbf{S}^*(\cdot, J^*) \leftarrow \mathbf{S}(\cdot, J^*) + \mathbf{S}(\cdot, I^*)$  ▷ Combine clusters  $J^*$  and  $I^*$  into  $J^*$   
     $\mathbf{S}^*(\cdot, I^*) \leftarrow []$   
     $\mathbf{S} \leftarrow \mathbf{S}^*$   
**end while**  
**while**  $min_{val} < 0$  **do** ▷ Step 2: Fine tuning by flipping border nodes  
     $\hat{\boldsymbol{\pi}} \leftarrow \mathbf{S} \boldsymbol{\pi}$   
     $\mathbf{F} \leftarrow \mathbf{S}^T \mathbf{W} \mathbf{S}$   
     $\delta(\mathbf{S}) \leftarrow \sum_J \frac{F_{JJ}}{\hat{\pi}_J}$   
     $\mathbf{L} \leftarrow \text{Set of all Edge Nodes}$  ▷ Find all "edge nodes"  
    **for**  $l$  **in**  $\mathbf{L}$  **do**  
         $C_l \leftarrow$  The cluster node  $l$  is currently assigned to  
         $C_o \leftarrow$  A cluster node  $l$  is connected to ▷  $C_o \neq C_l$ , if multiple choices for  $C_o$ , select one randomly  
         $\mathbf{S}^*(l) \leftarrow$  modified  $S$  with  $l$  assigned to  $C_o$  instead of  $C_l$   
         $\hat{\boldsymbol{\pi}}^* \leftarrow \mathbf{S}^* \boldsymbol{\pi}$   
         $\mathbf{F}^* \leftarrow \mathbf{S}^{*T} \mathbf{W} \mathbf{S}^*$   
         $\delta(\mathbf{S}^*(l)) \leftarrow \sum_J \frac{F_{JJ}^*}{\hat{\pi}_J}$   
    **end for**  
     $l^* \leftarrow \min_l(\delta(\mathbf{S}^*(l)))$  ▷ Find  $l^*$  with smallest value of  $\delta\zeta_{new}(l)$   
    **if**  $\delta(\mathbf{S}^*(l)) < \delta(\mathbf{S})$  **then**  
         $\mathbf{S} \leftarrow \mathbf{S}^*(l^*)$  ▷ Flip node  $l$  to other cluster  
    **else**  
        Leave for loop  
    **end if**  
**end while**


---

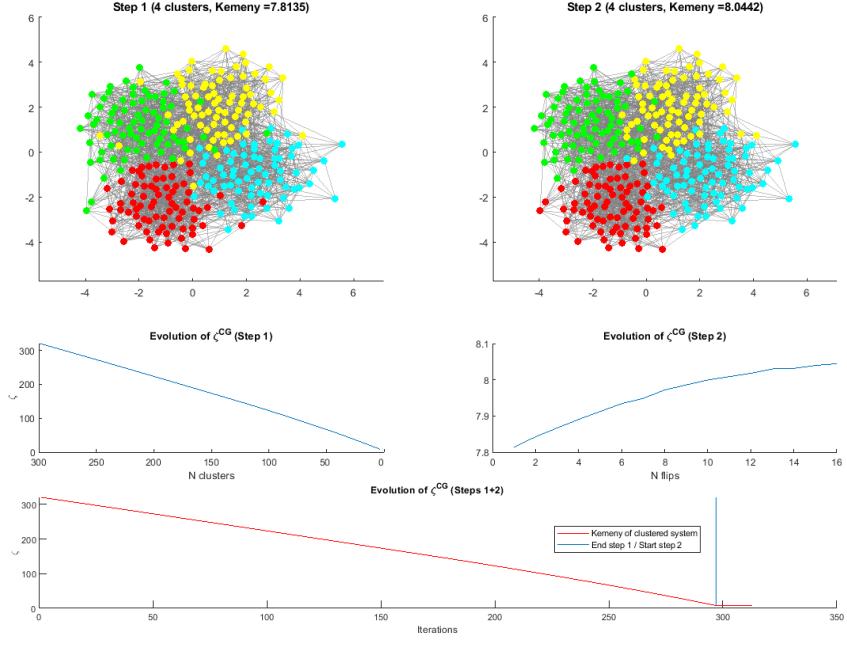


Figure 70: Top: Clustering of a 4-states SBM with 100 nodes, generated with inter- and intra-cluster edge probabilities of 0.1 and 0.7 respectively, into 4 clusters using the presented algorithm. Bottom: Evolution of the Kemeny constant of the clustered network along the simulation

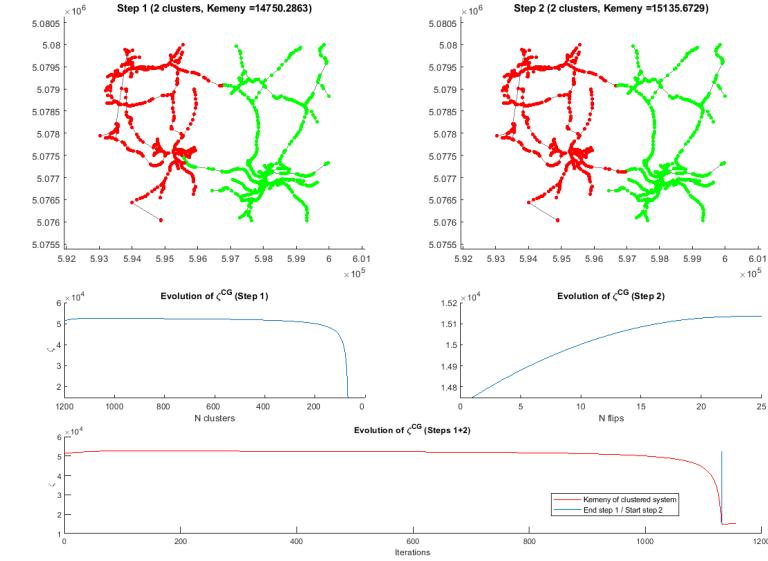


Figure 71: Top: Clustering of the subset of Luxembourg streets network (1132 nodes) into two clusters using the presented algorithm. Bottom: Evolution of the Kemeny constant of the clustered network along the simulation

#### 4.5.3.4 Results

Preliminary tests have shown a good potential to this approach, it seems to work faster than PTVC and finds similar results.

For SBMs, in Step 1 the Kemeny constant decreases linearly, whereas Step 2 increases it with decreasing increments. For the Luxembourg roads network, in Step 1 the kemeny constant decreases more and more quickly, whereas in Step 2 it behaves similarly to SBMs.

The SBM network have been clustered almost perfectly: only one node have been misassigned (not shown here).

#### 4.5.4 Conclusion

Leveraging on the Variational result for the Kemeny constant, and inspiring from the Louvain modularity maximization method, we propose a very efficient algorithm to cluster kinetic networks based on maximizing the Kemeny constant. We performed preliminary tests on SBM random networks and a real Road network and

observed that it was faster than the PTVC algorithm by a few order of magnitude, and provides similar results.

## 5 X - Plan and timetable for remainder work

Finalize the comparison of all new biasing strategies. Maybe try to publish a few papers about the methods performing particularly well.

## A APPENDIX - Dynamical Systems / Kinetic Networks

We start with Eq. (61) and multiply left and right hand sides times  $\mathbf{t}$ , from left, and times  $\mathbf{D}_n$  from right

$$\mathbf{tKD}_n = \mathbf{t}\pi\mathbf{1}_n^T\mathbf{D}_n - [\mathbf{D}_n^{-1}\pi\mathbf{1}_n^T\mathbf{t}^{-1} - \mathbf{t}(\mathbf{I} - \pi\mathbf{1}_n^T)\mathbf{t}^{-1}]^{-1} \quad (350)$$

Using Eq. (62) and  $\mathbf{D}_n^{-1}\pi = \mathbf{1}_n$ , we get

$$\begin{aligned} \mathbf{tKD}_n &= \mathbf{t}\pi[\pi]^T - \left[ \frac{1}{\zeta} \mathbf{1}_n[\pi]^T - \left( \mathbf{I} - \frac{1}{\zeta} \mathbf{t}\pi[\pi]^T \right) \right]^{-1} \\ &= \mathbf{t}\pi[\pi]^T + \left[ \mathbf{I} - \frac{1}{\zeta} (\mathbf{1}_n + \mathbf{t}\pi)[\pi]^T \right]^{-1} \end{aligned} \quad (351)$$

Upon using the Sherman-Morrison formula,

$$(\mathbf{I} + \mathbf{uv}^T)^{-1} = \mathbf{I} - \frac{\mathbf{uv}^T}{1 + \mathbf{u}^T\mathbf{v}} \quad (352)$$

(62), and  $\mathbf{p}^T\mathbf{1}_n = 1$ , we find

$$\mathbf{tKD}_n = -\mathbf{1}_n\mathbf{p}^T + \mathbf{I} \quad (353)$$

from which Eq. (64) follows.

## B APPENDIX 2 - Enhanced Sampling

Here we will argue that the constraint  $k_{zz} = -\sum_{j \neq z} k_{zj}$  does not required to be introduced in the likelihood with a lagrange multiplier, and can be implemented by directly replacing  $k_{zz}$  by  $-\sum_{j \neq z} k_{zj}$ . One could argue that the relation  $k_{zz} = -\sum_{j \neq z} k_{zj}$  should be added as a separate constraint. Let's consider setting  $k_{zz} = -\sum_{j \neq z} k_{zj}$  as a separate constraint. We can define the constraint  $C_\alpha$

$$C_\alpha \equiv \sum_z \alpha_z \left[ k_{zz} + \sum_{j \neq z} k_{zj} \right]$$

So that the log-Likelihood, in the case of a non-biased simulation, becomes

$$\log L = \sum_i \sum_{j \neq i} n_{ij}^\alpha \log(k_{ij}) + \sum_z k_{zz} t_z - C_\alpha \quad (354)$$

The partial derivatives become:

$$\frac{\partial \log L}{\partial k_{i''j''}} = \frac{n_{i''j''}}{k_{i''j''}} - \alpha_{i''}$$

$$\frac{\partial \log L}{\partial \alpha_{z''}} = -(k_{z''z''} + \sum_{j \neq z''} k_{z''j''})$$

$$\frac{\partial \log L}{\partial k_{z''z''}} = t_{z''} - \alpha_{z''}$$

We see that when formalising the constraint in this way, deriving with respect to  $k_{z''z''}$  will always give us the relation  $t_{z''} = \alpha_{z''}$ , and the relation given by deriving with respect to  $k_{z''z''}$  will remain unused.

This will stay unchanged with the introduction of constraints for Detailed Balance, because those constraints do not relate to the diagonal elements  $k_{z''z''}$ .

For this reason, we drop the constraint  $C_\alpha$ , to replace directly the terms  $k_{zz}$  by  $-\sum_{j \neq z} k_{zj}$ .

This way we are explicitly optimizing with the  $N(N-1)$  off-diagonal entries of the rates matrix as parameters, the values of  $k_{ij}$  where  $i \neq j$

## C APPENDIX 3 - Clustering

### C.1 Condition on the Coarse Grained MFPTs.

Consider a matrix  $\tau$  satisfying

$$[\hat{\mathbf{p}}^{\text{eq}}]^T \tau = \phi \mathbf{1}_N^T \quad (355)$$

where  $\phi$  is an arbitrary parameter. First we prove that any vector  $\hat{\tau} = \tau - \mathbf{w} \mathbf{1}_N^T$ , where  $\mathbf{w}$  is an arbitrary vector, will satisfy

$$[\hat{\mathbf{p}}^{\text{eq}}]^T = \frac{\mathbf{1}_N^T \hat{\tau}^{-1}}{\mathbf{1}_N^T \hat{\tau}^{-1} \mathbf{1}_N} \quad (356)$$

By using the Shermann-Morrison formula (352)

$$\hat{\tau}^{-1} = \tau^{-1} + \frac{1}{\phi} \frac{\tau^{-1} \mathbf{w} \hat{\mathbf{p}}^{\text{eq}}^T}{1 - \frac{1}{\phi} \hat{\mathbf{p}}^{\text{eq}}^T \mathbf{w}} \quad (357)$$

This gives

$$\mathbf{1}_N^T \hat{\tau}^{-1} = \frac{1}{\phi - \hat{\mathbf{p}}^{\text{eq}}^T \mathbf{w}} [\hat{\mathbf{p}}^{\text{eq}}]^T \quad (358)$$

and  $\mathbf{1}_N^T \hat{\tau}^{-1} \mathbf{1}_N = (\phi - [\hat{\mathbf{p}}^{\text{eq}}]^T \mathbf{w})^{-1}$ . Secondly, one can show that

$$\hat{\tau}^{-1} (\mathbf{D}_N^{-1} - \mathbf{1}_N \mathbf{1}_N^T) = \tau^{-1} (\mathbf{D}_N^{-1} - \mathbf{1}_N \mathbf{1}_N^T) \quad (359)$$

A choice of  $\tau$  that satisfies (355) for  $\hat{\mathbf{p}}^{\text{eq}} = \mathbf{A}^T \mathbf{p}^{\text{eq}}$  is

$$\tau = \frac{\phi}{\zeta} \mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}. \quad (360)$$

Since subtraction of any matrix  $\mathbf{w} \mathbf{1}_N^T$  will lead to the same equilibrium distribution and rate matrix, one can define the MFPTs  $\hat{\mathbf{t}}$  in the coarse grained system as

$$\hat{\mathbf{t}} = \tau - \frac{\phi}{\zeta} \mathbf{z} \mathbf{1}_N^T$$

with  $z_J = (\mathbf{D}_N^{-1} \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JJ}$ . This definition is guaranteed to satisfy  $\hat{t}_{JJ} = 0 \forall J$ , in addition to  $\hat{\mathbf{p}}^{\text{eq}} = \mathbf{1}_N^T \hat{\mathbf{t}}^{-1} / \mathbf{1}_N^T \hat{\mathbf{t}}^{-1} \mathbf{1}_N$ . Setting  $\phi = \zeta$  corresponds to the choice (291). For this choice, the resulting Kemeny constant in the coarse grained system is  $\hat{\zeta} \equiv 1/\mathbf{1}_N^T \hat{\mathbf{t}}^{-1} \mathbf{1}_N = \zeta - [\hat{\mathbf{p}}^{\text{eq}}]^T \hat{\mathbf{w}}$ , and the variational principle  $\hat{\zeta} \leq \zeta$  holds. We show in (C.3) that this choice corresponds to the coarse graining protocol defined by Hummer and Szabo in [62].

