THE UNIVERSITY OF CHICAGO

TOWARDS THE ACCURATE ESTIMATION OF RARE EVENTS IN MOLECULAR DYNAMICS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMISTRY

BY
ERIK HENNING THIEDE

CHICAGO, ILLINOIS
AUGUST 2019

Dedicated to Ralf and Barbara Thiede.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to my advisors Aaron Dinner and Jonathan Weare. Throughout my doctoral studies, they have given me invaluable support and advice. I could not have asked for a better set of PhD advisors.

I also would like to thank my current and past colleagues, both in the Dinner and Weare groups and at the University of Chicago as a whole. My time with you has been stimulating and enjoyable. In particular, I would like to acknowledge my co-authors. Prof. Brian van Koten was my co-author on Chapter 2. As ever, I admire the rigor and clarity with which he dedicates himself to his work. Chapter 3 was written under the joint mentorship of Prof. Dimitrios Giannakis. Dimitrios is not only a brilliant scholar but a complete mensch. My co-author in Chapter 4 was Rob Webber. Rob combines an excellent mind with an infectious optimism. Working with him is a delight.

I would further like to thank my committee members, Prof. Benoit Roux and Prof. Mazziotti. They have helped guide me through critical points of my doctoral path, and I am honored that they could be part of my thesis committee.

Finally, I am deeply indebted to my partner, Serafina Ye Jin Ha Kim. Her contribution to this work cannot be overstated.

This work was supported by National Institutes of Health (NIH) Grant Number 5 R01 GM109455-02 and by the Molecular Software Sciences Institute (MolSSI) Software Fellows program. Computing resources were provided by the University of Chicago Research Computing Center (RCC). The Fip35 WW domain data was provided by D.E. Shaw Research.

ABSTRACT

Molecular dynamics simulations can give atomistic insight into chemical systems and processes. However, key molecular motions often depend on statistically rare events. Quantitatively describing these motions requires the use of *enhanced-sampling schemes* that increase the probability of seeing rare events in simulations. In this thesis, we use mathematical approaches to analyze enhanced-sampling algorithms and introduce new ones.

We first turn our attention to umbrella sampling, one of the most widely used enhanced sampling algorithms. In umbrella sampling, forces that bias the system towards the rare events are applied. By combining data from multiple, differently biased simulations, averages against the probability distribution of the unbiased ensemble can be recovered. We analyze this scheme, formally justifying why umbrella sampling works and demonstrating how the scheme scales with certain design choices. We also introduce a new algorithm for recombining the data from separate simulations. This formulation allows for the first rigorous analysis of the error in umbrella sampling.

While umbrella sampling can exponentially accelerate convergence of statistical estimates, the biasing procedure irrevocably alters the system's dynamics. Consequently, umbrella sampling calculations can only produce averages against the system's equilibrium distribution. This prevents direct estimation of dynamical statistics, such as chemical rates or committor probabilities. To address this issue, we introduce a new scheme that connects dynamical statistics to operator-theoretic descriptions of the system's dynamics. Using the operator-theoretic description enables us to avoid the challenging task of directly sampling the ensemble of reactive pathways. Moreover, the scheme does not require knowledge of a good low-dimensional description of the system, and does not require tight control over the system's dynamics. We also show that dynamical estimates from Markov state models (MSMs) correspond to a specific realization of our scheme.

Finally, we turn our attention to the problem of dimensionality reduction. One common approach is to consider the spectral properties of the operators discussed in the previous

section. A common scheme for estimating dynamical quantities is the variational approach to conformational dynamics (VAC). Just as our work generalizes the estimation of rates and committors using MSMs, VAC generalizes the calculation of the eigenvalues and eigenvectors of the MSM transition matrix. We analyze VAC schemes and study how the choice of basis set, the amount of sampling, and the choice of lag time in the scheme affects the approximation of the system's slow modes. Our analysis shows that the output of VAC can be strongly dependent on the lag time and leads to new heuristics for choosing this parameter.

A unifying theme in this work is the importance of sampling error. Not only does it motivate the development of new enhanced sampling schemes, error analyses give us insight into existing numerical schemes. In umbrella sampling, error analysis informs simulation design choices and suggests how computational resources may be better allocated. In VAC, it guides parameter choice and helps inform what can be reasonably expected from these schemes. Our work shows how error propagates through numerical schemes can lead to improved algorithms for performing molecular dynamics.

CHAPTER 1

INTRODUCTION

Trajectory averages in molecular dynamics simulations allow us to predict quantities central to chemical theory, such as free energies and chemical rates. However, the associated processes often depend on rare events. If one were to attempt to estimate these quantities by directly integrating the system's equations of motion forward, the amount of computational resources required could become prohibitively expensive. Analyzing these systems requires the use of enhanced sampling schemes, which attempt to increase the probability of seeing rare events in simulations.

While enhanced sampling schemes increase the prevalence of rare events in the dataset, these schemes can cause simulation error to propagate in complex ways. This is particularly true for Umbrella Sampling [2], a now standard scheme for estimating thermodynamic averages associated with rare events. In umbrella sampling, the computational effort is divided across separate simulations, each of which is biased to focus sampling onto a specific region of the system's configuration space. By looking at the overlapping sections of the separate simulations, one can determine how to reweight the data to remove the effect of the biases and recover averages against the unbiased distribution. This reweighting can lead statistical error to propagate through the calculation in complicated ways.

In Chapter 2, we cover a new algorithm for recombining data across umbrella sampling calculations, the *Eigenvector Method for Umbrella Sampling* (EMUS). EMUS allows one to directly propagate error estimates through the recombination procedure. Most immediately, this allows one to derive scaling arguments about the performance of umbrella sampling as a function of the method's various design choices. We show that averages estimated using umbrella sampling can converge exponentially faster than averages calculated via direct sampling. Moreover, we also correct the previously held belief that the efficiency of umbrella sampling always increases linearly with the number of simulations used. EMUS also yields estimates of the asymptotic variance of calculated averages. This quantifies the total error

in the umbrella sampling calculation. Error estimates can be decomposed into contributions from each individual window, allowing the practitioner to determine which simulation contributes the most to the overall error of the calculation.

While umbrella sampling allows accurate estimation of thermodynamic averages, the biasing procedure strongly modifies the dynamics. This prevents one from estimating dynamical quantities such as rates and committors. One approach to addressing this is to replace the biasing forces with intelligent splitting schemes [3]. However, this requires detailed control of the dynamics, as well as communication between each of the simulations used. Moreover, umbrella sampling schemes typically require knowledge of one or two *collective variables* that provide good control of the reaction; these are degrees of freedom known to drive the chemical process across states.

In Chapter 3, we give a new enhanced sampling scheme for dynamical quantities, the *Dynamical Galerkin Approximation*. In this scheme, we connect key dynamical quantities to the solutions of operator equations that use the system's dynamical operators. We expand the solution using a linear combination of basis functions, and estimate the resulting terms using empirical averages of correlation functions. This removes the need for harvesting full reactive pathways for the chemical process, which are difficult to construct numerically. Estimating the correlation functions only requires short unbiased trajectories distributed across the system's phase space, which can be gathered with minimal control over the dynamics. We show that this scheme is able to handle larger collective variable spaces, allowing it to be applied with less prior information.

While the previous chapters have focused on constructing quantitative descriptions of the system's dynamics, in Chapter 4 we investigate the question of constructing more qualitative descriptors of the dynamics. In particular, we consider the problem of finding slow subspaces of the system's dynamics: the function space spanned by the top K eigenfunctions of the system's transition operator. The *Variational Approach to Conformational* dynamics is a popular family of algorithms for finding this subspace, and includes commonly used algo-

rithms such as Markov State Modelling and Time-lagged Independent Component Analysis [4]. However, constructing a VAC algorithm may require multiple design choices, the effects of which are only partially understood. In particular, the accuracy of the calculated subspace depends on the interplay between the choice of basis functions used by the scheme, the amount of sampling error in estimated averages, and the choice of a parameter known as the lag time. We investigate the effect of sampling error on VAC estimates using a perturbation analysis and numerical experiments. This gives greater insight into the performance of VAC schemes, and helps inform the choice of lag time.

CHAPTER 2

EIGENVECTOR METHOD FOR UMBRELLA SAMPLING

Umbrella sampling efficiently yields equilibrium averages that depend on exploring rare states of a model by biasing simulations to windows of coordinate values and then combining the resulting data with physical weighting. Here, we introduce a mathematical framework that casts the step of combining the data as an eigenproblem. The advantage to this approach is that it facilitates error analysis. We discuss how the error scales with the number of windows. Then, we derive a central limit theorem for averages that are obtained from umbrella sampling. The central limit theorem suggests an estimator of the error contributions from individual windows, and we develop a simple and computationally inexpensive procedure for implementing it. We demonstrate this estimator for simulations of the alanine dipeptide and show that it emphasizes low free energy pathways between stable states in comparison to existing approaches for assessing error contributions. We discuss the possibility of using the estimator and, more generally, the eigenvector method for umbrella sampling to guide adaptation of the simulation parameters to accelerate convergence.

2.1 Introduction

One of the main uses of molecular simulations is the calculation of equilibrium averages. For understanding reaction processes, the free energy projected onto selected coordinates (collective variables) is of special interest. It relates directly to the probabilities of the coordinates taking particular values, and it can provide valuable information about the stable states, the barriers between them, and the origin of their stabilization. Furthermore, it is the starting point for most rate theories. Although in principle the free energy can be estimated from a long unbiased simulation, in practice doing so is challenging because bottlenecks slow the exploration of the configuration space. In other words, transitions between regions of the space are very infrequent in comparison to local fluctuations.

Various methods have been introduced to overcome this problem. Here, we consider one of the oldest and still most widely used such methods, umbrella sampling (US) [2]. In this approach, the collective-variable interval of interest is covered by a series of simulations, in each of which the system is biased such that sampling is restricted to a relatively narrow window of values of the collective variables. This can be accomplished by addition of a biasing potential that is small in the window and large outside it. The information from the different simulations must be combined, and the effect of the bias removed, to obtain the overall free energy profile. This requires consistently normalizing the probabilities in different windows, a task that is complicated by the fact that the simulations are run independently.

Considerable effort has been devoted to determining how best to combine the results from different simulations. Initially, researchers manually adjusted the zero of free energy in each window to make the full free energy profile continuous and, often, smooth; conflicting results arising from limited sampling at the window peripheries were removed. The desire to use all the simulation data motivated the introduction of estimators that allow for systematically combining the data from different simulations. By far, the most widely used of these in chemical physics applications is the weighted histogram analysis method (WHAM). The multistate Bennett acceptance ratio (MBAR) method, as it is referred to in the molecular-simulation literature and will be referred to here, is closely related but does not rely on binning the data [5, 6, 7, 8]. Both WHAM and MBAR can be derived from maximum-likelihood or minimum asymptotic variance principles assuming independent, identically distributed sampling in each window, and have corresponding statistical optimality properties under those conditions. Recent extensions seek to improve performance when the sampling is limited and to extend the algorithm to more general ensembles [9, 10].

In the present paper, we introduce an alternative scheme for estimating the free energy from US simulation data. In this approach, the normalization constants needed to combine information from separate simulation windows are the components of the eigenvector of a stochastic matrix that can be constructed from running averages in the windows. We thus

term our method Eigenvector Method for Umbrella Sampling (EMUS). The advantage of our method is that it lends itself to error analysis. Following previous work [11, 12, 13] we measure error with the asymptotic variance.

Our paper is organized as follows. After giving some background on US in Section 2.2, we formulate EMUS in Section 2.3. In Sections 2.4 and 2.5, we show that EMUS performs comparably to WHAM and MBAR, and discuss its connection with the latter. In Section 2.6, we use scaling arguments with simplifying assumptions to show that accounting for the error associated with combining the data is important and limits the speedups that can be achieved by increasing the number of simulation windows. In Section 2.7, we provide the full numerical analysis, which applies generally, without simplifying assumptions. Specifically, we derive a central limit theorem for averages from EMUS and use it to develop a means for estimating the error contributions from individual windows. We demonstrate the method for the free energy projected onto the ϕ and ψ dihedral angles of the alanine dipeptide and compare the error contributions with those from an estimator introduced by Zhu and Hummer [14]. We conclude in Section 2.9. The majority of this work was previously published in [15]. Section 2.8 was adapted from work in [16].

2.2 Background on Umbrella Sampling

Here, we review umbrella sampling and establish basic terms and notation. The goal is the calculation of an average of an observable g over a time-independent probability distribution π:

$$\langle g \rangle = \int g(x)\pi(x)dx. \tag{2.1}$$

At thermal equilibrium, π is the Boltzmann distribution:

$$\pi(x) = \frac{\exp\left(-H_0(x)/k_BT\right)}{\int \exp\left(-H_0(x)/k_BT\right) dx}, \tag{2.2}$$

where H_0 is the system Hamiltonian, k_B is Boltzmann's constant, and T is the temperature. In particular, we can express the the free energy difference between two states S_1 and S_2 as

$$\Delta G = -k_B T \ln \left(\frac{\langle \mathbb{1}_{S1} \rangle}{\langle \mathbb{1}_{S2} \rangle} \right), \tag{2.3}$$

where $\mathbb{1}$ is the indicator function

$$\mathbb{1}_S(x) = \begin{cases} 1 & \text{if } x \in S, \text{ and} \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

Similarly, the reversible work to constrain a collective variable $q(x)$ to a particular value q', also known as the potential of mean force (PMF), may be written as

$$W(q') = -k_B T \ln \left\langle \delta \left(q - q' \right) \right\rangle. \tag{2.5}$$

For complex systems, averages of the form in (2.1) must be evaluated numerically. Typically, this is done by generating a chain of related configurations, X_t, using Monte Carlo methods or molecular dynamics, and by assuming ergodicity. Namely, as the number of configurations N goes to infinity, $\langle g \rangle$ is the limit of the sample mean:

$$\bar{g} = \frac{1}{N} \sum_{t=0}^{N-1} g(X_t). \tag{2.6}$$

In all practical sampling methods, successive configurations are strongly correlated. While ergodicity guarantees that sample means converge to averages over π, convergence can be extremely slow if the correlation between subsequent points is strong. This is the case when sampling π relies on visiting low-probability states, such as transition states of chemical reactions.

US methods address this issue by enforcing sampling of different regions of configuration

space (windows), introducing L nonnegative *bias functions* ψ_i and then using L independent simulations to sample from the biased probability distributions

$$\pi_i(x) \propto \psi_i(x)\pi(x). \tag{2.7}$$

The essential idea is that sampling each π_i is fast because ψ_i is chosen so that relatively likely states under ψ_i are not separated by relatively unlikely states. This is accomplished by restricting the set of states on which ψ_i is non-negligible so that π is closer to constant on that set. In Section 2.6.3 we make this point more carefully by examining a regime in which umbrella sampling can be shown to be exponentially more efficient than direct simulation. A popular choice is to use bias functions that take a Gaussian form:

$$\psi_i(q) = \exp\left(-\frac{1}{2}k_i\left(q - q_i^0\right)^2 / k_B T\right), \tag{2.8}$$

such that

$$\pi_i(x) \propto \exp\left[-\left(H_0(x) + \frac{1}{2}k_i\left(q - q_i^0\right)^2\right) / k_B T\right]. \tag{2.9}$$

This corresponds to adding a harmonic potential centered at q_i^0 with spring constant k_i to the system Hamiltonian. We call the relative normalization constant (or partition function) of the i-th biased distribution z_i:

$$z_i = \frac{\int \psi_i(x)\pi(x)dx}{\sum_{k=1}^{L}\int \psi_k(x)\pi(x)dx}. \tag{2.10}$$

We also define the free energy in window i as

$$G_i = -k_B T \ln z_i. \tag{2.11}$$

We denote averages over the biased distributions by

$$\langle g \rangle_i = \int g(x)\pi_i(x)\,dx.$$

Overall averages of interest, $\langle g \rangle$, can be estimated as z_i-weighted sum of averages computed in each of the windows. We detail our prescription in the next section.

2.3 The Eigenvector Method for Umbrella Sampling

In this section, we present the Eigenvector Method for Umbrella Sampling (EMUS). We begin by defining

$$g^* \equiv \frac{g}{\sum_{k=1}^{L} \psi_k}.$$

for any function g. Then, we observe that

$$\begin{aligned}
\langle g \rangle &= \int g(x)\pi(x)\,dx \\
&= \int g(x) \left\{ \frac{\sum_{i=1}^{L} \psi_i(x) \left[\frac{\int \psi_i(x)\pi(x)dx}{\int \psi_i(x)\pi(x)dx} \right]}{\sum_{k=1}^{L} \psi_k(x)} \right\} \pi(x)\,dx \\
&= \sum_{i=1}^{L} \int \psi_i(x)\pi(x)dx \frac{\int g^*(x)\psi_i(x)\pi(x)dx}{\int \psi_i(x)\pi(x)dx} \\
&= \sum_{i=1}^{L} z_i \left(\sum_{k=1}^{L} \int \psi_k(x)\pi(x)dx \right) \langle g^* \rangle_i \,. \qquad (2.12)
\end{aligned}$$

The factor in parentheses can be taken out of the sum over i. To express this factor in terms of computable averages, we repeat the same steps with $g = 1$:

$$\sum_{k=1}^{L} \int \psi_k(x)\pi(x)dx = \frac{1}{\sum_{i=1}^{L} z_i \langle 1^* \rangle_i}. \qquad (2.13)$$

Substituting (2.13) into (2.12),

$$\langle g \rangle = \frac{\sum_{i=1}^{L} z_i \langle g^* \rangle_i}{\sum_{i=1}^{L} z_i \langle 1^* \rangle_i}. \tag{2.14}$$

Consequently, if we can evaluate the z_i, the $\langle 1^* \rangle_i$, and the $\langle g^* \rangle_i$ then we can assemble the original average $\langle g \rangle$ of interest. The averages $\langle g^* \rangle_i$ can be computed by sequences X_t^i (typically independent for each i) that sample π_i. Umbrella sampling methods differ primarily in how the z_i are computed.

To express the constants z_i in terms of averages over the biased distributions, we take $g(x) = \psi_j(x)$ in (2.12). Then, z_i solves

$$z_j = \sum_{i=1}^{L} z_i F_{ij}, \text{ where } F_{ij} = \left\langle \psi_j^* \right\rangle_i. \tag{2.15}$$

That is, the vector of normalization constants z is a left eigenvector of the matrix F with eigenvalue one. Under conditions to be elaborated upon in Section 2.3.2, the solution to (2.15) is uniquely specified when we notice that

$$\sum_{i=1}^{L} z_i = 1. \tag{2.16}$$

2.3.1 Computational Procedure

In the EMUS algorithm, we estimate the entries of F and the averages $\langle g^* \rangle_i$ and $\langle 1^* \rangle_i$ by sample means, then assemble the estimate of $\langle g \rangle$ using (2.14). To be precise, we denote the

sample means by

$$\bar{g}_i^* = \frac{1}{N_i} \sum_{t=0}^{N_i-1} \frac{g(X_t^i)}{\sum_k \psi_k(X_t^i)},$$
$$\bar{1}_i^* = \frac{1}{N_i} \sum_{t=0}^{N_i-1} \frac{1}{\sum_k \psi_k(X_t^i)}, \text{ and}$$
$$\bar{F}_{ij} = \frac{1}{N_i} \sum_{t=0}^{N_i-1} \frac{\psi_j(X_t^i)}{\sum_k \psi_k(X_t^i)}.$$

EMUS proceeds as follows:

1. Choose the biasing functions ψ_i.

2. Compute trajectories that sample states X_t^i from the biased distributions π_i.

3. Calculate the matrix $\bar{F}$ and the averages $\bar{g}_i^*$ and 1_i^*.

4. Calculate the vector of estimated normalization constants z^{EMUS} as the solution to

$$z_j^{\text{EMUS}} = \sum_{i=1}^{L} z_i^{\text{EMUS}} \bar{F}_{ij} \quad \text{with} \quad \sum_{i=1}^{L} z_i^{\text{EMUS}} = 1.$$

   We use QR factorization as given by Golub and Meyer [17]. See also Section 2.7.2.

5. Compute the estimate of $\langle g \rangle$:

$$\langle g \rangle^{\text{EMUS}} = \frac{\sum_{i=1}^{L} z_i^{\text{EMUS}} \bar{g}_i^*}{\sum_{i=1}^{L} z_i^{\text{EMUS}} \bar{1}_i^*} \tag{2.17}$$

   by substituting z^{EMUS} and the sample means in (2.14).

We remark that when one wishes to compute a free energy difference or a ratio of two observables, as in equation (2.3), it is not necessary to compute 1_i^*. Instead, one may use the formula

$$\frac{\langle \mathbb{1}_{S1} \rangle}{\langle \mathbb{1}_{S2} \rangle} = \frac{\sum_{i=1}^{L} z_i \langle \mathbb{1}_{S1} \rangle_i}{\sum_{i=1}^{L} z_i \langle \mathbb{1}_{S2} \rangle_i}$$

in place of (2.17).

2.3.2 The Eigenvector Problem

In this section, we give conditions under which the eigenvector problem has a unique solution. First, we show that F is a stochastic matrix; that is, each element F_{ij} is nonnegative and every row of F sums to one. For the latter,

$$\sum_{j=1}^{L} F_{ij} = \sum_{j=1}^{L} \left\langle \frac{\psi_j}{\sum_{k=1}^{L} \psi_k} \right\rangle_i = \left\langle \frac{\sum_{j=1}^{L} \psi_j}{\sum_{k=1}^{L} \psi_k} \right\rangle_i = 1.$$

The entries of F are nonnegative since we require that the bias functions be nonnegative. One can show that the matrix $\bar{F}$ is also stochastic by similar arguments.

A stochastic matrix J has a unique eigenvector with eigenvalue one if and only if it is irreducible: for every possible grouping of the indices into two distinct sets, A and B, $J_{ij} \neq 0$ for some $i \in A$ and $j \in B$ [18]. In fact, this statement remains true when J is non-negative with largest eigenvalue equal to one. For any such matrix we let $z(J)$ denote the continuous function returning the unique left eigenvector of J corresponding to eigenvalue one.