However, other choices are possible. In particular, it is possible to choose  $\phi$  in such a way that the sum of relaxation times in the clustered and in the original system are the same  $\hat{\zeta} = \zeta$ , by requiring

$$\hat{\zeta} \equiv \phi - [\hat{\mathbf{p}}^{\text{eq}}]^T \mathbf{w} = \zeta \quad (361)$$

with  $\mathbf{w} = \mathbf{z} \phi / \zeta$ . This leads to

$$\phi = \frac{\zeta}{1 - \frac{1}{\zeta} [\hat{\mathbf{p}}^{\text{eq}}]^T \mathbf{z}} \quad (362)$$

and results in the choice, for the MFPTs in the clustered system, given in (297).

### C.2 Derivation of Eq. (296)

Multiplying (291) times  $\hat{p}_J^{\text{eq}}$ , we can make use of (48) to express the MFPT in the coarse-grained system, in terms of the coarse grained rate matrix, to get

$$\begin{aligned} & (\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JI} - (\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JJ} \\ &= (\mathbf{P}^{\text{eq}} \mathbf{1}_N^T - \hat{\mathbf{K}})^{-1}_{JJ} - (\mathbf{P}^{\text{eq}} \mathbf{1}_N^T - \hat{\mathbf{K}})^{-1}_{JI} \end{aligned} \quad (363)$$

Defining  $u_J = (\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1})_{JJ}$  and  $v_J = (\mathbf{P}^{\text{eq}} \mathbf{1}_N^T - \mathbf{R})_{JJ}^{-1}$ , equation (363) can be written in matrix form and rearranged to yield an expression for the reduced rate matrix

$$\mathbf{R} = \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - [(\mathbf{v} + \mathbf{u}) \mathbf{1}_N^T - \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}]^{-1}. \quad (364)$$

The vector  $\mathbf{v}$  can be determined by demanding that  $\mathbf{P}^{\text{eq}}$  is the steady state of the dynamics described by  $\mathbf{R}$ , i.e.  $\mathbf{R}\mathbf{P}^{\text{eq}} = 0$ . Using  $\mathbf{1}_N^T \mathbf{P}^{\text{eq}} = 1$ ,  $\mathbf{D}_N^{-1} \mathbf{P}^{\text{eq}} = \mathbf{1}_N$ ,  $\mathbf{A}\mathbf{1}_N = \mathbf{1}_n$  and  $\mathbf{D}_n\mathbf{1}_n = \mathbf{p}_n^{\text{eq}}$ , as well as that an invertible matrix has the same eigenvectors as its inverse (with inverse eigenvalues), we get  $\mathbf{v} = \mathbf{P}^{\text{eq}} - \mathbf{u} + \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{p}^{\text{eq}}$ . Substituting (C.2) in (364) this finally gives (296). We check below that (296) automatically satisfies also the condition 0. By multiplying the above equation times  $\mathbf{1}_N^T$  from left and equating to zero, we get

$$\zeta \mathbf{1}_N^T = \mathbf{1}_n^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \quad (365)$$

where we have used  $\mathbf{1}_N^T \mathbf{A}^T = \mathbf{1}_n^T$ ,  $\mathbf{1}_n^T \mathbf{D}_n = [\mathbf{p}^{\text{eq}}]^T$  and (62). Substituting (58) into the above equation

$$\begin{aligned} \zeta \mathbf{1}_N^T &= \mathbf{1}_n^T \mathbf{z} \mathbf{1}_n^T \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \\ &\quad - \mathbf{1}_n^T (\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}, \end{aligned} \quad (366)$$

and using  $\mathbf{1}_n^T \mathbf{z} = (1 + \zeta) \mathbf{1}_N^T$ ,  $\mathbf{1}_n^T (\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} = \mathbf{1}_n^T$ ,  $\mathbf{1}_n^T \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} = \mathbf{1}_N^T$ , and  $\mathbf{1}_n^T \mathbf{p}^{\text{eq}} = 1$ , shows that (366) is identically satisfied.

Finally, it is straightforward to show that (296) coincides with (295). Upon multiplying (296), from left, times  $[\mathbf{P}^{\text{eq}} \mathbf{1}_N^T + \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{p}^{\text{eq}} \mathbf{1}_N^T - \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}]$ , and using  $\mathbf{1}_N^T \mathbf{R} = 0$ , we get

$$\begin{aligned} -\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{R} &= [\mathbf{P}^{\text{eq}} \mathbf{1}_N^T + \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{p}^{\text{eq}} \mathbf{1}_N^T] \\ &\quad - \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - \mathbf{I}. \end{aligned} \quad (367)$$

Expanding the product on the r.h.s.,

$$\begin{aligned} -\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{R} &= \mathbf{P}^{\text{eq}} \mathbf{1}_N^T + \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{p}^{\text{eq}} \mathbf{1}_N^T \\ &\quad - \mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - \mathbf{I}. \end{aligned} \quad (368)$$

and using  $\mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{P}^{\text{eq}} \mathbf{1}_N^T = \mathbf{D}_n \mathbf{A} \mathbf{1}_N \mathbf{1}_N^T = \mathbf{D}_n \mathbf{1}_n \mathbf{1}_N^T = \mathbf{p}^{\text{eq}} \mathbf{1}_N^T$ , the second and third term on the r.h.s. cancel, leading to

$$\mathbf{A}^T \mathbf{D}_n \mathbf{t} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \mathbf{R} = \mathbf{I} - \mathbf{P}^{\text{eq}} \mathbf{1}_N^T \quad (369)$$

and thus to (295).

### C.3 Retrieval of Hummer-Szabo Coarse Graining

In this section we show that the proposed coarse graining, based on equation (291), coincides with the one proposed by Hummer and Szabo in [62], which equates the areas underneath the correlation functions

$$\sum_{i \in I, j \in J} \int_0^\infty dt C_{ij}(t) = \int_0^\infty dt C_{IJ}(t) \quad (370)$$

By inserting (58) in (296), we have

$$\begin{aligned} \mathbf{R} &= \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - [\mathbf{P}^{\text{eq}} \mathbf{1}_N^T + \mathbf{A}^T \mathbf{z} \mathbf{1}_n^T \mathbf{p}^{\text{eq}} \mathbf{1}_N^T] \\ &\quad - \mathbf{A}^T (\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} \mathbf{p}^{\text{eq}} \mathbf{1}_N^T - \mathbf{A}^T \mathbf{z} \mathbf{1}_n^T \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} \\ &\quad + \mathbf{A}^T (\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}]^{-1} \end{aligned} \quad (371)$$

Using  $(\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} \mathbf{p}^{\text{eq}} = \mathbf{p}^{\text{eq}}$ ,  $\mathbf{A}^T \mathbf{p}^{\text{eq}} = \mathbf{P}^{\text{eq}}$  and  $\mathbf{1}_n^T \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1} = \mathbf{1}_N^T$  this simplifies to

$$\mathbf{R} = \mathbf{P}^{\text{eq}} \mathbf{1}_N^T - [\mathbf{A}^T (\mathbf{p}^{\text{eq}} \mathbf{1}_n^T - \mathbf{K})^{-1} \mathbf{D}_n \mathbf{A} \mathbf{D}_N^{-1}]^{-1} \quad (372)$$

which coincides with the expression derived by Hummer-Szabo by imposing (370). In contrast to (296), this formulation requires the inversion of a large dimensional matrix, hence (296) may be computationally more efficient when MFPTs and equilibrium distribution are known.

The partial derivatives become:

$$\frac{\partial \log L}{\partial k_{i''j''}} = \frac{n_{i''j''}}{k_{i''j''}} - \alpha_{i''}$$

$$\frac{\partial \log L}{\partial \alpha_{z''}} = -(k_{z''z''} + \sum_{j \neq z''} k_{z''j})$$

$$\frac{\partial \log L}{\partial k_{z''z''}} = t_{z''} - \alpha_{z''}$$

We see that when formalizing the constraint in this way, deriving with respect to  $k_{z''z''}$  will always give us the relation  $t_{z''} = \alpha_{z''}$ , and the relation given by deriving with respect to  $k_{z''z''}$  will remain unused.

This will stay unchanged with the introduction of constraints for Detailed Balance, because those constraints do not relate to the diagonal elements  $k_{z''z''}$ .

For this reason, we drop the constraint  $C_\alpha$ , to replace directly the terms  $k_{zz}$  by  $-\sum_{j \neq z} k_{zj}$ .

This way we are explicitly optimizing with the  $N(N-1)$  off-diagonal entries of the rates matrix as parameters, the values of  $k_{ij}$  where  $i \neq j$

#### C.4 Diffusion on 1D Symmetric potential: 3-state clustering

In this section we consider a system diffusing in a 1D potential symmetric about 0. We consider clustering the system in three clusters, that we shall denote with 1, 2, 3. Taking 0 as the center of the 1D space, we denote with  $a$  the position of the boundary between cluster 2 and 3 and, due to the symmetry of the potential,  $-a$  will denote the boundary between 1 and 2.

The clustered mean first passage times are given in Eq. (308) and can be written compactly as

$$t_{JI}^{CG} = \frac{1}{\Pi_J} \int dx \pi(x) [\bar{t}_{xI} - \bar{t}_{xJ}] \quad (373)$$

in terms of the MFPTs to the single position  $x$  from the clusters  $I$  and  $J$ , respectively, as defined in Eq. (310).

Inserting this into the expression for the Kemeny constant provided for a symmetric system coarse-grained into three clusters, in Eq. (307), we obtain:

$$\zeta^{CG}/2 = \int_{-\infty}^{-a} dx \pi(x) (\bar{t}_{x2} - \bar{t}_{x1}) \quad (374)$$

Differentiating with respect to  $a$  using the Leibniz integral rule yields

$$\begin{aligned} \frac{\partial \zeta^{CG}/2}{\partial a} &= -\pi(-a)(\bar{t}_{-a2} - \bar{t}_{-a1}) \\ &\quad + \int_{-\infty}^{-a} dx \pi(x) \frac{\partial}{\partial a} (\bar{t}_{x2} - \bar{t}_{x1}) \end{aligned} \quad (375)$$

From Eq. (310) we have

$$\bar{t}_{x2} = \int_{-a}^a dz p_2(z) t_{xz} \quad (376)$$

with

$$p_2(x) = \frac{\pi(x)}{\Pi_2}, \quad \Pi_2 = \int_{-a}^a dx \pi(x) \quad (377)$$

hence

$$\begin{aligned} \frac{\partial}{\partial a} \bar{t}_{x2} &= p_2(a) t_{xa} + p_2(-a) t_{x-a} \\ &\quad + \int_{-a}^a dz \frac{\partial}{\partial a} p_2(z) t_{xz} \end{aligned} \quad (378)$$

Substituting

$$\begin{aligned} \frac{\partial}{\partial a} p_2(z) &= \frac{\partial}{\partial a} \frac{\pi(z)}{\Pi_2(a)} = -\frac{\pi(z)}{\Pi_2^2(a)} \frac{\partial}{\partial a} \Pi_2(a) \\ &= -\frac{p_2(z)}{\Pi_2(a)} \frac{\partial}{\partial a} \Pi_2(a) \\ &= -[p_2(a) + p_2(-a)] p_2(z) \end{aligned} \quad (379)$$

we get

$$\begin{aligned} \frac{\partial}{\partial a} \bar{t}_{x2} &= p_2(a) t_{xa} + p_2(-a) t_{x-a} \\ &\quad - [p_2(a) + p_2(-a)] \bar{t}_{x2} \end{aligned} \quad (380)$$

Similarly, for  $\bar{t}_{x1} = \int_{-\infty}^{-a} dz p_1(z) t_{xz}$

$$\frac{\partial}{\partial a} \bar{t}_{x1} = p_1(-a) (\bar{t}_{x1} - t_{x-a}) \quad (381)$$

Collecting all the terms, and using  $p_2(a)\pi(x) = p_2(x)\pi(a)$  and similarly for  $-a$

$$\begin{aligned}
\frac{\partial}{\partial a} \zeta^{CG}/2 &= -\pi(-a)(\bar{t}_{-a2} - \bar{t}_{-a1}) + \pi(a) \int_{-\infty}^{-a} dx p_2(x) t_{xa} \\
&\quad + \pi(-a) \int_{-\infty}^{-a} dx p_2(x) t_{x-a} \\
&\quad - [\pi(a) + \pi(-a)] \int_{-\infty}^{-a} dx p_2(x) \bar{t}_{x2} \\
&\quad - \pi(-a) \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a})
\end{aligned} \tag{382}$$