In the case of the particular stochastic matrix F defined in (2.15) these statements imply that if, for any division of the indices into sets A and B, there is sufficient overlap between the sets $\cup_{i\in A}\{x : \psi_i(x) > 0\}$ and $\cup_{j\in B}\{x : \psi_j(x) > 0\}$ then there will be a unique solution $z(F)$ to (2.15) which will necessarily equal the relative normalization constants z defined in (2.10). Because $z(J)$ is a continuous function of its arguments, $z^{\text{EMUS}} = z(\bar{F})$ converges to z as $\bar{F}$ converges to F. Consequently, EMUS produces a consistent estimator in the sense that if the sample averages used to estimate the entries F_{ij} and $\langle g^* \rangle_i$ converge (in probability or with probability one) to the true values, then the estimate of $\langle g \rangle$ also converges (in the same sense).

2.4 The Connection between EMUS and MBAR

Building upon earlier work in the statistics literature[5, 6, 19], Shirts and Chodera[8] suggested a class of algorithms for estimating free energy differences between states, which they termed MBAR. This method is similar to WHAM but does not require binning the simulation data to form histograms (see Tan *et al.*[11]). In this section, we explain the relation between EMUS and MBAR [8]. We also derive a new iterative method for solving the MBAR equations, and we show that our iteration leads naturally to a new family of related consistent estimators.

Shirts and Chodera's[8] starting point is the identity (see their (5))

$$z_j \sum_{i=1}^{L} \left\langle \alpha_{ij}(x)\psi_i(x)\pi(x) \right\rangle_j = \sum_{i=1}^{L} z_i \left\langle \alpha_{ij}(x)\psi_j(x)\pi(x) \right\rangle_i , \tag{2.18}$$

where $\alpha_{ij}(x)$ is an arbitrary function. They proposed the choice

$$\alpha_{ij}^{\mathrm{MBAR}}(x) = \frac{n_i/z_i}{\sum_k \psi_k(x)\pi(x)n_k/z_k}, \tag{2.19}$$

where n_i is the number of uncorrelated samples in window i. Substituting (2.19) into (2.18) gives

$$z_j = \sum_{i=1}^{L} z_i \left\langle \frac{\psi_j(x)n_i/z_i}{\sum_k \psi_k(x)n_k/z_k} \right\rangle_i . \tag{2.20}$$

We can cast (2.20) in a form reminiscent of EMUS by writing

$$z_j = \sum_{i=1}^{L} z_i F_{ij}(z), \tag{2.21}$$

where

$$F_{ij}(w) = \left\langle \frac{\psi_j(x)n_i/w_i}{\sum_k \psi_k(x)n_k/w_k} \right\rangle_i \tag{2.22}$$

for any vector w with positive entries. EMUS corresponds to setting $w = n$ so that

$$\alpha_{ij}^{\text{EMUS}}(x) = \frac{1}{\sum_k \pi(x)\psi_k(x)},$$

and (2.18) reduces to the eigenproblem (2.15).

In practice, one must replace the matrix $F_{ij}(w)$ in (2.22) by the sample mean approximation

$$\bar{F}_{ij}(w) = \frac{1}{N_i} \sum_{t=0}^{N_i-1} \left[\frac{\psi_j\left(X_t^i\right) n_i/w_i}{\sum_k \psi_k\left(X_t^i\right) n_k/w_k} \right]. \tag{2.23}$$

Substituting $\bar{F}_{ij}(z)$ for $F_{ij}(z)$ in (2.21) yields the equation

$$z_j^{\text{MBAR}} = \sum_{i=1}^{L} z_i^{\text{MBAR}} \bar{F}_{ij}(z^{\text{MBAR}}) \tag{2.24}$$

for z^{MBAR}, which we refer to here as the MBAR estimator. If the samples X_t^i are independent, MBAR is the nonparametric maximum-likelihood estimator of z [5].

In practice, the samples X_t^i are not independent for a given i, and the n_i must be estimated from data. Several algorithms for estimating the n_i have been proposed [20, 21] Shirts and Chodera[8]. base their estimates on the integrated autocorrelation times of physically-motivated coordinates, and we follow this common practice here. In fact, once the n_i have been estimated, Shirts and Chodera[8] suggest replacing sample averages over all N_i points by sample averages over the n_i points obtained by including only every N_i/n_i-th sample along the trajectory. We note that both the subsampling approach and the one in (2.23) correspond to approximations of expression (2.18) with (2.19), and we regard both as variations on the MBAR estimator. When the samples are independent, the two approaches are the same. In tests of the iterative EMUS algorithm introduced below, we find estimates to be insensitive to the choice of n_i and they can be set equal to 1, though in that case the estimator no longer corresponds directly to MBAR.

As written above, the MBAR estimator (2.24) resembles an eigenvector problem. How-

ever, the dependence of $\bar{F}(z)$ on z implies that the solution must be obtained self-consistently. The approach advocated by Shirts and Chodera for computing the MBAR estimator corresponds in the framework described here to solving (2.24) by a Newton-type iteration. However, the eigenvector form of (2.24) suggests an alternative approach. Rather than Newton's method, we employ the following algorithm:

1. As an initial guess for z^{MBAR}, choose a vector z^0 with positive entries. Estimate the n_i. Set $m = 0$.

2. (a) Calculate $\bar{F}_{ij}(z^m)$ according to (2.23).

   (b) Calculate a new estimate z^{m+1} of z^{MBAR} by solving the eigenproblem

$$z_j^{m+1} = \sum_{i=1}^{L} z_i^{m+1} \bar{F}_{ij}(z^m). \tag{2.25}$$

3. If $\max_i |z_i^{m+1} - z_i^m|/z_i^m >$ Tolerance,

   (a) Increment m;

   (b) Go to Step 2.

To show that this iteration makes sense, we must prove that the eigenproblem (2.25) always has a unique solution and that z^m converges to z^{MBAR} as m goes to infinity. To see that the eigenproblem has a solution, first observe that if $\bar{F}_{ij}(w)$ is irreducible for one vector with positive entries, w, then it is irreducible for all vectors with positive entries. When applying the EMUS method, we assume that both $F = F(n)$ and $\bar{F} = \bar{F}(n)$ are irreducible. Thus, we may assume that $\bar{F}(w)$ is irreducible for any w. Moreover, observe that for any positive vector w, the vector with entries n_i/w_i is a right eigenvector of $\bar{F}(w)$ with eigenvalue one and positive entries. It follows from the Perron-Frobenius theorem that the matrix $\bar{F}(w)$ has a unique left eigenvector $z(\bar{F}(w))$ with eigenvalue one and that $z(\bar{F}(w))$ has positive entries. Thus, the eigenproblem always has a unique solution. We do not have

a proof that the iterates converge. However, since z^{MBAR} is a fixed point of the iteration, if the iterates do converge their limit must be z^{MBAR}. In practice, we find that the iteration converges quickly, usually to a relative error of 10^{-6} within 10 iterates.

In addition to its apparently rapid convergence, another argument in favor of the algorithm that we introduce above for solving (2.24) is that each iteration of the scheme results in a new consistent estimator. We will use the term iterative EMUS to refer to this family of estimators. With the initial guess $z^0 = n$, the result, z^1, of the first iteration is the EMUS estimator defined in Section 2.3. In Supplement 2.10, we show that for any fixed finite number of iterations m, z^m is also a consistent estimator of the vector z of normalization constants. By contrast, other schemes, such as Newton's method, for solving (2.24) may require that the number of iterations goes to infinity to obtain a consistent estimate. We also remark that the consistency result in Supplement 2.10 holds as long as the n_i converge to non-random, positive values with increasing numbers of samples N_i. They can be chosen as described above, or simply set to a fixed value.

Differences between the iterative EMUS scheme above and the application of Newton's method proposed by Shirts and Chodera [8] are mostly matters of implementation. As we will see in the next section, the results are not very sensitive to these computational details; most of the accuracy in the iterative EMUS approach is achieved in the first step. In any case, we remind the reader that the primary goal of this paper is to characterize those properties of the broader umbrella sampling approach that are essential to its success, not to analyze details of implementation.

While we focus here on potentials of mean force, the MBAR estimator has been applied to a broader category of free energy problems, including the analysis of single-molecule pulling experiments and alchemical free energy calculations [8, 22, 23]. The close relation between EMUS and MBAR indicates that error analysis of EMUS may provide insight into the sources of error in MBAR for these problems, but we do not pursue this idea further in the present work.

2.5 Numerical Comparison

To test the algorithm numerically, we performed 100 independent umbrella sampling calculations for the PMF of the ϕ coordinate of the alanine dipeptide (i.e., N-acetyl-alanyl-N'-methylamide) in vacuum. Simulations were run using GROMACS version 5.1.1 with harmonic bias potentials applied using the PLUMED 2.2.0 software package [24, 25]. The molecule was represented by the CHARMM 27 force field without CMAP corrections [26] with covalent bonds to hydrogen atoms constrained by the SHAKE algorithm [27]. Twenty windows were evenly spaced along the ϕ dihedral angle. The force constant $k_i = 0.00760535 \times 10^{-2}$ kcal mol^{-1} degree^{-2} such that the standard deviation of the Gaussian bias functions was 9°. In each window, we integrated the equations of motion with the GROMACS leap-frog Langevin integrator with a 1 fs time step. The system was equilibrated for 40 ps and then sampled for 100 ps, saving structures every 10 fs.

The data was then analyzed with EMUS, Grossfield's implementation of WHAM [28], and the algorithm proposed by Zhu and Hummer (ZH) (see equation A1 and following discussion in Supplement A of the reference [14]. The data was also analyzed with pyMBAR [8]; as pyMBAR gave results virtually identical to WHAM, the results are not shown. In Figure 2.1, we show the resulting average potentials of mean force, as well as the standard deviation of the estimates over the 100 runs. WHAM and EMUS converge to the same result. This is to be expected, as both algorithms are consistent (i.e., they converge to the exact result as the amount of samples in each window tends to infinity; see Section 2.7), although WHAM exhibits a small bias from the binning of data for the histograms [8]. Although the standard deviations of the free energies are generally higher for EMUS than for WHAM, they are of comparable magnitude. Compared to the ZH algorithm, EMUS also has a higher standard deviation. However, ZH is based on thermodynamic integration and uses the trapezoidal rule to calculate free energy differences between windows [14]. As a consequence it suffers from noticeable quadrature error [29], which causes the barrier height to converge to an artificially low value.

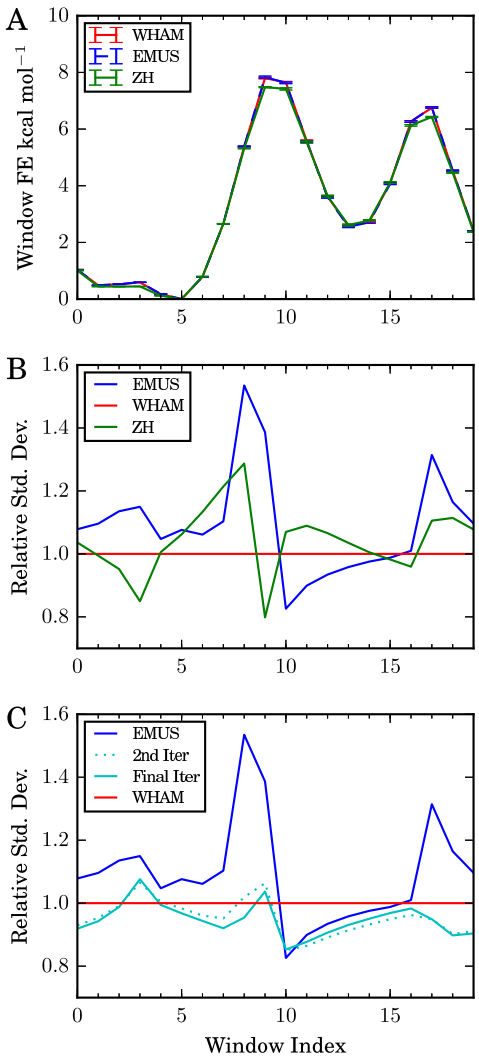


Figure 2.1: Comparison of umbrella sampling methods applied to simulation data for the alanine dipeptide. (A) Average window free energies, G_i, for the indicated methods. Error bars are estimated standard deviations of the means. (B) Standard deviation of each method relative to that of the WHAM algorithm. Colors are the same as in (A). (C) EMUS as the first step in a self-consistent iteration to solve the MBAR equations (see text). The number of uncorrelated samples in each window (n_i) was estimated by calculating the integrated autocorrelation of the ϕ dihedral angle from each trajectory. Results shown are for identical molecular dynamics data (see text for simulation details); the methods differ only with respect to combination of the data to estimate the free energies.