Using  $\pi(a) = \pi(-a)$ , due to the symmetry of the potential, and that  $\forall x \in 1 \bar{t}_{x2} = t_{x-a} + \bar{t}_{-a2}$  and  $t_{xa} = t_{x-a} + t_{-aa}$

$$\begin{aligned}
\frac{1}{\pi(a)} \frac{\partial}{\partial a} \zeta^{CG}/2 &= -(\bar{t}_{-a2} - \bar{t}_{-a1}) \\
&\quad + \int_{-\infty}^{-a} dx p_2(x) (t_{x-a} + t_{-aa}) \\
&\quad + \int_{-\infty}^{-a} dx p_2(x) t_{x-a} \\
&\quad - 2 \int_{-\infty}^{-a} dx p_2(x) (t_{x-a} + \bar{t}_{-a2}) \\
&\quad - \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a})
\end{aligned}$$

Simplifying, and using  $\Pi_1 = \Pi_3$  due to the symmetry of the potential,

$$\begin{aligned}
\frac{1}{\pi(a)} \frac{\partial}{\partial a} \zeta^{CG}/2 &= -\bar{t}_{-a2} + \bar{t}_{-a1} + \frac{\Pi_1}{\Pi_2} t_{-aa} - 2 \frac{\Pi_1}{\Pi_2} \bar{t}_{-a2} \\
&\quad - \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a}) \\
&= - \left( 1 + 2 \frac{\Pi_1}{\Pi_2} \right) \bar{t}_{-a2} + \bar{t}_{-a1} + \frac{\Pi_1}{\Pi_2} t_{-aa} \\
&\quad - \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a}) \\
&= -\frac{1}{\Pi_2} \bar{t}_{-a2} + \bar{t}_{-a1} + \frac{\Pi_1}{\Pi_2} t_{-aa} \\
&\quad - \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a})
\end{aligned}$$

Equating  $\partial \zeta^{CG}/\partial a$  to zero we obtain

$$-\frac{1}{\Pi_2} \bar{t}_{-a2} + \bar{t}_{-a1} + \frac{\Pi_1}{\Pi_2} t_{-aa} = \int_{-\infty}^{-a} dx p_1(x) (\bar{t}_{x1} - t_{x-a})$$

Because of symmetry, this is equivalent to

$$-\frac{1}{\Pi_2} \bar{t}_{a2} + \bar{t}_{a3} + \frac{\Pi_3}{\Pi_2} t_{-aa} = \int_{-\infty}^{-a} dx p_1(x) [\bar{t}_{x1} - t_{x-a}]$$

One can write the right hand side (RHS) in terms of the Kemeny constant

$$\begin{aligned}
\zeta^{CG}/2 &= \int_{-\infty}^{-a} dx \pi(x)(\bar{t}_{x2} - \bar{t}_{x1}) \\
&= \Pi_1 \int_{-\infty}^{-a} dx p_1(x)[\bar{t}_{x2} - \bar{t}_{x1}] \\
&= \Pi_1 \int_{-\infty}^{-a} dx p_1(x)[(t_{x-a} + \bar{t}_{-a2}) - \bar{t}_{x1}] \\
&= \Pi_1 \int_{-\infty}^{-a} dx p_1(x)[t_{x-a} - \bar{t}_{x1}] + \Pi_1 \bar{t}_{-a2}
\end{aligned} \tag{383}$$

to get

$$-\frac{1}{\Pi_2} \bar{t}_{a2} + \bar{t}_{a3} + \frac{\Pi_3}{\Pi_2} t_{-aa} = \bar{t}_{-a2} - \frac{\zeta^{CG}}{2\Pi_1} = \bar{t}_{-a2} - t_{12}^{CG} \tag{384}$$

where in the last step we used Eq. (307), or, using equivalence of states 1 and 3,

$$\Pi_1 t_{-aa} - \bar{t}_{a2} + \Pi_2 (\bar{t}_{-a1} - \bar{t}_{-a2} + t_{12}^{CG}) = 0 \tag{385}$$

Another identity can be obtained by using the relation of the quantities above to the Kemeny constant of the original process  $\zeta = \int_{-\infty}^{\infty} dx \pi(x) t_{xy}$ . Writing

$$\begin{aligned}
\frac{1}{\Pi_1} \int_{-\infty}^{-a} dx \pi(x) \bar{t}_{x1} &= \frac{1}{\Pi_1^2} \int_{-\infty}^{-a} dx \pi(x) \int_{-\infty}^{-a} dy \pi(y) t_{xy} \\
&= \frac{1}{\Pi_1^2} \int_{-\infty}^{-a} dy \pi(y) \left( \zeta - \int_{-a}^{\infty} dx \pi(x) t_{xy} \right) \\
&= \frac{1}{\Pi_1} \left( \zeta - \int_{-a}^{\infty} dx \pi(x) \bar{t}_{x1} \right)
\end{aligned} \tag{386}$$

using Eq. (373) and  $\bar{t}_{x2} = t_{x-a} + \bar{t}_{-a2} \forall x \in 1$ , we obtain

$$\begin{aligned}
t_{12}^{CG} &= \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} (t_{x-a} + \bar{t}_{-a2} - \bar{t}_{x1}) \\
&= \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} t_{x-a} + \bar{t}_{-a2} - \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} \bar{t}_{x1} \\
&= \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} t_{x-a} + \bar{t}_{-a2} \\
&\quad - \frac{1}{\Pi_1} \left[ \zeta - \int_{-a}^{+\infty} dx \pi(x) \bar{t}_{x1} \right] \\
&= \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} t_{x-a} + \bar{t}_{-a2} \\
&\quad - \frac{1}{\Pi_1} \left[ \zeta^{CG} - \int_{-a}^{+\infty} dx \pi(x) (t_{x-a} + \bar{t}_{-a1}) \right] \\
&= \int_{-\infty}^{-a} dx \frac{\pi(x)}{\Pi_1} t_{x-a} + \bar{t}_{-a2} \\
&\quad - \frac{1}{\Pi_1} \left[ \int_{-\infty}^{-a} dx \pi(x) t_{x-a} - (1 - \Pi_1) \bar{t}_{-a1} \right] \\
&= \bar{t}_{-a2} + \frac{1 - \Pi_1}{\Pi_1} \bar{t}_{-a1}
\end{aligned}$$

Substituting in Eq. (385), one finally obtains a relation between MFPTs and the stationary distribution of the coarse-grained that holds when the Kemeny constant is maximized with respect to the two boundary positions  $-a$  and  $a$ .

$$\Pi_1 t_{-aa} - \bar{t}_{a2} + \frac{\Pi_2}{\Pi_1} \bar{t}_{-a1} = 0 \tag{387}$$

This expression will be tested in Sec. (4.3.5.1).

A more interpretable result can be derived as follows. Defining

$$\hat{t}_{J\alpha} = \int_J dx \frac{\pi(x)}{\Pi_J} (t_{x\alpha} - \bar{t}_{xJ}) \quad (388)$$

and using Eq. (307) and (383) we can interpret  $\hat{t}_{1-a} = -\bar{t}_{-a2} + t_{12}^{CG}$  as the MFPT to cluster 1 from its right boundary. Then rearranging Eq. (382) we have

$$\begin{aligned} \frac{\partial}{\partial a} \zeta^{CG}/2 &= -\pi(-a)(\bar{t}_{-a2} - \bar{t}_{-a1}) \\ &\quad + \pi(a) \int_{-\infty}^{-a} dx p_2(x) (t_{xa} - \bar{t}_{x2}) \\ &\quad + \pi(-a) \int_{-\infty}^{-a} dx p_2(x) (t_{x-a} - \bar{t}_{x2}) \\ &\quad + \pi(-a)(t_{12}^{CG} - \bar{t}_{-a2}) \end{aligned} \quad (389)$$

Using the symmetry of the potential, we can expand the above equation as follows (this corresponds to starting the derivation from  $\zeta^{CG} = \Pi_1 t_{12}^{CG} + \Pi_3 t_{32}^{CG}$  instead of  $\zeta^{CG} = 2\Pi_1 t_{12}^{CG}$ ):

$$\begin{aligned} \frac{\partial}{\partial a} \zeta^{CG} &= -\pi(-a)(\bar{t}_{-a2} - \bar{t}_{-a1}) \\ &\quad + \pi(a) \int_{-\infty}^{-a} dx p_2(x) (t_{xa} - \bar{t}_{x2}) \\ &\quad + \pi(-a) \int_{-\infty}^{-a} dx p_2(x) (t_{x-a} - \bar{t}_{x2}) \\ &\quad + \pi(-a)(t_{12}^{CG} - \bar{t}_{-a2}) \\ &\quad - \pi(a)(\bar{t}_{a2} - \bar{t}_{a3}) \\ &\quad + \pi(a) \int_a^{+\infty} dx p_2(x) (t_{xa} - \bar{t}_{x2}) \\ &\quad + \pi(-a) \int_a^{+\infty} dx p_2(x) (t_{x-a} - \bar{t}_{x2}) \\ &\quad + \pi(a)(t_{32}^{CG} - \bar{t}_{a2}) \end{aligned} \quad (390)$$

Writing

$$\begin{aligned}
\frac{\partial \Pi_2 t_{22}^{CG}}{\partial a} &= \pi(a)(\bar{t}_{a2} - \bar{t}_{a2}) - \pi(-a)(\bar{t}_{-a2} - \bar{t}_{-a2}) \\
&\quad + \int_{-a}^a dx \pi(x) \left[ \frac{\partial \bar{t}_{x2}}{\partial a} - \frac{\partial \bar{t}_{x2}}{\partial a} \right] \\
&= \int_{-a}^a dx \pi(x) \left\{ p_2(a)t_{xa} + p_2(-a)t_{x-a} \right. \\
&\quad \left. - [p_2(a) + p_2(-a)]\bar{t}_{x2} - p_2(a)t_{xa} - p_2(-a)t_{x-a} \right. \\
&\quad \left. + [p_2(a) + p_2(-a)]\bar{t}_{x2} \right\} \\
&= \int_{-a}^a dx \pi(x)p_2(a)(t_{xa} - \bar{t}_{x2} - t_{xa}) \\
&\quad + \int_{-a}^a dx \pi(x)p_2(-a)(t_{x-a} - \bar{t}_{x2} - t_{x-a}) \\
&\quad + \int_{-a}^a dx \pi(x)[p_2(a) + p_2(-a)]\bar{t}_{x2} \\
&= \int_{-a}^a dx \pi(a)p_2(x)(t_{xa} - \bar{t}_{x2}) \\
&\quad + \int_{-a}^a dx \pi(-a)p_2(x)(t_{x-a} - \bar{t}_{x2}) \\
&\quad + \pi(a) \int_{-a}^a dx p_2(x)(\bar{t}_{x2} - t_{xa}) \\
&\quad + \pi(-a) \int_{-a}^a dx p_2(x)(\bar{t}_{x2} - t_{x-a}) \\
&= -\pi(a)\hat{t}_{2a} - \pi(-a)\hat{t}_{2-a} \\
&\quad + \pi(a) \int_{-a}^a dx p_2(x)(t_{xa} - \bar{t}_{x2}) \\
&\quad + \pi(-a) \int_{-a}^a dx p_2(x)(t_{x-a} - \bar{t}_{x2})
\end{aligned} \tag{391}$$

We now add Eq. (391) to both sides of Eq. (390). Equation (391) is equal to zero, hence it will not alter the result, but will allow to simplify the RHS:

$$\begin{aligned}
\frac{\partial}{\partial a} \zeta^{CG} &= \pi(a)[(\hat{t}_{3a} + \bar{t}_{a3}) - (\hat{t}_{2a} + \bar{t}_{a2})] \\
&\quad + \pi(-a)[(\hat{t}_{1-a} + \bar{t}_{-a1}) - (\hat{t}_{2-a} + \bar{t}_{-a2})] \\
&\quad + \pi(a) \int_{-\infty}^{+\infty} dx p_2(x)(t_{xa} - \bar{t}_{x2}) \\
&\quad + \pi(-a) \int_{-\infty}^{+\infty} dx p_2(x)(t_{x-a} - \bar{t}_{x2})
\end{aligned} \tag{393}$$

Writing again  $\pi(a)p_2(x) = p_2(a)\pi(x)$  in the last two terms, we note that these vanish as  $\int_{-\infty}^{+\infty} dx \pi(x)t_{x-a}$  and  $\int_{-\infty}^{+\infty} dx \pi(x)t_{xa}$  are equal to the Kemeny constant of the original system,  $\zeta^{\text{orig}}$ , and  $\int_{-\infty}^{+\infty} dx \pi(x)\bar{t}_{x2} = \int dy p_2(y) \int_{-\infty}^{+\infty} dx \pi(x)t_{xy} = \zeta^{\text{orig}}$  as well. Hence, we are left with

$$\begin{aligned}
\frac{\partial}{\partial a} \zeta^{CG} &= \pi(-a) [(\hat{t}_{1-a} + \bar{t}_{-a1}) - (\hat{t}_{2-a} + \bar{t}_{-a2})] \\
&\quad + \pi(a) [(\hat{t}_{3a} + \bar{t}_{a3}) - (\hat{t}_{2a} + \bar{t}_{a2})]
\end{aligned} \tag{394}$$

Finally, using the definition of round-trip times given in Eq. (312) we find

$$\frac{\partial \zeta^{CG}}{\partial a} = \pi(a)[t_{3a}^{RT} - t_{2a}^{RT}] + \pi(-a)[t_{1-a}^{RT} - t_{2-a}^{RT}]$$

From symmetry of the potential, the above equation simplifies to

$$\frac{\partial \zeta^{CG}}{\partial a} = 2\pi(a)[t_{3a}^{RT} - t_{2a}^{RT}]. \tag{395}$$

## C.5 Diffusion on 1D potential: $m$ -state clustering

Considering a continuous one-dimensional space clustered into  $m$  clusters, each cluster  $J$  can be characterized by two boundaries,  $b_{J-1}$  and  $b_J$ , i.e. one has  $J = (b_{J-1}, b_J)$  for each state  $J = 1, \dots, m$ . The first and final (i.e.  $m$ th) clusters are bounded by the lower and upper boundaries  $b_0$  and  $b_m$  of the configuration space, which can take arbitrary values, including  $\pm\infty$ . Using the continuous formulation of Eq. (288)

$$\zeta^{CG} = \zeta^{orig} - \sum_{J=1}^m \frac{1}{\Pi_J} \int_J dy \int_J dx \pi(y) t_{yx} \pi(x) \quad (396)$$

and Eq. (310) we can write

$$\zeta^{CG} = \zeta^{orig} - \sum_{J=1}^m \int_{b_{J-1}}^{b_J} dy \pi(y) \bar{t}_{yJ} \quad (397)$$

The Kemeny derivative with respect to a boundary position  $b_L$  takes only contributions from  $J = L$  or  $J = L+1$ , giving

$$\begin{aligned} \frac{\partial \zeta^{CG}}{\partial b_L} = & -\frac{\partial}{\partial b_L} \left[ \int_{b_{L-1}}^{b_L} dy \pi(y) \bar{t}_{yL} \right. \\ & \left. + \int_{b_L}^{b_{L+1}} dy \pi(y) \bar{t}_{y(L+1)} \right] \end{aligned} \quad (398)$$

We apply the Leibniz differentiation rule to the first ( $\alpha$ ) and second ( $\beta$ ) integrals of Eq. (398) separately:

$$\begin{aligned} \frac{\partial}{\partial b_L} \alpha &= \pi(b_L) \bar{t}_{b_L L} + \int_{b_{L-1}}^{b_L} dy \pi(y) \frac{\partial}{\partial b_L} \bar{t}_{yL} \\ \frac{\partial}{\partial b_L} \beta &= -\pi(b_L) \bar{t}_{b_L (L+1)} + \int_{b_L}^{b_{L+1}} dy \pi(y) \frac{\partial}{\partial b_L} \bar{t}_{y(L+1)} \end{aligned} \quad (399)$$

Expanding  $\bar{t}_{yL}$  and using the Leibnitz rule again, we get

$$\begin{aligned} \frac{\partial}{\partial b_L} \bar{t}_{yL} &= \frac{\pi(b_L)}{\Pi_L} (t_{yb_L} - \bar{t}_{yL}) \\ \frac{\partial}{\partial b_L} \bar{t}_{y(L+1)} &= \frac{\pi(b_L)}{\Pi_{L+1}} (\bar{t}_{y(L+1)} - t_{yb_L}) \end{aligned} \quad (400)$$

Inserting Eq. (400) in Eq. (399) and rearranging we have

$$\begin{aligned} \frac{\partial \alpha}{\partial b_L} &= \pi(b_L) \left[ \bar{t}_{b_L L} + \int_{b_{L-1}}^{b_L} dy \frac{\pi(y)}{\Pi_L} (t_{yb_L} - \bar{t}_{yL}) \right] \\ \frac{\partial \beta}{\partial b_L} &= -\pi(b_L) \bar{t}_{b_L (L+1)} \\ &- \frac{\pi(b_L)}{\Pi_{L+1}} \int_{b_L}^{b_{L+1}} dy \pi(y) (t_{yb_L} - \bar{t}_{y(L+1)}) \end{aligned} \quad (401)$$

Using for the integrals the shorthand notation provided in Eq. (388) we can write

$$\begin{aligned} \frac{\partial}{\partial b_L} \alpha &= \pi(b_L) (\bar{t}_{b_L L} + \hat{t}_{Lb_L}) \\ \frac{\partial}{\partial b_L} \beta &= -\pi(b_L) (\bar{t}_{b_L (L+1)} + \hat{t}_{(L+1)b_L}) \end{aligned}$$

Inserting Eq. (402) in Eq. (398)

$$\begin{aligned} \frac{\partial \zeta^{CG}}{\partial b_L} = & \pi(b_L) [(\bar{t}_{b_L (L+1)} + \hat{t}_{(L+1)b_L}) \\ & - (\bar{t}_{b_L L} + \hat{t}_{Lb_L})] \end{aligned} \quad (402)$$

and using the definition of round-trip times given in Eq. (312) we obtain

$$\frac{\partial \zeta^{CG}}{\partial b_L} = \pi(b_L) \left[ t_{b_L(L+1)}^{RT} - t_{b_L L}^{RT} \right] \quad (403)$$

This shows that the derivative vanishes when the following equality is satisfied:

$$t_{b_L(L+1)}^{RT} = t_{b_L L}^{RT} \quad (404)$$

This provides an intuitive interpretation of our results: the optimal clusters  $L$  and  $L+1$  (obtained by maximizing the Kemeny constant) are those which have the same round-trip times to the separating barrier  $b_L$ .

## C.6 Discrete random walk on 1D potential: $m$ -state clustering

Here we consider the case of a 1D discrete system, i.e. a random walk on a linear chain. In discrete systems, we replace the partial derivative, of the coarse-grained Kemeny constant  $\zeta^{CG}$ , with respect to the barrier position  $b_L$ , with the finite difference between the assignments of a node  $b_L$  to clusters  $L$  and  $L+1$ , respectively. In particular if the clusters are ordered so that  $J < J+1 < J+2 < \dots$ , moving the barrier  $\alpha$  to "the right", corresponds to moving the node  $b_L$  to "the left", i.e. from cluster  $J+1$  to  $J$ . Hence, the equivalent to  $\partial \zeta^{CG} / \partial b_L$  in discrete space is:

$$\Delta \zeta^{CG}(b_L, L+1 \rightarrow L) = \zeta^{CG}(b_L \in L) - \zeta^{CG}(b_L \in L+1) \quad (405)$$

We can obtain a computationally efficient formula for the finite difference of the coarse-grained Kemeny constant. Following a similar derivation as in the previous section, expanding Eq. (402) and replacing the integrals with sums, we obtain

$$\begin{aligned} \frac{\Delta \zeta^{CG}}{\pi_{b_L}} = & \frac{1}{\Pi_L^2} \sum_{i \in L} \sum_{j \in L} \pi_j t_{ji} \pi_i - \frac{1}{\Pi_L} \sum_{j \in L} \pi_j t_{jb_L} \\ & - \frac{1}{\Pi_L} \sum_{i \in L} \pi_i t_{b_L i} - \frac{1}{\Pi_{L+1}^2} \sum_{i \in L+1} \sum_{j \in L+1} \pi_i \pi_j t_{ji} \\ & + \frac{1}{\Pi_{L+1}} \sum_{j \in L+1} \pi_j t_{jb_L} + \frac{1}{\Pi_{L+1}} \sum_{i \in L+1} \pi_i t_{b_L i}, \end{aligned} \quad (406)$$

We use the following property of the mean first passage times in one-dimensional space :

$$t_{ji} = \begin{cases} t_{jb_j} + t_{b_j j} & \text{if } j > b_j > i \\ t_{jb_j} - t_{ib_j} & \text{if } j < i < b_j \end{cases} \quad (407)$$

and expand the double sums in Eq. (406) as follows:

$$\begin{aligned}
\sum_{i \in L} \sum_{j \in L} \pi_j t_{ji} \pi_i &= \sum_{i \in L} \pi_i \left[ \sum_j \pi_j t_{ji} - \sum_{j \notin L} \pi_j t_{ji} \right] \\
&= \sum_{i \in L} \pi_i \left( \zeta^{orig} - \sum_{S=1}^{L-1} \sum_{j \in S} \pi_j t_{ji} - \sum_{S=L+1}^M \sum_{j \in S} \pi_j t_{ji} \right) \\
&= \sum_{i \in L} \pi_i \left[ \zeta^{orig} - \sum_{S=1}^{L-1} \sum_{j \in S} \pi_j (t_{jb_L} - t_{ib_L}) \right. \\
&\quad \left. - \sum_{S=L+1}^M \sum_{j \in S} \pi_j (t_{jb_L} + t_{b_L i}) \right] \\
&= \Pi_L \zeta^{orig} - \sum_{i \in L} \sum_{S=1}^{L-1} \sum_{j \in S} \pi_i \pi_j (t_{jb_L} - t_{ib_L}) \\
&\quad - \sum_{i \in L} \sum_{S=L+1}^M \sum_{j \in S} \pi_i \pi_j (t_{jb_L} + t_{b_L i}) \\
&= \Pi_L \left( \zeta^{orig} - \sum_{j \notin L} \pi_j t_{jb_L} \right) + \sum_{S=1}^{L-1} \Pi_S \sum_{i \in L} \pi_i t_{ib_L} \\
&\quad - \sum_{S=L+1}^M \Pi_S \sum_{i \in L} \pi_i t_{b_L i} \\
&= \sum_{S=1}^L \Pi_S \sum_{j \in L} \pi_j t_{jb_L} - \sum_{S=L+1}^M \Pi_S \sum_{i \in L} \pi_i t_{b_L i}
\end{aligned}$$

and, similarly:

$$\begin{aligned}
\sum_{i \in L+1} \sum_{j \in L+1} \pi_j t_{ji} \pi_i &= \\
&= \sum_{S=L+1}^M \Pi_S \sum_{j \in L+1} \pi_j t_{jb_L} - \sum_{S=1}^L \Pi_S \sum_{i \in L+1} \pi_i t_{b_L i}
\end{aligned}$$

Inserting these equations into Eq. (406) yields the finite difference of the Kemeny constant on a linear chain

$$\begin{aligned}
\frac{\Delta \zeta^{CG}}{\pi_{b_L}} &= \frac{\sum_{S < L} \Pi_S}{\Pi_L^2} \sum_{j \in L} \pi_j t_{jb_L} - \frac{\sum_{S \geq L} \Pi_S}{\Pi_L^2} \sum_{i \in L} \pi_i t_{b_L i} \\
&\quad + \frac{\sum_{S \leq L+1} \Pi_S}{\Pi_{L+1}^2} \sum_{i \in L+1} \pi_i t_{b_L i} - \frac{\sum_{S > L+1} \Pi_S}{\Pi_{L+1}^2} \sum_{j \in L+1} \pi_j t_{jb_L}.
\end{aligned} \tag{408}$$

## C.7 Random walks on complex networks: $m$ -state clustering

Below we will show that the result obtained in Appendix C.5 holds in complex networks, up to an approximation.

As a discrete equivalent of the Kemeny constant-derivative with respect to barrier position, we consider the finite difference in the coarse-grained Kemeny constant  $\zeta^{CG}$  between the case when node  $\alpha$  is assigned to cluster  $B$  or  $A$ .

$$\Delta \zeta^{CG}(\alpha, A \rightarrow B) = \zeta^{CG}(\alpha \in B) - \zeta^{CG}(\alpha \in A) \tag{409}$$

For convenience we define  $F_{JI}$  as follows:

$$F_{JI} = \sum_{i \in I} \sum_{j \in J} \pi_j t_{ji} \pi_i \tag{410}$$

This leads to a shorter formulation of Eq. (288) and Eq. (308) :

$$\zeta^{CG} = \zeta^{orig} - \sum_J \frac{F_{JJ}}{\Pi_J} \quad (411)$$

$$t_{JI}^{CG} = \frac{F_{JI}}{\Pi_J \Pi_I} - \frac{F_{JJ}}{\Pi_J^2} \quad (412)$$

If we consider splitting a cluster  $\Gamma$  into two clusters  $A$  and  $B$ , i.e.  $\Gamma = A \cup B$ , the following property holds:

$$F_{\Gamma\Gamma} = F_{AA} + F_{AB} + F_{BA} + F_{BB} \quad (413)$$

Additionally we define  $A^+ = A \cup \{\alpha\}$  and  $B^+ = B \cup \{\alpha\}$ , i.e. as the union of the node  $\alpha$  and the clusters  $A$  and  $B$  respectively, implying that  $A$  and  $B$  do not contain  $\alpha$ , hence

$$\Pi_{A^+} = \Pi_A + \pi_\alpha \quad (414)$$

Using Eq. (411) and dropping  $(\alpha, A \rightarrow B)$  for simplicity, we rewrite Eq. (409) as follows:

$$\begin{aligned} \Delta\zeta^{CG} &= \left[ \zeta^{orig} - \sum_{J \notin \{A, B\}} \frac{F_{JJ}}{\Pi_J} - \frac{F_{B^+B^+}}{\Pi_{B^+}} - \frac{F_{AA}}{\Pi_A} \right] \\ &\quad - \left[ \zeta^{orig} - \sum_{J \notin \{A, B\}} \frac{F_{JJ}}{\Pi_J} - \frac{F_{A^+A^+}}{\Pi_{A^+}} - \frac{F_{BB}}{\Pi_B} \right] \end{aligned} \quad (415)$$

Simplifying the first two terms in the brackets and rearranging, we have

$$\begin{aligned} \Delta\zeta^{CG} &= \left[ \frac{F_{BB}}{\Pi_B} - \frac{F_{B^+B^+}}{\Pi_{B^+}} \right] - \left[ \frac{F_{AA}}{\Pi_A} - \frac{F_{A^+A^+}}{\Pi_{A^+}} \right] \\ &= \frac{\Pi_{B^+}F_{BB} - \Pi_B F_{B^+B^+}}{\Pi_{B^+}\Pi_B} - \frac{\Pi_{A^+}F_{AA} - \Pi_A F_{A^+A^+}}{\Pi_{A^+}\Pi_A} \end{aligned} \quad (416)$$