In Figure 2.1 C, we apply the self-consistent iteration. For this calculation, we estimate the number of independent samples in each window (n_i) from the integrated autocorrelation time of the ϕ dihedral angle time series. We plot the standard deviation of the values of z calculated after the first iteration (EMUS), the second iteration, and after convergence to a relative residual smaller than 10^{-6}. In general, convergence is achieved after an average of 9 iterations; none of the 100 data sets required more than 15 iterations. However, we note that after two iterations, the estimates of z already have a standard deviation equivalent to that of the WHAM algorithm.

2.6 Justification for umbrella sampling by scaling arguments

The quality of a statistical estimate from umbrella sampling depends strongly on the choices made for the simulation windows. In this section, we discuss how the error scales as properties of the simulation change. We begin in Subsection 2.6.1 with a description of a prevalent justification for the use of US. We show in Subsection 2.6.2 that this argument is incomplete and, in turn, misleading. In Subsection 2.6.3, we provide an alternative justification; namely, we show that in the low temperature limit, the cost to achieve a fixed accuracy by US grows slowly compared to direct simulation. In this Section we make several simplifying assumptions that allow us to draw precise conclusions about the scaling properties of EMUS. In Section 2.7 we provide error bounds for EMUS under much more general assumptions.

2.6.1 Scaling in the Limit of Many Windows

To justify umbrella sampling, it is often suggested that the total computational time required to accurately sample statistics is inversely proportional to the number of windows, L[30, 20, 31, 32, 33]. The argument for this scaling proceeds as follows.

- Divide a one-dimensional collective variable space into L windows of equal length, inversely proportional to L (i.e., L^{-1}).

- Assume the windows are small enough that no free energy barriers exist in each window. The time to explore a window should be diffusion limited and proportional to the length of the window squared. Therefore, the simulation time required to accurately sample statistics in one window is also proportional to L^{-2}.

- Because there are L windows, the total simulation time required to compute averages to fixed accuracy should scale as $L \times L^{-2} = L^{-1}$.

While this argument is now standard [30, 20, 31, 32, 33], Virnau and Müller [34] observed that the error for computing the free energy difference between phases of Lennard-Jones particles with an approximately fixed amount of sampling was insensitive to the number of windows in practice, and they noted that the argument above neglects the error associated with combining the data from different simulation windows. This intuition is supported by our analysis in the next subsection, which shows that the total computational cost to achieve a fixed accuracy should be insensitive to the choice of L, so long as it is sufficiently large.

2.6.2 A Simple Model Problem

To perform a more precise analysis, we make a number of simplifying assumptions. We emphasize that these assumptions are in force only for the purposes of the scaling arguments in this section. We provide more general error bounds for EMUS in Section 2.7.

Assumption 2.6.1. *The total computation time, N, is divided equally among the windows such that $N_i = N/L$.*

Assumption 2.6.2. *The ψ_i are functions on the one-dimensional interval $[0, 1]$, and the set of points where ψ_i is non-zero, $\{q : \psi_i > 0\}$, is an interval of length $|\{q : \psi_i > 0\}| \leq \gamma/L$. We also assume that $\psi_i \psi_j = 0$ unless $|j - i| \leq 1$. Consequently, both the exact matrix F and the sample mean $\bar{F}$ are tri-diagonal.* This assumption clearly does not hold when the bias functions ψ_i are Gaussian. Nonetheless, the rapid decay of Gaussian bias functions away from their peaks guarantees that entries of F and $\bar{F}$ far from the diagonal are very small,

such that we expect our conclusions to still hold (though their justification would be more complicated).

Assumption 2.6.3. *The overlap of ψ_i and $\psi_{i\pm1}$ (i.e., the integral of their product) is large enough that*

$$\min\{F_{i,i+1}, F_{i,i-1}\} > \delta > 0 \tag{2.26}$$

for all L and for all $i \leq L$. If our last assumption holds, but this one does not, then we can find more than one vector z satisfying equations (2.15) and (2.16). This assumption is a slightly stronger version of the notion of irreducibility that we defined earlier (see Section 2.3.2). Note that we require the irreducibility to hold uniformly in the large L limit, and we thus introduce the δ, which is independent of L.

Assumption 2.6.4. *Sample averages computed in* different *windows are independent, i.e., $\bar{F}_{i,i\pm1}$ and $\bar{F}_{j,j\pm1}$ for $j \neq i$ are independent.* We do *not* assume (here or anywhere else in this paper) that samples generated within a single window are independent. Indeed, even if the samples from π_i are independent, $\bar{F}_{i,i+1}$ and $\bar{F}_{i,i-1}$ are dependent random variables.

As an example average, let us consider the error in the free energy difference between the first and last windows:

$$\Delta\bar{G}_{L,1} = -k_B T \ln\left(\frac{z_L}{z_1}\right). \tag{2.27}$$

Assumption 2.6.2 is sufficient for $z(\bar{F})$ to be in detailed balance with $\bar{F}$ (Kelly [35], Lemma 1.5 and Section 1.3):

$$z_{i+1}(\bar{F}) = z_i(\bar{F})\frac{\bar{F}_{i,i+1}}{\bar{F}_{i+1,i}}. \tag{2.28}$$

Using (2.28) recursively,

$$\begin{aligned}\Delta\bar{G}_{L,1} &= -k_B T \ln\left(\frac{\prod_{i=1}^{L-1}\bar{F}_{i,i+1}}{\prod_{i=1}^{L-1}\bar{F}_{i+1,i}}\right) \\ &= -k_B T \ln \bar{F}_{1,2} + k_B T \ln \bar{F}_{L,L-1} + k_B T \sum_{i=2}^{L-1} \ln\left(\frac{\bar{F}_{i,i+1}}{\bar{F}_{i,i-1}}\right).\end{aligned} \tag{2.29}$$

To understand the error (variance) of the terms in (2.29), we must further specify $F_{i,i+1}$ and $F_{i,i-1}$.

Assumption 2.6.5. *For N_{min} and L_{min} sufficiently large, when $N_i \geq N_{min}$ and $L \geq L_{min}$,*

$$\frac{K_{min}}{N_i L^2} \leq \text{var}\left(\ln\left(\frac{\bar{F}_{i,i+1}}{\bar{F}_{i,i-1}}\right)\right) \leq \frac{K_{max}}{N_i L^2} \tag{2.30}$$

for $i = 2, 3, \ldots, L-1$, and the same upper and lower bounds hold for $\text{var}\left(\ln F_{1,2}\right)$ *and* $\text{var}\left(\ln F_{L,L-1}\right)$. This is just a precise interpretation of the diffusion limited sampling assumption made in the standard justification of US reproduced in the last subsection. Under such an assumption we expect both $\bar{F}_{i,i+1}$ and $\bar{F}_{i,i-1}$ to have variance on the order of $1/(N_i L^2)$ and, in light of (2.26), the function $\ln(x/y)$ is smooth near $(x, y) = (F_{i,i+1}, F_{i,i-1})$. These considerations are closely related to Lemma 2.7.2 in the next section.

With all the assumptions in hand, we now complete the argument by taking the variance of both sides of (2.29). Since samples from different windows are independent, the variance of $\bar{G}_{L,1}$ is a sum of contributions from each window:

$$\text{var}\left(\Delta\bar{G}_{L,1}\right) = \text{var}\left(k_B T \ln \bar{F}_{1,2}\right) + \text{var}\left(k_B T \ln \bar{F}_{L,L-1}\right) + \sum_{i=2}^{L-1} \text{var}\left(k_B T \ln\left(\frac{\bar{F}_{i,i+1}}{\bar{F}_{i,i-1}}\right)\right). \tag{2.31}$$

Using (2.30) and substituting $N_i = N/L$, we find that, as long as $N/L \geq N_{min}$ and $L \geq L_{min}$,

$$(k_B T)^2 \frac{K_{min}}{N} \leq \text{var}\left(\Delta\bar{G}_{L,1}\right) \leq (k_B T)^2 \frac{K_{max}}{N}. \tag{2.32}$$

To verify that this conclusion carries over to harmonic bias potentials, we performed multiple umbrella sampling calculations for a Brownian particle on a flat potential on the interval $[0, 1]$ with a stepsize of 1.0×10^{-6} and $k_B T = 1.0$ using Gaussian bias functions with a standard deviation of $1/L$. The number of windows was varied from $L = 10$ to 46 in steps of 2. For each value, a total of 10^7 steps were distributed equally in the windows and the US calculation was repeated 480 times. We then calculated the mean square error

of the free energy difference between the first and last window over the 480 replicates and determined how the mean square error scaled with L. An L^{-1} scaling would predict that mean square error would decrease inversely with the number of windows used. By contrast, the data plotted in Figure 2.2 support a scaling of L^0, consistent with (2.32).

It is worth noting that the inverse scaling with total cost N in (2.32) is exactly the scaling one would expect for the variance of an estimate of the free energy difference $\Delta\bar{G}_{L,1}$ constructed from a molecular dynamics trajectory of length N. Because US and direct simulations of comparable total numbers of steps require comparable computational effort (ignoring the overhead associated with combining the simulation data, which is typically small in comparison with the computational cost of the sampling), the benefits of US must be encoded in the constants K_{min} and K_{max}. A dramatic demonstration of this observation is the purpose of the next subsection.

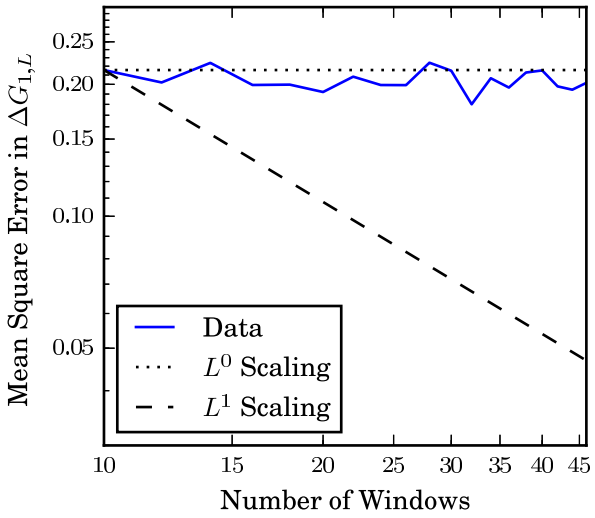


Figure 2.2: The scaling of umbrella sampling error with number of windows on a flat potential. A Brownian particle on a flat, one-dimensional potential was simulated for 480 identical runs, and the free energy difference between the first and last windows was calculated, as described in the text. Here, the mean square error from the exact result is plotted against the number of windows. The lines show the scaling in error predicted by the L^{-1} and L^0 scalings. Fitting the data on a log-log scale gives a scaling exponent of -0.026 ± 0.028.