Expanding  $F_{X^+X^+}$  and  $\Pi_{X^+}$  on the numerator, by using Eq. (413) and Eq. (414) respectively, and simplifying the cancelling terms:

$$\begin{aligned} \Delta\zeta^{CG} &= \left[ \frac{\Pi_\alpha F_{BB}}{\Pi_{B^+}\Pi_B} - \frac{F_{\alpha B} + F_{B\alpha} + F_{\alpha\alpha}}{\Pi_{B^+}} \right] \\ &\quad - \left[ \frac{\Pi_\alpha F_{AA}}{\Pi_{A^+}\Pi_A} - \frac{F_{A\alpha} + F_{\alpha A} + F_{\alpha\alpha}}{\Pi_{A^+}} \right] \end{aligned} \quad (417)$$

From  $t_{\alpha\alpha} = 0$  we have  $F_{\alpha\alpha} = 0$ , then dividing by  $\pi_\alpha$  and rearranging, we have:

$$\begin{aligned} \frac{\Delta\zeta^{CG}}{\pi_\alpha} &= \left[ \frac{F_{A\alpha}}{\pi_\alpha\Pi_{A^+}} - \frac{F_{AA}}{\Pi_{A^+}\Pi_A} + \frac{F_{\alpha A}}{\pi_\alpha\Pi_{A^+}} \right] \\ &\quad - \left[ \frac{F_{B\alpha}}{\pi_\alpha\Pi_{B^+}} - \frac{F_{BB}}{\Pi_{B^+}\Pi_B} + \frac{F_{\alpha B}}{\pi_\alpha\Pi_{B^+}} \right] \end{aligned} \quad (418)$$

Under the approximation  $\Pi_A^+ \simeq \Pi_A$  and  $\Pi_B^+ \simeq \Pi_B$ , which hold for  $\pi_\alpha \ll \Pi_A, \Pi_B$ :

$$\begin{aligned} \frac{\Delta\zeta^{CG}}{\pi_\alpha} &= \left[ \frac{F_{A\alpha}}{\pi_\alpha\Pi_A} - \frac{F_{AA}}{\Pi_A\Pi_A} + \frac{F_{\alpha A}}{\pi_\alpha\Pi_A} \right] \\ &\quad - \left[ \frac{F_{B\alpha}}{\pi_\alpha\Pi_B} - \frac{F_{BB}}{\Pi_B\Pi_B} + \frac{F_{\alpha B}}{\pi_\alpha\Pi_B} \right] \end{aligned} \quad (419)$$

We can use the formula for the coarse-grained mean first passage times in terms of  $F$ , Eq. (412), and simplify:

$$\frac{\Delta\zeta^{CG}}{\pi_\alpha} = [t_{A\alpha}^{CG} + t_{\alpha A}^{CG}] - [t_{B\alpha}^{CG} + t_{\alpha B}^{CG}] \quad (420)$$

As  $t_{A\alpha}^{CG}$  is the mean first passage time from a cluster that contains a single node  $\alpha$  to cluster  $A$ , this is the discrete-space analogue of the mean first passage time  $\hat{t}_{A\alpha}$  defined in Eq. (388) for diffusion in continuous space and, similarly,  $t_{\alpha A}^{CG}$  is analogous to  $\bar{t}_{\alpha A}$  defined in Eq. (310).

Upon defining, as before, the round-trip time  $t_{\alpha A}^{RT} = t_{A\alpha}^{CG} + t_{\alpha A}^{CG}$ , we can finally rewrite (420) in the intuitive way:

$$\Delta\zeta^{CG}(\alpha, A \rightarrow B) = \pi_\alpha [t_{\alpha A}^{RT} - t_{\alpha B}^{RT}] \quad (421)$$

This suggests that, given a network with  $M$  clusters, the cluster  $B$  to which a new node  $\alpha$  should be assigned, in order to maximize the Kemeny constant, is the one with minimum round trip time to  $\alpha$ , provided that  $\pi_\alpha \ll \Pi_L \forall L = 1, \dots, M$ .

## D SI - Clustering

The discretized  $N = 100$ -state irregular 1D 4-well potential ( $v$ ) was constructed using the following code:

```

 $v = [];$ 
for  $i = 1 : 4$ 
    if  $i == 2$ 
         $x = \text{linspace}(-4 * \pi, 4 * \pi, 2 * N/5);$ 
    else
         $x = \text{linspace}(-4 * \pi, 4 * \pi, N/5);$ 
    end
     $v_{seg} = -i * 3/4 * [\sin(x/4 - \pi * 3/2) + 1];$ 
     $v = [v \ v_{seg}];$ 
end
(422)

```

Table 7: Stochastic block model network reduced to three clusters.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	<b>63.931</b>	<b>40.774</b>	<b>0.40156</b>
Kemeny	<b>63.931</b>	<b>40.774</b>	<b>0.40156</b>
$\tau_2$	<b>63.931</b>	<b>40.774</b>	<b>0.40156</b>
Modularity	<b>63.931</b>	<b>40.774</b>	<b>0.40156</b>

Table 8: Santa Fe Collaboration network reduced to three clusters.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	<b>303.07</b>	<b>233.4</b>	<b>0.41022</b>
Kemeny	<b>303.07</b>	<b>233.4</b>	<b>0.41022</b>
$\tau_2$	<b>303.07</b>	<b>233.4</b>	<b>0.41022</b>
Modularity	<b>303.07</b>	<b>233.4</b>	<b>0.41022</b>

Table 9: Santa Fe Collaboration network reduced to four clusters.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	319.25	233.63	0.41209
Kemeny	<b>319.82</b>	<b>238.59</b>	0.41247
$\tau_2$	<b>319.82</b>	<b>238.59</b>	0.41247
Modularity	318.16	236.31	<b>0.41742</b>

Table 10: Political books network reduced to two clusters.

Clustering method	Kemeny	$\tau_2$	Modularity
PCCA+	<b>24.481</b>	<b>24.481</b>	0.35232
Kemeny	<b>24.481</b>	<b>24.481</b>	0.35232
$\tau_2$	<b>24.481</b>	<b>24.481</b>	0.35232
Modularity	24.372	24.372	<b>0.35345</b>

## E Acknowledgements

I thank M. Weber, and A. Martin, A. Szabo, A.Kells, F. Sicard, F.Faizi, and particularly A. Annibale and E. Rosta for numerous discussions and suggestions on the work. This work was supported by funding from **EPSRC** (EP/R013012/1) and **ERC** (project 757850 BioNet).

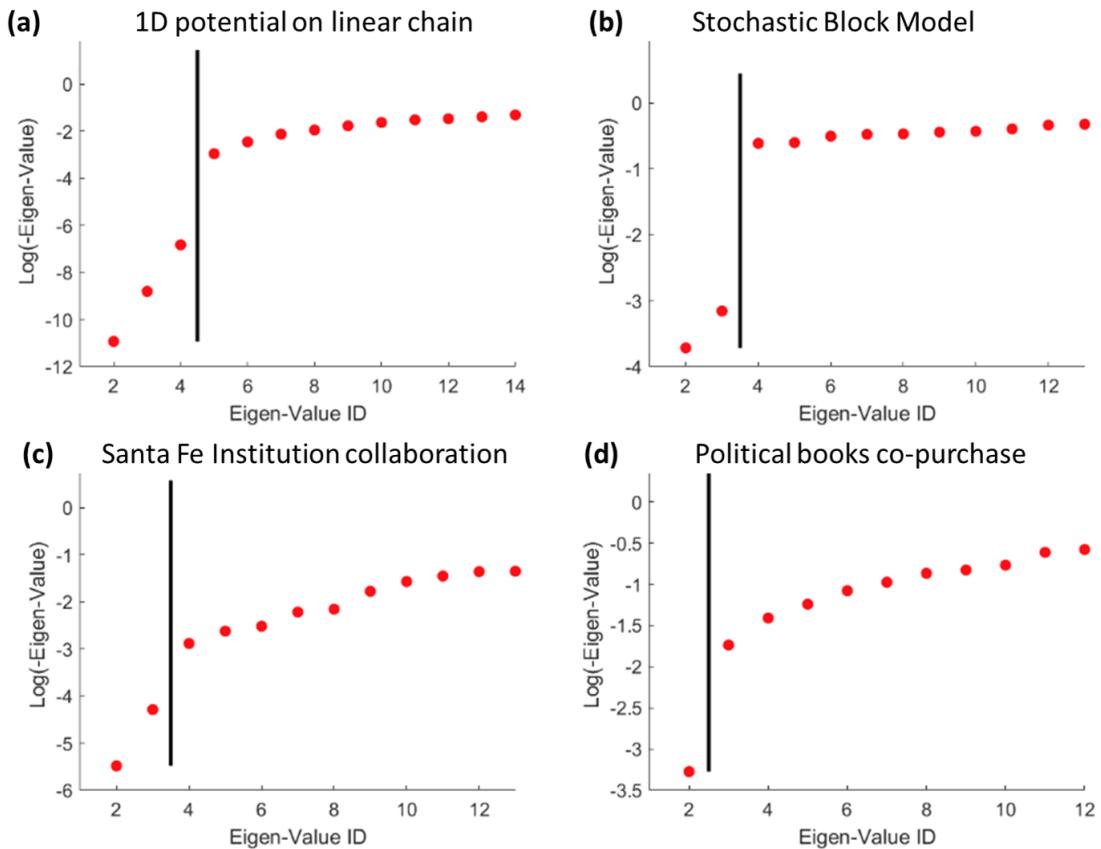


Figure 72: Spectral gap analysis of the networks considered. Negative logarithm of the dominant eigenvalues of the rate matrix in the networks: (a) 1D multi-well potential on a linear chain, (b) Stochastic Block Model, (c) Santa Fe Institute collaboration, and (d) Political Books co-purchase. Black vertical line indicates the largest gap between subsequent eigenvalues.

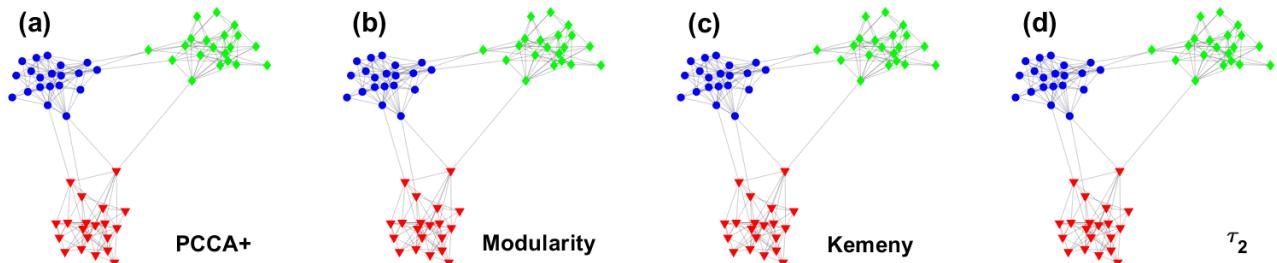


Figure 73: Clustering of the Stochastic Block Model network into three clusters.

## References

- [1] F. Noé and E. Rosta. “Special Topic: Markov Models of Molecular Kinetics”. In: *preprint: arXiv:1911.00774* (2019).
- [2] S. Doerr and G. De Fabritiis. “On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations”. In: *Journal of Chemical Theory and Computation* 10.5 (2014). PMID: 26580533, pp. 2064–2069. DOI: 10.1021/ct400919u. eprint: <https://doi.org/10.1021/ct400919u>. URL: <https://doi.org/10.1021/ct400919u>.
- [3] Callum J. Dickson et al. “Structure–Kinetic Relationships of Passive Membrane Permeation from Multi-scale Modeling”. In: *Journal of the American Chemical Society* 139.1 (2017). PMID: 27951634, pp. 442–

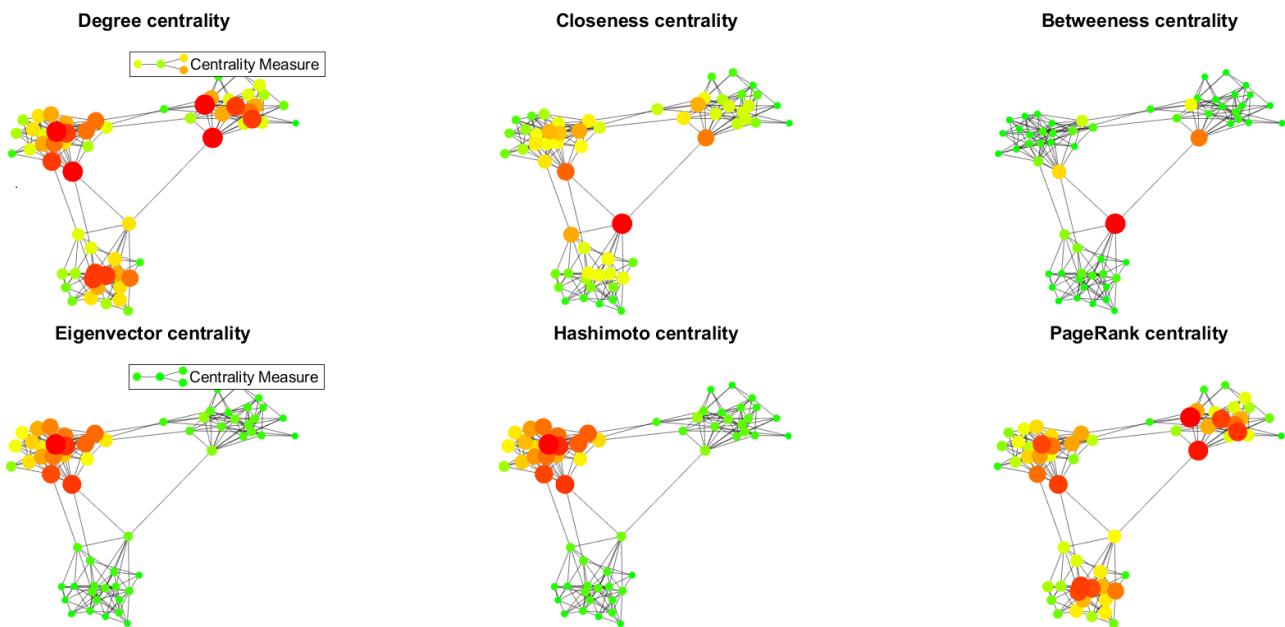


Figure 74: Centrality measures on the Stochastic Block Model network.