2.6.3 The Low Temperature Limit

To understand the benefits of umbrella sampling, we must study its performance when free energy barriers are large in comparison with the temperature, i.e., the low temperature limit ($k_BT \to 0$). In this limit, the cost of direct sampling increases exponentially with $1/T$, while, as we show, the cost of umbrella sampling increases only algebraically. A formal discussion is given in Reference [16]; here, we present a simple plausibility argument.

Owing to the free energy barriers, the assumption of diffusive dynamics in each window no longer holds. Instead, we expect a form typical of reaction rate theories in each window. We define ΔW_i as the maximum difference in the PMF in window i:

$$\Delta W_i = \max_{\{q:\, \psi_i(q)>0\}} \{W(q)\} - \min_{\{q:\, \psi_i(q)>0\}} \{W(q)\}. \tag{2.33}$$

Assumption 2.6.5′. *We now replace the upper and lower bounds in* (2.30) *by the upper bound*

$$\mathrm{var}\left(\ln\left(\frac{\bar{F}_{i,i+1}}{\bar{F}_{i,i-1}}\right)\right) \le \frac{K}{k_BTN_iL^2}\exp\left(\frac{\Delta W_i}{k_BT}\right) \tag{2.34}$$

for $i = 2, 3, \ldots, L-1$ *with analogous replacements for* $i = 1$ *and* $i = L$*, as long as* $N_i \ge N_{min}$ *and* $L \ge L_{min}$*. The constant* K *here is assumed to be independent of temperature.* This bound captures the diffusion limited sampling assumption when L is very large, but is more detailed than (2.30) in that it captures (crudely) the increasing difficulty of the sampling problem as the temperature decreases with all other parameters held fixed. Under reasonable additional assumptions on the underlying potential, the bias functions ψ_i, and the sampling scheme, one can rigorously establish an asymptotic (large N_i) bound of the form in (2.34) [16].

Substituting this new bound into (2.31), we find that, if $L \ge L_{min}$ and $N/L \ge N_{min}$, then

$$\mathrm{var}\left(\Delta\bar{G}_{L,1}\right) \le \frac{Kk_BT}{NL}\sum_{i=1}^{L}\exp\left(\frac{\Delta W_i}{k_BT}\right). \tag{2.35}$$

As the temperature decreases, we choose to increase L such that $\Delta W_i/k_BT$ is bounded above. This can be achieved by scaling L linearly with $1/T$: if the derivative of the PMF is bounded (in absolute value) by W'_{max}, choosing L so that

$$L \geq \frac{W'_{max}}{\Omega k_B T} \tag{2.36}$$

ensures that $\Delta W_i/k_BT$ is bounded by Ω (since we have assumed that the argument of W is in $[0,1]$). On the other hand, our assumption that the length of $\{q : \psi_i > 0\}$ (Assumption 2.6.2) does not exceed γ/L implies that

$$\frac{\Delta W_i}{k_BT} \leq \frac{W'_{max}\gamma}{k_BTL}. \tag{2.37}$$

Consequently, as long as (2.36) holds,

$$\frac{\Delta W_i}{k_BT} \leq \Omega\gamma.$$

Finally, substituting this result into (2.35) we find that if $L \geq L_{min}$ and $N/L \geq N_{min}$, then

$$\text{var}\left(\Delta\bar{G}_{L,1}\right) \leq \frac{Kk_BT\exp(\Omega\gamma)}{N} \leq \frac{KLk_BT\exp(\Omega\gamma)}{N_{min}}.$$

With the best possible (smallest) choice of L allowed by (2.36), this bound becomes

$$\text{var}\left(\Delta\bar{G}_{L,1}\right) \leq \frac{KW'_{max}\exp(\Omega\gamma)}{\Omega N_{min}}. \tag{2.38}$$

The remarkable feature of the bound in (2.38) is that it is independent of T. This does not mean that the cost to achieve a fixed accuracy is independent of T. However, it does imply that as the temperature is decreased we do not have to increase N_{min} to maintain a fixed accuracy. Expression (2.36) and the fact that $N_i \geq N_{min}$ together imply that the computational cost of obtaining an accurate estimate of $\Delta\bar{G}_{L,1}$ by US increases algebraically

with $(k_B T)^{-1}$. That scaling is to be compared to exponential in $(k_B T)^{-1}$ to achieve the same accuracy by direct simulation.

2.7 Analysis of the Error of EMUS

In this section, we study the error of EMUS in full generality, without imposing the simplifying assumptions of the previous section. Our main results are a central limit theorem for EMUS (Theorem 2.7.4 below) and an easily computed, practical error estimator which reveals the contributions of the different windows to the total error. These results may be used to compare the efficiency of EMUS and other methods and to study how the efficiency of EMUS depends on parameters such as the number of samples allocated to each window.

2.7.1 A Central Limit Theorem for EMUS

Before developing the error analysis, we define a single notation for EMUS which incorporates both the case of a free-energy difference and the case of an ensemble average. In either case, one must compute $\bar{F}$ and also $\bar{g}^*_{1,i}$ and $\bar{g}^*_{2,i}$ for two real valued functions g_1 and g_2. To compute a free energy difference, we choose based on (2.3)

$$g_1 = \mathbb{1}_{S1} \text{ and } g_2 = \mathbb{1}_{S2}.$$

To compute an ensemble average $\langle g \rangle$, we choose based on (2.14)

$$g_1 = g \text{ and } g_2 = 1.$$

We furthermore define the function

$$v_i(x) = \left(\psi^*_1(x), \ldots, \psi^*_L(x), g^*_{1,i}(x), g^*_{2,i}(x)\right) \tag{2.39}$$

so that

$$\bar{v}_i = \frac{1}{N_i} \sum_{t=0}^{N_i-1} v_i(X_t^i) = \left(\bar{F}_{i1}, \ldots, \bar{F}_{iL}, \bar{g}_{1,i}^*, \bar{g}_{2,i}^* \right),$$

where we remind the reader that, for each i, the process X_t^i samples the biased distribution π_i. Define

$$\bar{v} = (\bar{v}_1, \ldots, \bar{v}_L),$$

and let

$$\langle v \rangle = (\langle v_1 \rangle_1, \ldots, \langle v_L \rangle_L)$$

denote the corresponding vector of exact averages. The EMUS estimator takes the form $B(\bar{v})$, where for a free-energy difference,

$$B(\bar{v}) = -kT \log \left(\frac{\sum_{i=1}^{L} z_i(\bar{F}) \bar{g}_{1,i}^*}{\sum_{i=1}^{L} z_i(\bar{F}) \bar{g}_{2,i}^*} \right), \tag{2.40}$$

and for an ensemble average,

$$B(\bar{v}) = \frac{\sum_{i=1}^{L} z_i(\bar{F}) \bar{g}_{1,i}^*}{\sum_{i=1}^{L} z_i(\bar{F}) \bar{g}_{2,i}^*}. \tag{2.41}$$

We now proceed with the error analysis. First, we characterize the error of the sample means over the biased distributions. As discussed by Frenkel and Smit [20, Supplement D], the variance of a sample mean may be expanded in terms of the integrated autocovariance of the process. We define the autocovariance function of $v_i(X_t^i)$ to be

$$C_i(t) = \left\langle (v_i(X_0^i) - \langle v_i \rangle_i)(v_i(X_t^i) - \langle v_i \rangle_i)^{\mathrm{T}} \right\rangle_i,$$

where $^{\mathrm{T}}$ denotes a vector transpose, and here the outer $\langle \ldots \rangle_i$ denotes the exact average not only over X_0^i sampled from π_i but also subsequent points of the sequence X_t^i. Note that

$C_i(t)$ is a $(L+2) \times (L+2)$ matrix. We define the integrated autocovariance to be

$$\Sigma_i = \sum_{t=-\infty}^{\infty} C_i(t). \tag{2.42}$$

The integrated autocovariance is the leading order coefficient in an expansion of the covariance $\bar{v}_i$ (see Frenkel and Smit [20, D.1.3]):

$$\operatorname{cov}(\bar{v}_i) = \frac{\Sigma_i}{N_i} + o\left(\frac{1}{N_i}\right), \tag{2.43}$$

where $o(1/N_i)$ denotes terms that go to zero faster than N_i (i.e., $N_i\, o(1/N_i) \to 0$).

Under certain conditions on the process X_t^i, one can strengthen the expansion of the covariance (2.43) to a central limit theorem (CLT) for $\bar{v}_i$. We expect such a CLT to hold for most problems and most sampling methods in computational statistical physics. However, to avoid a lengthy and technical digression, we simply take the CLT as an assumption; this assumption is justified in more detail in Reference [16], and we refer to Lelièvre *et al.* [36, Section 2.3.1.2] for a general discussion of the CLT in the context of computational statistical physics.

Assumption 2.7.1 (Central Limit Theorem for $\bar{v}_i$)**.** *We assume that*

$$\sqrt{N_i}\left(\bar{v}_i - \langle v_i \rangle\right) \xrightarrow{\mathrm{d}} \mathbf{N}\left(0, \Sigma_i\right) \tag{2.44}$$

where $\Sigma_i \in \mathbb{R}^{(L+2)\times(L+2)}$ is the integrated autocovariance matrix defined in (2.42)*. The symbol $\xrightarrow{\mathrm{d}}$ denotes convergence in distribution as $N_i \to \infty$.* Notice that when the elements of the sequence X_t^i are independent and drawn from π_i then $\Sigma_i = \left\langle (v_i - \langle v_i \rangle_i)(v_i - \langle v_i \rangle_i)^{\mathrm{T}} \right\rangle_i / N_i$. More generally, samples are correlated, so Σ_i includes a factor that accounts for the time to decorrelate.

Having characterized the errors in the sample means, we now study how these errors propagate through the EMUS algorithm. Our goal is to prove a CLT for EMUS. We accomplish

this using the delta method.

Lemma 2.7.2 (The Delta Method; Proposition 6.2 of Bilodeau and Brenner [37])**.** *Let θ_N be a sequence of random variables taking values in $\mathbb{R}^d$. Assume that a central limit theorem holds for θ_N with mean $\mu \in \mathbb{R}^d$ and asymptotic covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$; that is, assume*

$$\sqrt{N}(\theta_N - \mu) \xrightarrow{\mathrm{d}} \mathbf{N}(0, \Sigma).$$

Let $\Phi : \mathbb{R}^d \to \mathbb{R}$ be a function differentiable at μ. Then we have the central limit theorem

$$\sqrt{N}\left(\Phi\left(\theta_N\right) - \Phi\left(\mu\right)\right) \xrightarrow{\mathrm{d}} \mathbf{N}\left(0, \nabla\Phi^{\mathrm{T}}\left(\mu\right)\Sigma\nabla\Phi\left(\mu\right)\right)$$

for the sequence of random variables $\Phi(\theta_N)$.