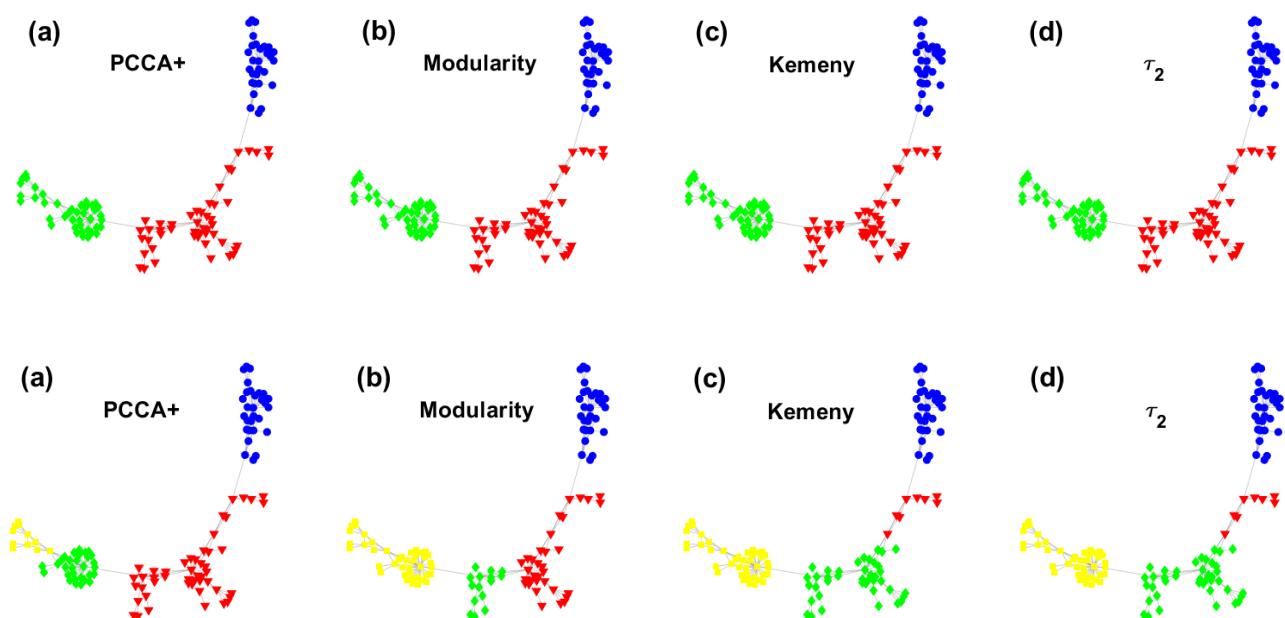


Figure 75: Clustering of the Santa Fe Institute collaboration network into three (top) and four (bottom) clusters.

452. DOI: 10.1021/jacs.6b11215. eprint: <https://doi.org/10.1021/jacs.6b11215>. URL: <https://doi.org/10.1021/jacs.6b11215>.

- [4] Magd Badaoui et al. “Calculating Kinetic Rates and Membrane Permeability from Biased Simulation”. In: *J. Phys. Chem. B* (2018).
- [5] Helmut Grubmüller and Paul Tavan. “Molecular dynamics of conformational substates for a simplified protein model”. In: *J. Chem. Phys.* 101.6 (1994), pp. 5047–5057. DOI: 10.1063/1.467427.
- [6] Susanna Kube and Marcus Weber. “A coarse graining method for the identification of transition rates between molecular conformations”. In: *J. Chem. Phys.* 126.2 (2007), p. 024103.
- [7] Frank Noé et al. “Hierarchical analysis of conformational dynamics in biomolecules: transition networks of metastable states”. In: *J. Chem. Phys.* 126.15 (2007), 04B617.
- [8] John D Chodera et al. “Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics”. In: *J. Chem. Phys.* 126.15 (2007), 04B616.

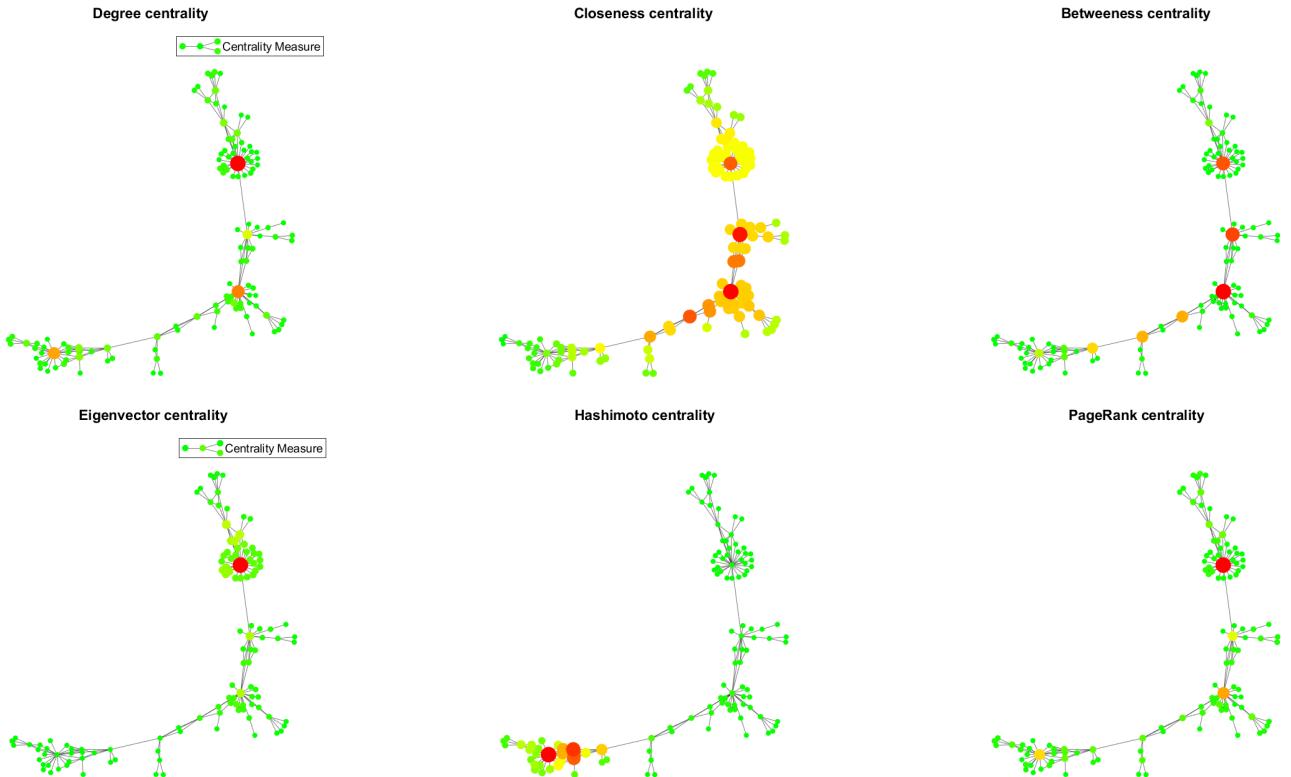


Figure 76: Centrality measures on the Santa Fe Institute collaboration network.

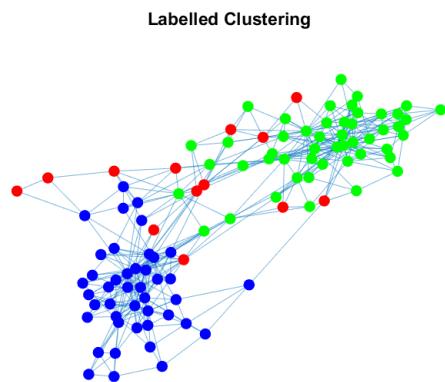


Figure 77: Labelling of the political books co-purchase network into three clusters: "conservative" (blue), "neutral" (red) and "liberal" (green) by Mark Newman [140].

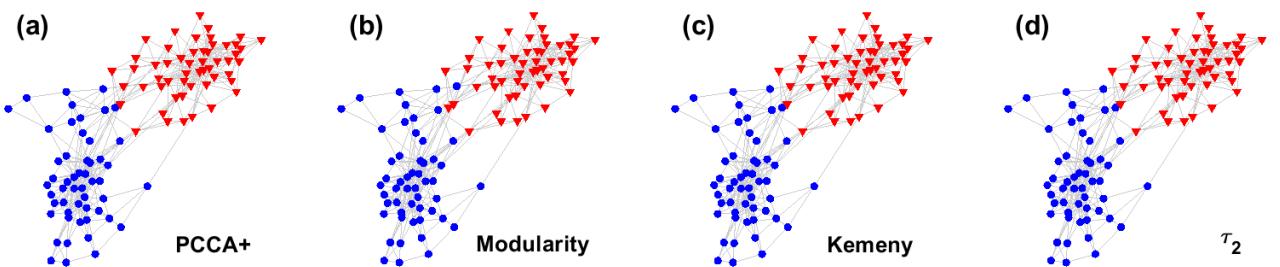


Figure 78: Clustering of the political books co-purchase network into two clusters.

[9] DJ Bicout and Attila Szabo. "Entropic barriers, transition states, funnels, and exponential protein folding kinetics: a simple model". In: *Protein Science* 9.3 (2000), pp. 452–465.

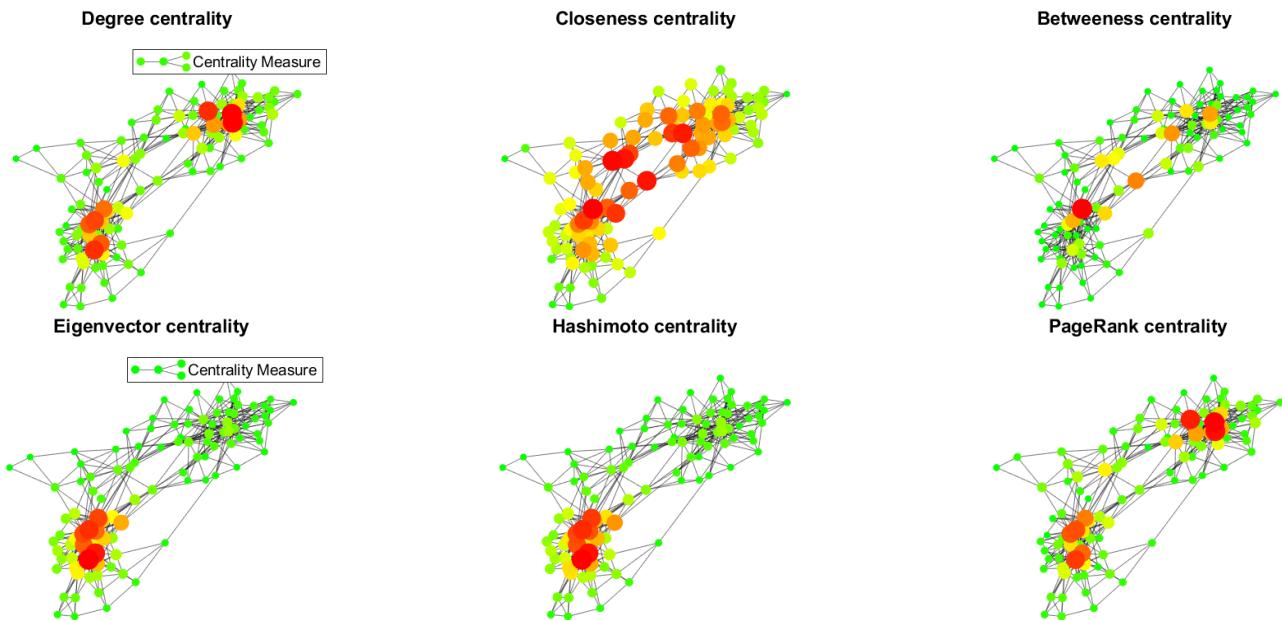


Figure 79: Centrality measures on the political books co-purchase network.