To motivate the delta method, we observe that if X has distribution $\mathbf{N}(\mu, \Sigma)$, then $\nabla\Phi(\mu)^{\mathrm{T}}X$ has distribution $\mathbf{N}\left(\Phi(\mu), \nabla\Phi^{\mathrm{T}}\left(\mu\right)\Sigma\nabla\Phi\left(\mu\right)\right)$. That is, according to the delta method, the asymptotic distribution of $\Phi(X)$ is the linearization of Φ at μ applied to the asymptotic distribution of X. Thus, one may regard the delta method as a rigorous version of the standard error propagation formula based on linearization.

We prove the CLT for EMUS by applying the delta method with $\bar{v}$ taking the place of θ_N and with the function B taking the place of Φ. We require the following assumptions in addition to Assumption 2.7.1.

Assumption 2.7.3. *We assume:*

1. *The proportion of the total number of samples drawn from each window is constant in the limit as $N \to \infty$; that is,*

$$\lim_{N\to\infty} N_i/N = \kappa_i. \tag{2.45}$$

2. *Sampling in different windows is independent; that is, $\bar{v}_i$ is independent of $\bar{v}_j$ when $j \neq i$.*

3. The biasing functions ψ_i are chosen so that F is irreducible; see Section 2.3.2.

We now give the CLT for EMUS.

Theorem 2.7.4 (Central Limit Theorem for EMUS)**.** *Let Assumptions 2.7.1 and 2.7.3 hold. Let*

$$\frac{\partial B}{\partial \bar{v}_i} = \left(\frac{\partial B}{\partial \bar{F}_{i1}}, \ldots, \frac{\partial B}{\partial \bar{F}_{iL}}, \frac{\partial B}{\partial \bar{g}^*_{1,i}}, \frac{\partial B}{\partial \bar{g}^*_{2,i}} \right) \in \mathbb{R}^{L+2}$$

denote the partial derivative of B with respect to $\bar{v}_i$, evaluated at v. Under the assumptions stated above,

$$\sqrt{N}\left(B\left(\bar{v}\right) - B\left(\langle v \rangle\right)\right) \xrightarrow{\mathrm{d}} \mathbf{N}(0, \sigma^2),$$

where

$$\sigma^2 = \sum_{i=1}^{L} \frac{1}{\kappa_i} \left(\frac{\partial B}{\partial v_i}^{\mathrm{T}} \Sigma_i \frac{\partial B}{\partial v_i} \right). \tag{2.46}$$

We refer to σ^2 as the asymptotic variance of EMUS.

Proof. First, we write down a central limit theorem for $(\bar{v}_1, \ldots, \bar{v}_L)$. We have that

$$\sqrt{N}(\bar{v}_i - \langle v_i \rangle_i) \xrightarrow{\mathrm{d}} \mathbf{N}(0, \kappa_i^{-1}\Sigma_i) \tag{2.47}$$

by Assumption 2.7.1 and (2.45). Since the sampling in different windows is assumed to be independent, (2.47) implies

$$\sqrt{N}(\bar{v} - \langle v \rangle) \xrightarrow{\mathrm{d}} \mathbf{N}(0, \Sigma),$$

where $\Sigma \in \mathbb{R}^{L(L+2)\times L(L+2)}$ is the block diagonal matrix

$$\Sigma = \left[\begin{array}{c|c|c|c} \Sigma_1/\kappa_1 & 0 & 0 & \ldots \\ \hline 0 & \Sigma_2/\kappa_2 & 0 & \ldots \\ \hline 0 & 0 & \Sigma_3/\kappa_3 & \ldots \\ \hline \vdots & \vdots & \vdots & \ddots \end{array}\right].$$

Second, we verify that B is differentiable at $\bar{v}$. Since F is assumed to be an irreducible

stochastic matrix, $z(\bar{F})$ is differentiable at $\bar{F}$. We refer to Thiede *et al.* [38] Lemma 3.1 for a complete explanation. It follows from the chain rule that B is differentiable at $\bar{v}$.

Finally, applying Lemma 2.7.2 with B playing the role of Φ and $\bar{v}$ the role of θ_N concludes the proof. □

The asymptotic variance σ^2 appearing in Theorem 2.7.4 measures the rate at which the error of EMUS decreases with the number of samples. To make this precise, we observe that Theorem 2.7.4 is equivalent to the following asymptotic result concerning confidence intervals. For every $\alpha > 0$,

$$\lim_{N\to\infty} \mathbf{P}\left[|B(\bar{v}) - B(\langle v\rangle)| \leq \frac{\alpha\sigma}{\sqrt{N}}\right] = \operatorname{erf}\left(\frac{\alpha}{\sqrt{2}}\right), \tag{2.48}$$

where $\mathbf{P}$ denotes a probability and erf denotes the error function.

The asymptotic variance is commonly used to measure the efficiency of an estimator. We refer to van der Vaart [39] for an explanation and for a discussion of other possibilities. In Section 2.7.2, we explain how the proportion κ_i of samples allocated to each window may be adjusted to minimize the asymptotic variance of EMUS, thereby maximizing efficiency.

We note that a central limit theorem similar to Theorem 2.7.4 has been proved for the MBAR estimator by Gill *et al.* [6, Proposition 2.2]. However, the authors of this work do not study the dependence of the asymptotic variance on the parameters, as we do. In fact, the MBAR estimator is significantly more complicated than EMUS, and its dependence on the number of windows and the allocation of samples is harder to understand.

Reference [16] use a result similar to Theorem 2.7.4 to generalize the conclusions of Section 2.6 to periodic and multi-dimensional reaction coordinates and to a wider class of observables than free energy differences. We show both that the asymptotic variance is constant in the limit of large L and that the work required to compute an average to fixed precision increases only algebraically in the low temperature limit. In addition, we use recently developed perturbation estimates for Markov chains [38] to quantify the dependence

of the asymptotic variance of EMUS on the degree to which the bias functions overlap.

2.7.2 Estimating the Asymptotic Variance of EMUS

Our goal in this section is to derive a computable estimate $\bar{\sigma}^2$ of the asymptotic variance σ^2, which can be decomposed to assess the contributions from individual windows to errors in averages. We recall that formula (2.46) for σ^2 involves partial derivatives of B. Our estimate $\bar{\sigma}^2$ of σ^2 requires explicit formulas for these partial derivatives. We provide the appropriate expressions, both for ensemble averages and for free-energy differences, in Lemma 2.7.5. Following the partial derivatives, we present an algorithm for evaluating $\bar{\sigma}^2$ and demonstrate it for the alanine dipeptide. Finally, we compare with the output of a procedure from Zhu and Hummer (ZH) [14] in Section 2.7.2.

Lemma 2.7.5. *We have the following formulas for $\partial B/\partial \bar{v}_i$:*

1. *When EMUS is used to compute an ensemble average $\langle f \rangle$, B is defined by* (2.41)*, and we have*

$$\begin{aligned}
\frac{\partial B}{\partial \bar{F}_{ij}}(\bar{v}) &= \frac{\sum_k z_i(\bar{F})(I-\bar{F})^{\#}_{jk}(\bar{g}^*_k - B(\bar{v})\bar{1}^*_k)}{\sum_k z_k(\bar{F})\bar{1}^*_k}, \\
\frac{\partial B}{\partial \bar{g}^*_{1,i}}(\bar{v}) &= \frac{z_i(\bar{F})}{\sum_k z_k(\bar{F})\bar{1}^*_k}, \text{ and} \\
\frac{\partial B}{\partial \bar{g}^*_{2,i}}(\bar{v}) &= -\frac{B(\bar{v})z_i(\bar{F})}{\sum_k z_k(\bar{F})\bar{1}^*_k}.
\end{aligned}$$

2. *When EMUS is used to compute a free-energy difference, B is defined by* (2.40)*, and*

we have

$$\frac{\partial B}{\partial \bar{F}_{ij}}(\bar{v}) = kTz_i(\bar{F})\left(\frac{\sum_k (I-F)^{\#}_{jk}\mathbb{1}^*_{S2,k}}{\sum_k z_k(\bar{F})\mathbb{1}^*_{S2,k}} - \frac{\sum_k (I-F)^{\#}_{jk}\mathbb{1}^*_{S1,k}}{\sum_k z_k(\bar{F})\mathbb{1}^*_{S1,k}}\right),$$
$$\frac{\partial B}{\partial \bar{g}^*_{1,i}}(\bar{v}) = kT\frac{z_i(\bar{F})}{\sum_k z_k(\bar{F})\mathbb{1}^*_{S1,k}}, \text{ and}$$
$$\frac{\partial B}{\partial \bar{g}^*_{2,i}}(\bar{v}) = -kT\frac{z_i(\bar{F})}{\sum_k z_k(\bar{F})\mathbb{1}^*_{S2,k}}.$$

3. *When EMUS is used to compute the free energy of a window, B is defined by* (2.11)*, and we have*

$$\frac{\partial B}{\partial \bar{F}_{ij}}(\bar{v}) = \frac{z_i(F)}{z_k(F)}(I-F)^{\#}_{jk}, \text{ and}$$
$$\frac{\partial B}{\partial \bar{g}^*_{i,1}}(\bar{v}) = \frac{\partial B}{\partial \bar{g}^*_{i,2}}(\bar{v}) = 0.$$

Note that for a free energy difference between windows, we can simply subtract derivatives for the corresponding windows.

Proof. We begin by reminding the reader that the output of EMUS is the vector of window normalization constants, z, which depends on the sample mean $\bar{F}$. Because all other averages and, in turn, their derivatives rely on z, we need to determine the sensitivity of each element of z to each element of $\bar{F}$ (i.e., $\partial z_k/\partial \bar{F}_{ij}$). Since $\bar{F}$ is a stochastic matrix, some care must be taken in defining this derivative. We resolve the technical difficulties in detail elsewhere; see reference [16] and Thiede *et al.* [38, Lemma 3.1]. Here, to obtain the derivative $\partial z_k/\partial \bar{F}_{ij}$, evaluated at $\bar{F}$, we perturb around $\bar{F}$:

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} z_k(\bar{F}+\varepsilon E) = \sum_{i,j=1}^{L} \frac{\partial z_k}{\partial \bar{F}_{ij}}(\bar{F})E_{ij}, \tag{2.49}$$

where E is an arbitrary matrix, ε is a scalar, and we assume that the sum $\bar{F}+\varepsilon E$ is also a stochastic matrix. The righthand side follows from the chain rule, effectively treating each

element of the matrix as a separate argument to each element z. Then, we employ a relation from Golub and Meyer [17, Theorem 3.1]:

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0} z_k(\bar{F}+\varepsilon E) = z(\bar{F})^{\mathrm{T}} E (I-\bar{F})^{\#} e_k, \tag{2.50}$$

where # denotes the group inverse, a generalized matrix inverse similar to the Moore-Penrose inverse. It is defined as satisfying $AA^{\#}A = A$, $A^{\#}AA^{\#} = A^{\#}$, $AA^{\#} = A^{\#}A$. We refer to Golub and Meyer [17] for further discussion of the group inverse and an algorithm for computing it. Finally, we equate (2.49) and (2.50) and solve for the derivative of interest:

$$\frac{\partial z_k}{\partial \bar{F}_{ij}}(\bar{F}) = z_i(\bar{F})(I-\bar{F})^{\#}_{jk}. \tag{2.51}$$

Thus the sensitivity of each element of z to each element of $\bar{F}$ can be computed from linear algebra operations.

With (2.51), we can now compute the derivatives of B. We derive the formulas for the free-energy difference explicitly; the other cases are similar. In this case,

$$B(\bar{v}) = kT \log\left(\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{2,k}\right) - kT \log\left(\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{1,k}\right).$$

By the chain rule,

$$\frac{\partial}{\partial \bar{g}^*_{1,i}} \log\left(\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{1,k}\right) = \frac{z_i(\bar{F})}{\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{1,k}},$$

and

$$\frac{\partial}{\partial \bar{F}_{ij}} \log\left(\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{1,k}\right) = \frac{\sum_{k=1}^{L} \frac{\partial z_k}{\partial \bar{F}_{ij}}(\bar{F})\bar{g}^*_{1,k}}{\sum_{k=1}^{L} z_k(\bar{F})\bar{g}^*_{1,k}}.$$

The stated result follows by substituting $g_1 = \mathbb{1}_{S1}$, $g_2 = \mathbb{1}_{S2}$, and the expression in (2.51)

for $\partial z_k / \partial \bar{F}_{ij}$. $\square$

Computational Procedure

We now provide a practical procedure that uses the derivatives above to estimate σ^2 from trajectories that sample the distributions π_i. For clarity, we assume that the system is equilibrated (i.e., X_0^i has distribution π_i, so that the process X_t^i is stationary) throughout this section.

We begin by rewriting (2.46) as

$$\sigma^2 = \sum_{i=1}^{L} \frac{\chi_i^2}{\kappa_i} \tag{2.52}$$

where

$$\begin{aligned} \chi_i^2 &= \frac{\partial B}{\partial v_i}^{\mathrm{T}} \Sigma_i \frac{\partial B}{\partial v_i} \\ &= \frac{\partial B}{\partial v_i}^{\mathrm{T}} \left(\sum_{t=-\infty}^{\infty} C_i(t) \right) \frac{\partial B}{\partial v_i}. \end{aligned}$$

Defining the sequence

$$\zeta_t^i = \left. \frac{\partial B}{\partial v_i} \right|_{\langle v \rangle} \cdot \left(v_i(X_t^i) - \langle v_i \rangle_i \right) \tag{2.53}$$

we find that

$$\chi_i^2 = \sum_{t=-\infty}^{\infty} \left\langle \zeta_t^i \zeta_0^i \right\rangle$$

which is the integrated autocovariance of ζ_t^i.

We thus propose the following algorithm, given simulation data:

1. Compute $\bar{v}$.
2. Compute $z(\bar{F})$ and $(I - \bar{F})^{\#}$ using the algorithm of Golub and Meyer [17].
3. Evaluate $\partial B / \partial v_i$ at $\bar{v}$ using the formulas in Lemma 2.7.5.

4. Compute
$$\bar{\zeta}_t^i = \left.\frac{\partial B}{\partial v_i}\right|_{\bar{v}_i} \cdot (v_i(X_t^i) - \bar{v}_i).$$

5. Compute an estimate $\bar{\chi}_i^2$ of the integrated autocovariance of $\bar{\zeta}_t^i$ using an algorithm such as ACOR [1].

6. Compute the estimate of σ^2:
$$\bar{\sigma}^2 = \sum_{i=1}^{L} \frac{\bar{\chi}_i^2}{\kappa_i}. \tag{2.54}$$

Since $\bar{F}$, $\bar{v}$, and z are all computed in the process of obtaining the EMUS averages, estimating $\bar{\sigma}^2$ only requires one additional pass over the simulation data. This additional cost is insignificant compared with that of computing the trajectories.

Both (2.46) and its approximation (2.54)ure decompose the asymptotic variance of EMUS into a sum of contributions from each window. By comparing the sizes of terms in the sum, we can determine the degrees to which different windows contribute to the error. In principle, this information can be used to guide modification of the parameters of the simulation to improve efficiency. For instance, one might adjust the proportion of samples allocated to each window, κ_i, to minimize the asymptotic variance. From (2.52), the asymptotic variance σ^2 is minimized when $\kappa_i \propto \chi_i$ (see Zhu and Hummer [14, 42]). Consequently, we can define the *relative importance* of window i as

$$\mu_i = L\frac{\chi i}{\sum_{k=1}^{L} \chi_i}, \tag{2.55}$$

where the normalization is chosen so that $\mu_i = 1$, regardless of L, if all windows have the same importance. The relative importance represents how many samples would be allocated to a window to optimally estimate a specific observable, compared to a uniform distribution over all umbrellas.

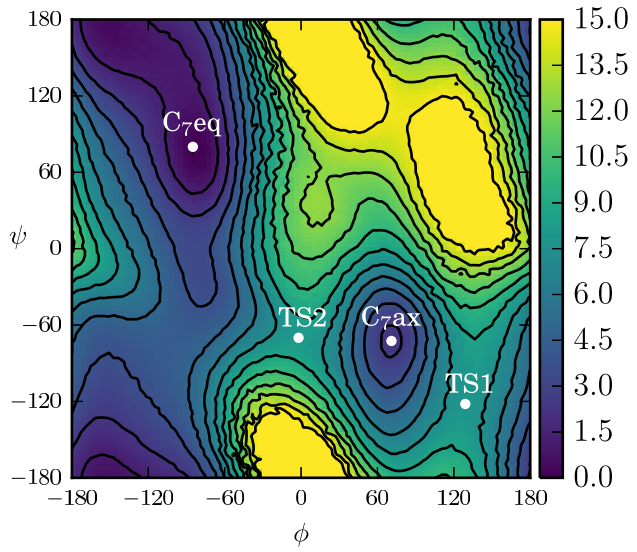


Figure 2.3: Potential of mean force obtained from US with biases on the ϕ and ψ dihedral angles. Major basins and barriers on pathways connecting them are indicated. The scale bar indicates PMF values in kcal/mol, and the contour spacing is 2 k_BT. The surface is constructed from simulation data accumulated in histograms with 100 bins in each collective variable. See text for simulation details.

To study the behavior of these estimates, we performed a two-dimensional umbrella sampling calculation with restraints on the ϕ and ψ dihedral angles of the alanine dipeptide. Parameters were the same as in the one-dimensional calculation above, with the addition of 20 bias functions in the ψ dihedral with the same force constant, creating a grid of 400 windows. Each window was equilibrated for 40 ps and sampled for a further 150 ps, with the collective variable values output every 10 fs.

In Figures 2.3 and 2.4A, we plot the two-dimensional PMF from EMUS and the importances for the free energy difference between two windows located at the C_7 equatorial and C_7 axial configurations. Comparison shows that the importances are high for windows on low free energy pathways between the two windows of interest. Two such pathways exist. In the representation in Figure 2.3, one proceeds up and to the left of the C_7 equatorial basin and then (via the periodic boundaries) enters the C_7 axial basin through transition

state 1 (TS1 in Figure 2.3). The other pathway proceeds down then right through transition state 2 (TS2 in Figure 2.3). Of these two pathways, the first has a lower free energy barrier. We observe that the EMUS importances are larger for windows located on this pathway. In contrast, windows off these pathways in regions with high free energies are given very low importances.

We expect the importances to depend on the computed average. To illustrate that this is the case numerically, we show the log importances for the free energy difference between a window in the C_7 axial basin and one located on TS1 in Figure 2.4B. Compared to Figure 2.4A, the importances are higher in the C_7 axial basin and lower in the C_7 equatorial basin, which highlights that the importances depend on the average computed and do not simply mirror the free energy. In Figures 2.4C and 2.4D, we plot the importances for estimating the window free energy (not the free energy difference) of the window on TS1 and the window in the C_7 axial basin, respectively. We note that the importances in the C_7 equatorial basin are higher in Figures 2.4C and 2.4D than in 2.4B. This suggests that when the free energy difference between the two windows is considered, there is some cancellation of the errors arising in the C_7 equatorial basin.

Comparison with Other Algorithms for Determining Error Contributions

Zhu and Hummer [14] proposed an algorithm for determining window free energies by calculating the mean restraining forces for each window and using thermodynamic integration to estimate free energy differences between adjacent windows. These are combined using least squares to calculate window free energies. Like EMUS, this algorithm allows one to construct error estimates that can be decomposed into contributions from individual windows. The authors give an expression for the error in the free energy of one window. This expression

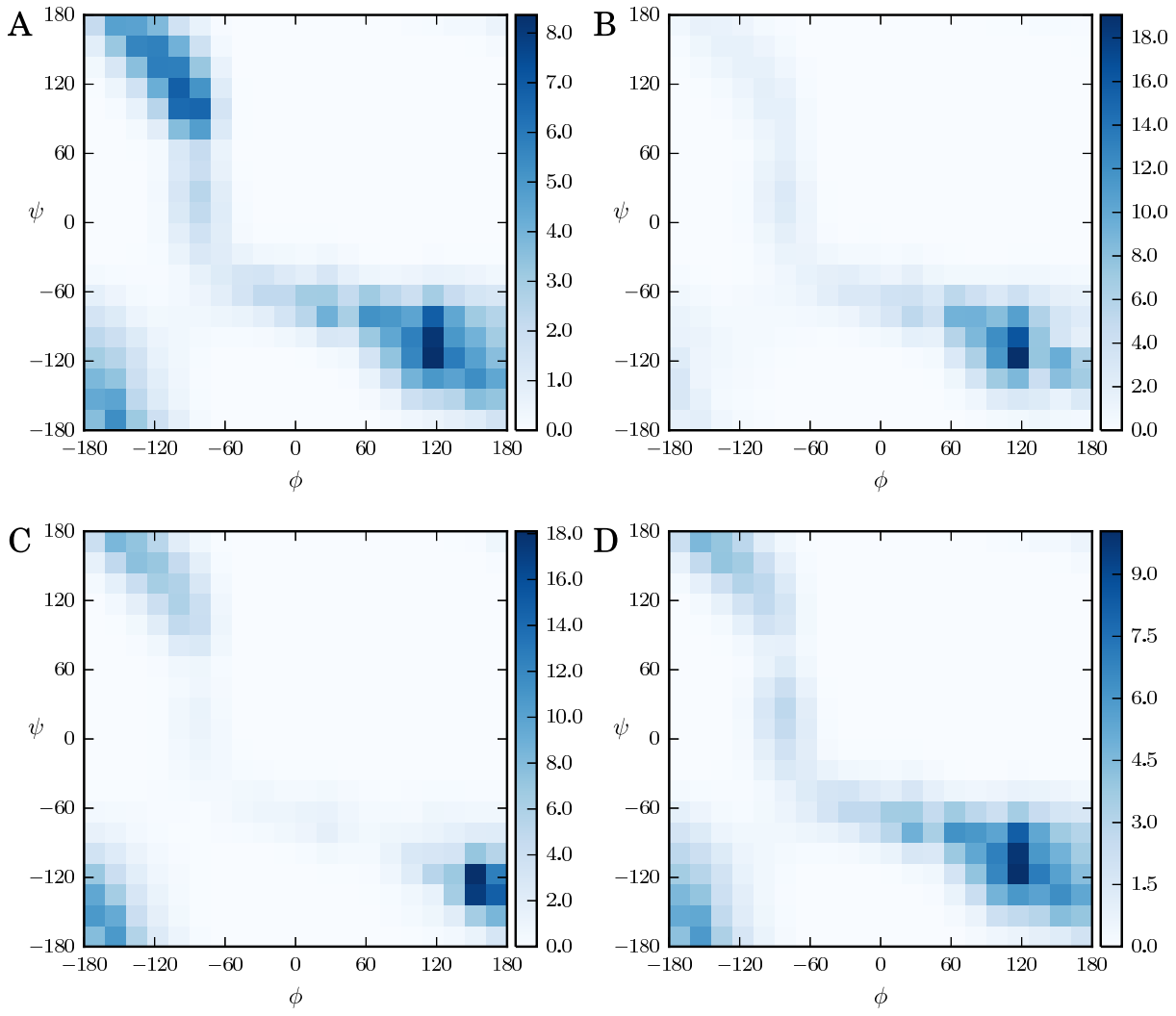


Figure 2.4: EMUS relative Importances. (A) Relative importances for the free energy difference between windows in the C_7 axial and C_7 equatorial basins. The window in the C_7 equatorial basin is centered at (–81,81), and the window in the C_7 axial basin at (63,–63). (B) Window importances for the free energy difference between windows in the C_7 axial basin and at TS1. Windows are centered at (63,–63) and (135,−117), respectively. (C) Importances for the free energy of the window at TS1. (D) Importances for the window in the C_7 axial basin.