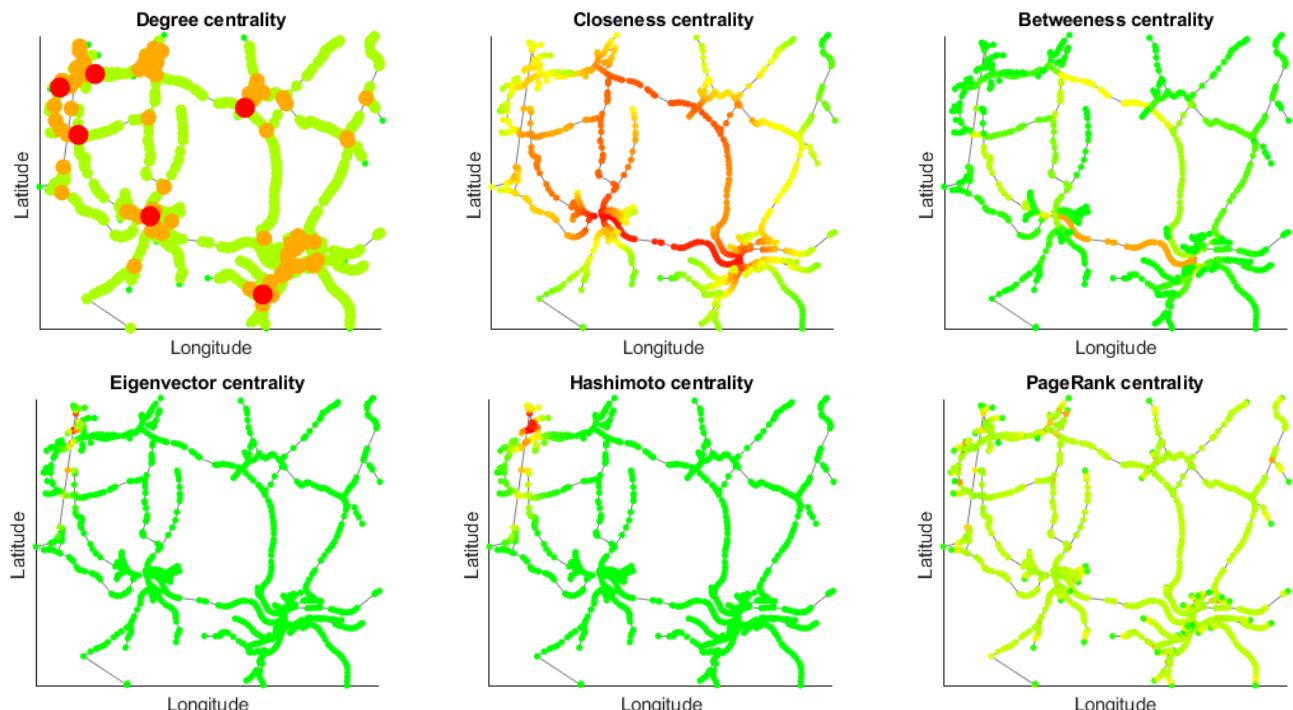


Figure 80: Centrality measures on the streets network.

- [10] Nicolae-Viorel Buchete and Gerhard Hummer. “Coarse master equations for peptide folding dynamics”. In: *J. Phys. Chem. B* 112.19 (2008), pp. 6057–6069.
- [11] 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”. In: *J. Phys. Chem. B* 121.22 (2017). PMID: 28489401, pp. 5509–5514. DOI: 10.1021/acs.jpcb.7b02083. eprint: <https://doi.org/10.1021/acs.jpcb.7b02083>. URL: <https://doi.org/10.1021/acs.jpcb.7b02083>.
- [12] A. Bujotzek and M. Weber. “EFFICIENT SIMULATION OF LIGAND–RECEPTOR BINDING PROCESSES USING THE CONFORMATION DYNAMICS APPROACH”. In: *Journal of Bioinformatics and Computational Biology* 07.05 (2009), pp. 811–831. DOI: 10.1142/S0219720009004369. eprint: <https://doi.org/10.1142/S0219720009004369>. URL: <https://doi.org/10.1142/S0219720009004369>.

- [13] Vijay S Pande, Kyle Beauchamp, and Gregory R Bowman. “Everything you wanted to know about Markov State Models but were afraid to ask”. In: *Methods* 52.1 (2010), pp. 99–105.
- [14] G.R. Bowman, V. S. Pande, and F. Noe, eds. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Netherlands: Springer, 2014.
- [15] Brooke E Husic and Vijay S Pande. “Markov state models: From an art to a science”. In: *Journal of the American Chemical Society* 140.7 (2018), pp. 2386–2396.
- [16] C. Wehmeyer et al. “Introduction to Markov state modeling with the PyEMMA software—v1. 0”. In: *Living J. Comp. Mol. Sci.* 1 (1 2019).
- [17] Linda Martini et al. “Variational Identification of Markovian Transition States”. In: *Phys. Rev. X* 7 (3 Sept. 2017), p. 031060. DOI: 10.1103/PhysRevX.7.031060. URL: <https://link.aps.org/doi/10.1103/PhysRevX.7.031060>.
- [18] A.M. Berezhkovskii and A. Szabo. “Committors, first-passage times, fluxes, Markov states, milestones, and all that”. In: *J. Chem. Phys.* 150 (2019), p. 054106.
- [19] Adam Kells et al. “Mean first passage times in variational coarse graining using Markov state models”. In: *J. Chem. Phys.* 150.13 (2019), p. 134107.
- [20] Adam Kells, Alessia Annibale, and Edina Rosta. “Limiting relaxation times from Markov state models”. In: *J. Chem. Phys.* 149.7 (2018), p. 072324. DOI: 10.1063/1.5027203. eprint: <https://doi.org/10.1063/1.5027203>. URL: <https://doi.org/10.1063/1.5027203>.
- [21] Attila Szabo, Klaus Schulten, and Zan Schulten. “First passage time approach to diffusion controlled reactions”. In: *J. Chem. Phys.* 72.8 (1980), pp. 4350–4357.
- [22] Angelo Perico et al. “Positional time correlation function for one-dimensional systems with barrier crossing: Memory function corrections to the optimized Rouse–Zimm approximation”. In: *J. Chem. Phys.* 98.1 (1993), pp. 564–573.
- [23] DJ Bicout and Attila Szabo. “First passage times, correlation functions, and reaction rates”. In: *J. Chem. Phys.* 106.24 (1997), pp. 10292–10298.
- [24] M. Kac. “On the notion of recurrence in discrete stochastic processes”. In: *Bull. Amer. Math. Soc.* 53 (1947), pp. 1002–1010.
- [25] L. Lovász. “Random Walks on Graphs: A Survey”. In: *Combinatorics, Paul Erdős is Eighty*. Ed. by D. Miklós, V. T. Sós, and T. Szőnyi. Vol. 2. Budapest: János Bolyai Mathematical Society, 1996, pp. 353–398.
- [26] Z. Zhang, T. Shan, and G. Chen. “Random walks on weighted networks”. In: *Phys. Rev. E* 87 (2013), p. 012112.
- [27] J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Princeton, NJ: Van Nostrand, 1960.
- [28] Peter G Doyle. “The Kemeny constant of a Markov chain”. In: *arXiv preprint arXiv:0909.2636* (2009).
- [29] J. J. Hunter. “The role of Kemeny’s constant in properties of Markov chains”. In: *Commun. Statist. Theory Meth.* 43 (2014), pp. 1309–1321.
- [30] A. K. Faradjian and R. Elber. “Computing time scales from reaction coordinates by milestoning”. In: *J. Chem. Phys.* 120 (2004), p. 10880.
- [31] E. Vanden-Eijnden et al. “On the assumptions underlying milestoning”. In: *J. Chem. Phys.* 129 (2008), p. 174102.
- [32] R. Elber. “A new paradigm for atomically detailed simulations of kinetics in biophysical systems.” In: *Q. Rev. Biophys.* 50 (2017), e8.
- [33] L. Lin, J. Lu, and E. Vanden-Eijnden. “A Mathematical Theory of Optimal Milestoning (with a Detour via Exact Milestoning)”. In: *Commun. Pure Appl. Math.* 71 (6 2018), pp. 1149–77.
- [34] C. H. Bennett. “Molecular Dynamics and Transition State Theory: The Simulation of Infrequent Events”. In: *Algorithms for Chemical Computations, ACS Symposium Series No. 46*. Washington, D.C.: American Chemical Society, 1977. Chap. 4, pp. 63–97.
- [35] C. Dellago et al. “Transition path sampling and the calculation of rate constants”. In: *J. Chem. Phys.* 108.5 (1998), pp. 1964–1977. DOI: 10.1063/1.475562.
- [36] D. Chandler. “Statistical mechanics of isomerization dynamics in liquids and the transition state approximation”. In: *The Journal of Chemical Physics* 68 (6 1978).
- [37] C. Dellago, P. G. Bolhuis, and D. Chandler. “Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements”. In: *The Journal of Chemical Physics* 108 (22 1998).

- [38] T.S. Van Erp, D. Moroni, and P.G. Bolhuis. “A novel path sampling method for the calculation of rate constants”. In: *The Journal of Chemical Physics* 118 (17 2003).
- [39] C. M. Grinstead and J. L. Snell. *Introduction to probability*. Providence, RI: American Mathematical Society, 1997.
- [40] D. Bini et al. “Why is Kemeny’s constant a constant?” In: *J. Appl. Prob.* 55 (4 2018), pp. 1025–1036.
- [41] P. Coolen-Schrijner and E. A. van Doorn. “The deviation matrix of a continuous-time Markov chain.” In: *Prob. Eng. Informat. Sci.* 16 (2002), pp. 351–366.
- [42] E. Weinan and E. Vanden-Eijnden. “Transition-Path Theory and Path-Finding Algorithms for the Study of Rare Events”. In: *Annu. Rev. Phys. Chem.* 61 (1 2010), pp. 391–420.
- [43] E. Vanden-Eijnden and M. Venturoli. In: *J. Chem. Phys.* 131 (2009), p. 044120.
- [44] Miao Lin Zhuo Qi Lee Wen-Jing Hsu. “Estimating Mean First Passage Time of BiasedWalks with Short Relaxation Time on Complex Networks”. In: *PLoS ONE* 9 (4 2014), e93348.
- [45] I.P. Androulakis, E. Yang, and R.R. Almon. “Analysis of Time-Series Gene Expression Data: Methods, Challenges, and Opportunities”. In: *Annu Rev Biomed Eng.* 9 (2007), pp. 205–228.
- [46] Johannes Kästner. “Umbrella sampling”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.6 (2011), pp. 932–942.
- [47] Alessandro Barducci, Massimiliano Bonomi, and Michele Parrinello. “Metadynamics”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.5 (2011), pp. 826–843.
- [48] Rafael C Bernardi, Marcelo CR Melo, and Klaus Schulten. “Enhanced sampling techniques in molecular dynamics simulations of biological systems”. In: *Biochimica et Biophysica Acta (BBA)-General Subjects* 1850.5 (2015), pp. 872–877.
- [49] C. Geyer. “Monte Carlo Maximum Likelihood for Depend Data”. In: *23rd Symposium on the Interface between Computing Science and Statistics*. Fairfax: Interface Foundation North America, 1991, p. 156.
- [50] K. Hukushima and K. Nemoto. “Exchange Monte Carlo Method and Application to Spin Glass Simulations”. In: *J. Phys. Soc. Jpn.* 65 (1996), p. 1604.
- [51] Enzo Marinari and Giorgio Parisi. “Simulated tempering: a new Monte Carlo scheme”. In: *Europhysics Letters* 19.6 (1992), p. 451.
- [52] Shankar Kumar et al. “The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method”. In: *Journal of computational chemistry* 13.8 (1992), pp. 1011–1021.
- [53] Edina Rosta and Gerhard Hummer. “Free energies from dynamic weighted histogram analysis using unbiased Markov state model”. In: *Journal of chemical theory and computation* 11.1 (2015), pp. 276–285.
- [54] Lukas S Stelzl and Gerhard Hummer. “Kinetics from replica exchange molecular dynamics simulations”. In: *Journal of chemical theory and computation* 13.8 (2017), pp. 3927–3935.
- [55] Richard Bellman. “A Markovian decision process”. In: *Journal of mathematics and mechanics* (1957), pp. 679–684.
- [56] Ronald A Howard. “Dynamic programming and markov processes.” In: (1960).
- [57] Andrew Wrobel. “On Markovian decision models with a finite skeleton”. In: *Zeitschrift für Operations Research* 28.1 (1984), pp. 17–27.
- [58] Marija Vucelja. “Lifting—a nonreversible Markov chain Monte Carlo algorithm”. In: *American Journal of Physics* 84.12 (2016), pp. 958–968.
- [59] Fahim Faizi, George Deligiannidis, and Edina Rosta. “Efficient Irreversible Monte Carlo Samplers”. In: *Journal of chemical theory and computation* 16.4 (2020), pp. 2124–2138.
- [60] Marcus Kaiser, Robert L Jack, and Johannes Zimmer. “Acceleration of convergence to equilibrium in Markov chains by breaking detailed balance”. In: *Journal of statistical physics* 168.2 (2017), pp. 259–287.
- [61] Robert Zwanzig. “From classical dynamics to continuous time random walks”. In: *Journal of Statistical Physics* 30.2 (1983), pp. 255–262.
- [62] Gerhard Hummer and Attila Szabo. “Optimal dimensionality reduction of multistate kinetic and Markov-state models”. In: *J. Phys. Chem. B* 119.29 (2014), pp. 9029–9037.
- [63] Gregory R Bowman, Xuhui Huang, and Vijay S Pande. “Using generalized ensemble simulations and Markov state models to identify conformational states”. In: *Methods* 49.2 (2009), pp. 197–201.
- [64] Amit Saxena et al. “A review of clustering techniques and developments”. In: *Neurocomputing* 267 (2017), pp. 664–681.

- [65] David Gfeller and Paolo De Los Rios. “Spectral coarse graining of complex networks”. In: *Physical review letters* 99.3 (2007), p. 038701.
- [66] Craig Gotsman. “On graph partitioning, spectral analysis, and digital mesh processing”. In: *2003 Shape Modeling International*. IEEE. 2003, pp. 165–171.
- [67] Muhammad Aqib Javed et al. “Community detection in networks: A multidisciplinary review”. In: *Journal of Network and Computer Applications* 108 (2018), pp. 87–111.
- [68] Hélio Almeida et al. “Is there a best quality metric for graph clusters?” In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2011, pp. 44–59.
- [69] Stephen G Kobourov, Sergey Pupyrev, and Paolo Simonetto. “Visualizing Graphs as Maps with Contiguous Regions.” In: *EuroVis (Short Papers)*. 2014.
- [70] David Chandler. “Statistical mechanics of isomerization dynamics in liquids and the transition state approximation”. In: *J. Chem. Phys.* 68.6 (1978), pp. 2959–2970.
- [71] Daniel Spielman. “Spectral graph theory”. In: *Combinatorial scientific computing*. 18. Citeseer, 2012.
- [72] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. “Finding community structure in very large networks”. In: *Physical review E* 70.6 (2004), p. 066111.
- [73] Mark EJ Newman. “Modularity and community structure in networks”. In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.
- [74] Vincent D Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.
- [75] Thomas Aynaud et al. “Multilevel local optimization of modularity”. In: *Graph Partitioning*. John Wiley and Sons, 2013, pp. 315–345.
- [76] Santo Fortunato and Marc Barthelemy. “Resolution limit in community detection”. In: *Proceedings of the national academy of sciences* 104.1 (2007), pp. 36–41.
- [77] Scott Emmons et al. “Analysis of network clustering algorithms and cluster quality metrics at scale”. In: *PloS one* 11.7 (2016), e0159161.
- [78] J-C Delvenne, Sophia N Yaliraki, and Mauricio Barahona. “Stability of graph communities across time scales”. In: *Proceedings of the national academy of sciences* 107.29 (2010), pp. 12755–12760.
- [79] Peter Deuflhard et al. “Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains”. In: *Linear Algebra and its Applications* 315.1-3 (2000), pp. 39–59.
- [80] Peter Deuflhard and Marcus Weber. “Robust Perron cluster analysis in conformation dynamics”. In: *Linear algebra and its applications* 398 (2005), pp. 161–184.
- [81] Marcus Weber. “Meshless Methods in Confirmation Dynamics”. In: (2006).
- [82] Susanna Kube and Marcus Weber. “A coarse graining method for the identification of transition rates between molecular conformations.” In: *J. Chem. Phys.* 126 2 (2007), p. 024103.
- [83] Susanna Röblitz and Marcus Weber. “Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification”. In: *Advances in Data Analysis and Classification* 7.2 (2013), pp. 147–179.
- [84] Shraddha K Popat and M Emmanuel. “Review and comparative study of clustering techniques”. In: *International journal of computer science and information technologies* 5.1 (2014), pp. 805–812.
- [85] Adam Kells et al. “Mean first passage times in variational coarse graining using Markov state models”. In: *The Journal of chemical physics* 150.13 (2019), p. 134107.
- [86] Adam Kells et al. “Correlation functions, mean first passage times, and the Kemeny constant”. In: *The Journal of Chemical Physics* 152.10 (2020), p. 104108.
- [87] Gerhard Hummer and Attila Szabo. “Optimal dimensionality reduction of multistate kinetic and Markov-state models”. In: *The Journal of Physical Chemistry B* 119.29 (2015), pp. 9029–9037.
- [88] Linda Martini et al. “Variational identification of Markovian transition states”. In: *Physical Review X* 7.3 (2017), p. 031060.
- [89] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. “Laplacian dynamics and multiscale modular structure in networks”. In: *arXiv preprint arXiv:0812.1770* (2008).
- [90] Marcus Weber and Konstantin Fackeldey. “G-PCCA: Spectral clustering for non-reversible Markov chains”. In: (2015).
- [91] Natasa Djurdjevac Conrad, Marcus Weber, and Christof Schütte. “Finding dominant structures of non-reversible Markov processes”. In: *Multiscale Modeling & Simulation* 14.4 (2016), pp. 1319–1340.

- [92] John G Kemeny and J Laurie Snell. "Finite Markov Chains. D Van Nostad Co". In: *Inc., Princeton, NJ* (1960).
- [93] Pauline Coolen-Schrijner and Erik A Van Doorn. "The deviation matrix of a continuous-time Markov chain". In: *Probability in the Engineering and informational Sciences* 16.3 (2002), p. 351.
- [94] J.L. Palacios. In: *Int. J. Quantum Chem.* 81 (2001).
- [95] J. Berkhout and B. F. Heidergott. "Analysis of Markov influence graphs". In: *Operations Research* 67 (3 2019), pp. 892–904.
- [96] Z. Zhang et al. "Optimal and suboptimal networks for efficient navigation measured by mean-first passage time of random walks". In: *Chaos* 22 (2012), p. 043129.
- [97] Linton C Freeman. "A set of measures of centrality based on betweenness". In: *Sociometry* (1977), pp. 35–41.
- [98] Alex Bavelas. "Communication patterns in task-oriented groups". In: *The Journal of the Acoustical Society of America* 22.6 (1950), pp. 725–730.
- [99] László Lovász. "Random walks on graphs". In: *Combinatorics, Paul Erdős is eighty* 2.1-46 (1993), p. 4.
- [100] Sam Northshield and José Luis Palacios. "On the commute time of random walks on graphs". In: *Brazilian Journal of Probability and Statistics* (1995), pp. 169–175.
- [101] Alexander M Berezhkovskii and Attila Szabo. "Committors, first-passage times, fluxes, Markov states, milestones, and all that". In: *The Journal of chemical physics* 150.5 (2019), p. 054106.
- [102] Huaijun Qiu and Edwin R Hancock. "Clustering and embedding using commute times". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.11 (2007), pp. 1873–1890.
- [103] Luh Yen et al. "Graph nodes clustering based on the commute-time kernel". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2007, pp. 1037–1045.
- [104] Régis Behmo, Nikos Paragios, and Véronique Prinet. "Graph commute times for image representation". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.
- [105] Gregory R. Bowman. "Improved coarse-graining of Markov state models via explicit consideration of statistical uncertainty". In: *J. Chem. Phys.* 137 (2012).
- [106] Hendrik Jung, Kei-ichi Okazaki, and Gerhard Hummer. "Transition path sampling of rare events by shooting from the top". In: *The Journal of chemical physics* 147.15 (2017), p. 152716.
- [107] M. A. Novotny. "Monte Carlo Algorithms with Absorbing Markov Chains: Fast Local Algorithms for Slow Dynamics". In: *Phys. Rev. Lett.* 74 (1 Jan. 1995), pp. 1–5. DOI: 10.1103/PhysRevLett.74.1. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.74.1>.
- [108] Michail Stamatakis and Dionisios G. Vlachos. "Equivalence of on-lattice stochastic chemical kinetics with the well-mixed chemical master equation in the limit of fast diffusion". In: *Computers & Chemical Engineering* 35.12 (2011), pp. 2602–2610. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2011.05.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0098135411001797>.
- [109] Daniel J. Sharpe and David J. Wales. "Nearly reducible finite Markov chains: Theory and algorithms". In: *The Journal of Chemical Physics* 155.14 (2021), p. 140901. DOI: 10.1063/5.0060978. eprint: <https://doi.org/10.1063/5.0060978>. URL: <https://doi.org/10.1063/5.0060978>.
- [110] Renaud Lambiotte, Jean-Charles Delvenne, and Mauricio Barahona. "Random walks, Markov processes and the multiscale modular organization of complex networks". In: *IEEE Transactions on Network Science and Engineering* 1.2 (2014), pp. 76–90.
- [111] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174.
- [112] M. R. Garey and D. S. Johnson. *Computers and intractability*. San Francisco: W.H. Freeman, 1979.
- [113] Andrea Kölzsich et al. "A periodic Markov model to formalize animal migration on a network". In: *Royal Society Open Science* 5.6 (2018), p. 180438. DOI: 10.1098/rsos.180438.
- [114] Aditya Gupta and Bhuwan Dhingra. "Stock market prediction using hidden markov models". In: *2012 Students Conference on Engineering and Systems*. IEEE. 2012, pp. 1–4.
- [115] Sergey Brin and Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". In: *COMPUTER NETWORKS AND ISDN SYSTEMS*. 1998, pp. 107–117.
- [116] John G Kemeny, J Laurie Snell, and Anthony W Knapp. "Properties of Markov Chains". In: *Denumerable Markov Chains*. Springer, 1976, pp. 79–105.
- [117] John G Kemeny, J Laurie Snell, and Anthony W Knapp. *Denumerable Markov chains: with a chapter of Markov random fields by David Griffeth*. Vol. 40. Springer Science & Business Media, 2012.

- [118] Marcus Weber and Konstantin Fackeldey. *G-PCCA: Spectral Clustering for Non-reversible Markov Chains*. Tech. rep. 15-35. Takustr. 7, 14195 Berlin: ZIB, 2015.
- [119] Ch Schütte et al. “A direct approach to conformational dynamics based on hybrid Monte Carlo”. In: *Journal of Computational Physics* 151.1 (1999), pp. 146–168.
- [120] Marcus Weber. *Clustering by using a simplex structure*. Tech. rep. 04-03. Takustr. 7, 14195 Berlin: ZIB, 2003.
- [121] Marcus Weber. *Improved Perron Cluster Analysis*. Tech. rep. 03-04. Takustr. 7, 14195 Berlin: ZIB, 2003.
- [122] Marcus Weber, Wasinee Rungsarityotin, and Alexander Schliep. *Perron Cluster Analysis and Its Connection to Graph Partitioning for Noisy Data*. Tech. rep. 04-39. Takustr. 7, 14195 Berlin: ZIB, 2004.
- [123] Susanna Kube and Marcus Weber. *Identification of Metastabilities in Monomolecular Conformation Kinetics*. Tech. rep. 06-01. Takustr. 7, 14195 Berlin: ZIB, 2005.
- [124] Susanna Kube and Marcus Weber. *Conformation Kinetics as a Reduced Model for Transition Pathways*. Tech. rep. 05-43. Takustr. 7, 14195 Berlin: ZIB, 2005.
- [125] Alexander K. Hartmann and Dieter W. Heermann. “Description of noble gas diffusion in a polymer matrix by a hopping model”. In: *J. Chem. Phys.* 108.22 (1998), pp. 9550–9557. DOI: 10.1063/1.476403.
- [126] Alexander K. Hartmann and H. Rieger. *Optimization Algorithms in Physics*. Weinheim: Wiley-VCH, 2001.
- [127] Alexander K. Hartmann and H. Rieger, eds. *New Optimization Algorithms in Physics*. Weinheim: Wiley-VCH, 2004.
- [128] Pasquale De Meo et al. “Generalized louvain method for community detection in large networks”. In: *2011 11th international conference on intelligent systems design and applications*. IEEE. 2011, pp. 88–93.
- [129] Marcelo Blatt, Shai Wiseman, and Eytan Domany. “Clustering data through an analogy to the Potts model”. In: *Advances in neural information processing systems* (1996), pp. 416–422.
- [130] Jörg Reichardt and Stefan Bornholdt. “Detecting fuzzy community structures in complex networks with a Potts model”. In: *Physical review letters* 93.21 (2004), p. 218701.
- [131] J. J. Moreno, H. G. Katzgraber, and Alexander K. Hartmann. “Finding low-temperature states with parallel tempering, simulated annealing and simple Monte Carlo”. In: *Int. J. Mod. Phys. C* 14.3 (2003), pp. 285–302.
- [132] David J Earl and Michael W Deem. “Parallel tempering: Theory, applications, and new perspectives”. In: *Physical Chemistry Chemical Physics* 7.23 (2005), pp. 3910–3916.
- [133] Jon Machta. “Strengths and weaknesses of parallel tempering”. In: *Physical Review E* 80.5 (2009), p. 056706.
- [134] Alexander K. Hartmann. *Big Practical Guide to Computer Simulations*. Singapore: World Scientific, 2015.
- [135] Robert H Swendsen and Jian-Sheng Wang. “Replica Monte Carlo simulation of spin-glasses”. In: *Physical review letters* 57.21 (1986), p. 2607.
- [136] Francisco Barahona. “On the computational complexity of Ising spin glass models”. In: *Journal of Physics A: Mathematical and General* 15.10 (1982), p. 3241.
- [137] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. “Stochastic blockmodels: First steps”. In: *Social Networks* 5.2 (1983), pp. 109–137. ISSN: 0378-8733. DOI: [https://doi.org/10.1016/0378-8733\(83\)90021-7](https://doi.org/10.1016/0378-8733(83)90021-7). URL: <https://www.sciencedirect.com/science/article/pii/0378873383900217>.
- [138] M. Girvan and M. E. J. Newman. “Community structure in social and biological networks”. In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.122653799>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.122653799>.
- [139] V. Krebs. *Unpublished*. <https://www.cc.gatech.edu/dimacs10/archive/clustering.shtml>. <http://www.orgnet.com/>. 2004.
- [140] Mark Newman. *Network data, Books about US politics*. <http://www-personal.umich.edu/~mejn/netdata/>. 2013.
- [141] David A Bader et al. *10th DIMACS Implementation Challenge-Graph Partitioning and Graph Clustering*. 2011.
- [142] David A Bader et al. *Graph partitioning and graph clustering*. Vol. 588. American Mathematical Society Providence, RI, 2013.