can be easily extended to the free energy difference between two windows, giving

$$\mathrm{var}\left(\Delta G_{ji}\right) = \sum_{k}^{L} \mathrm{var}\left(\sum_{\alpha}^{D} \left(c_{jk\alpha} - c_{ik\alpha}\right) \bar{f}_{\alpha}^{k}\right), \tag{2.56}$$

where $\bar{f}_{\alpha}^{k}$ is the average force exerted by the bias function for window k in the α-th dimension. The constants $c_{ik\alpha}$ and $c_{jk\alpha}$ are defined in Supplement A of Zhu and Hummer [14]. The authors propose that these error estimates are applicable to WHAM and other umbrella sampling algorithms.

Using the formalism introduced in Section 2.7.2, we define the process

$$\zeta_{t}^{k,ZH} = \sum_{\alpha}^{D} \left(c_{jk\alpha} - c_{ik\alpha}\right) \left(f_{\alpha}^{k}\left(X_{t}^{k}\right) - \left\langle f_{\alpha,n}^{k} \right\rangle\right), \tag{2.57}$$

and χ_{i}^{ZH} as the integrated autocovariance of $\zeta_{t}^{k,ZH}$. This allows us to define importances for the Zhu and Hummer algorithm analogously to those for EMUS (see (2.55)).

We applied the ZH error analysis to the two-dimensional umbrella sampling data used in Section 2.7.2 and calculated the importances for the same free energy difference as in Figure 2.3 (Figure 2.5A). Rather than falling along the low free energy pathways, as for EMUS, the ZH importances mirror the autocorrelation times (Figure 2.5B). This indicates that windows have large ZH importances if they have large fluctuations in free energy. We thus see that different algorithms emphasize different windows in US. We can understand the behaviors of these two algorithms by considering (2.53) and (2.57). The factor $\partial B/\partial v_i$ in (2.53) depends explicitly on the normalization constant for each window (see Lemma 2.7.5). By contrast, the factor $\left(c_{jk\alpha} - c_{ik\alpha}\right)$ in (2.57) depends only on the relative positions of the windows and not on their free energies.

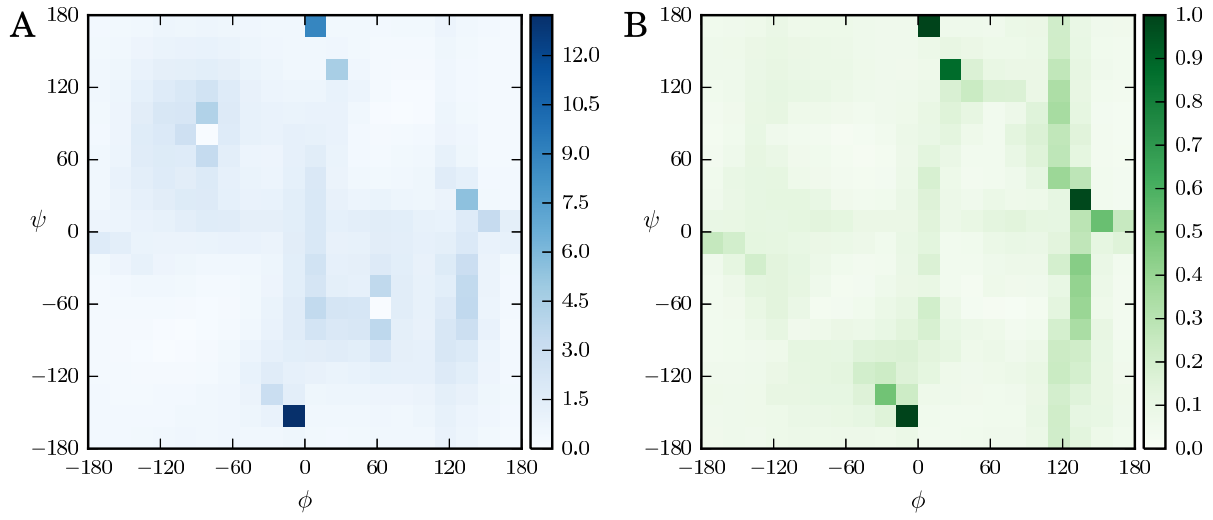


Figure 2.5: Comparison to Zhu and Hummer. (A) ZH estimates for the relative importances for the free energy difference between windows in the C_7 axial and C_7 equatorial basins. Compare with Figure 2.4A. (B) Autocorrelation times of the trajectory $\zeta_t^{k,ZH}$ in each window. The largest value observed is 3 ps, but the scale is limited to 1 ps for visual clarity.

2.8 EMUS for tails: An example from Bayesian inference

In addition to its use in molecular systems, umbrella sampling and the associated EMUS formalism can also be effectively used for problems in Bayesian Inference. We demonstrate the use of EMUS for efficiently exploring and visualizing distributions. In particular, we show how EMUS may be used to efficiently compute both marginal densities and also tail probabilities of the form $\mathbf{P}[\eta(Z) \geq \varepsilon^{-1}]$ where $\eta(Z)$ is a real valued function of a high-dimensional random variable Z. For both tails and marginals, there is a natural and easy to implement choice of strata, which we describe in Section 2.8.1.

In Section 2.8.3, we calculate two different one-dimensional marginals of the posterior distribution of the hierarchical Bayesian mixture model described in Section 2.8.2. For one marginal, the natural stratification suffices. For the other, it does not, but a preliminary computation made with the natural stratification suggests a better choice of strata. We use this example to explain how to diagnose and correct problems related to poorly chosen

strata: Our results will serve to guide the practice of stratified MCMC.

2.8.1 The natural stratification for tails and marginals

Here, we briefly explain how EMUS can be used to estimate tail probabilities and low-dimensional marginals of high-dimensional distributions. Let $\Omega \subset \mathbb{R}^d$; let π be a probability distribution on Ω; and let $\eta : \Omega \to \mathbb{R}$. Suppose that one wishes to estimate the very small tail probability $\mathbf{P}[\eta(Z) \geq \varepsilon^{-1}]$. In this case, it is natural to stratify in η only. That is, one may choose a partition of unity $\{\phi_i\}_{i=1}^L$ on $\mathbb{R}$ and define bias functions

$$\psi_i(x) = \phi_i(\eta(x)) \text{ for } i = 1, \ldots, L \tag{2.58}$$

depending only on η. For a partition of unity, one might choose a regular grid of piecewise constant functions. We refer to (2.58) as the *natural stratification*. To compute the tail probability, one uses EMUS to estimate $\pi(\mathbb{1}_{[\varepsilon^{-1},\infty)} \circ \eta)$.

Computing marginal densities is similar. Suppose now that $\eta : \Omega \to \mathbb{R}^\ell$. To estimate the marginal π_η of π in η, one chooses a partition of unity $\{\phi_i\}_{i=1}^L$ on $\mathbb{R}^\ell$, again defining bias functions by (2.58). One then uses EMUS to compute averages of *histogram bins*, which are functions of the form

$$b_{\eta_0}(\eta(x)) = \mathbb{1}_{\eta_0 + h[-1,1]^\ell}(\eta(x)). \tag{2.59}$$

We have

$$\lim_{h \to 0} \frac{1}{(2h)^\ell} \pi[b_{\eta_0}] = \pi_\eta(\eta_0),$$

so for small h the averages of the histogram bins approximate π_η.

We expect EMUS with the natural stratification will be dramatically more efficient than direct sampling *as long the biased distributions are no harder to sample than the target distribution* π. Essentially, this is because with the natural stratification very small averages like $\mathbf{P}[\eta(Z) \geq \varepsilon^{-1}]$ over the target distribution π are expressed as functions of much larger

averages over the biased distributions π_i. Unfortunately, however, for general functions η, the biased distributions of the natural stratification need not be easy to sample. In Section 2.8.3, we give one example where the natural stratification works and one where it does not. In the case where it does not, we explain how to make a better choice of strata.

2.8.2 A hierarchical Bayesian mixture model

Here, we review the hierarchical Bayesian mixture model proposed in [40], and we discuss the difficulties which complicate inference under this model. Our calculations in Section 2.8.3 demonstrate the use of EMUS in investigating and overcoming these difficulties.

In the hierarchical mixture model, the data vector

$$\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n$$

consists of independent identically distributed samples drawn from a mixture distribution of the form

$$p(y_i|\phi) = \sum_{k=1}^{K} q_k \nu(y_i; \mu_k, \lambda_k^{-1}),$$

where K is the number of mixture components, q_k is the weight of the k'th mixture component, $\nu(\cdot; \mu_k, \lambda_k^{-1})$ is the normal density with mean μ_k and variance λ_k^{-1}, and ϕ is the vector of parameters

$$\phi = (\mu_1, \ldots, \mu_K, \lambda_1, \ldots, \lambda_K, q_1, \ldots, q_{K-1}).$$

(Since $p(y_i|\phi)$ is a probability distribution, $q_1 + \cdots + q_K = 1$, and $q_1, \ldots, q_{K-1}$ determine

q_K.) The following prior distribution is imposed on ϕ:

$$\begin{aligned}
\mu_i &\sim \mathbf{N}(m, \kappa^{-1}) \\
\lambda_k &\sim \text{Gamma}(\alpha, \beta) \\
\beta &\sim \text{Gamma}(g, h) \\
(q_1, \ldots, q_{K-1}) &\sim \text{Dirichlet}_K(1, \ldots, 1).
\end{aligned}$$

As in [41, 42], we choose

$$m = M,\ \kappa = \frac{4}{R^2},\ \alpha = 2,\ g = 0.2,\ \text{and}\ h = \frac{100g}{\alpha R^2}$$

where R and M are the range and the mean of the observed data, respectively. The posterior density is

$$\begin{aligned}
p(\theta|\mathbf{y}) = \frac{\kappa^{K/2} g^h \beta^{K\alpha+g-1}}{Z_K \Gamma(\alpha)^K \Gamma(g) (2\pi)^{\frac{n+K}{2}}} &\left(\prod_{k=1}^{K} \lambda_k \right)^{\alpha-1} \\
&\times \exp\left\{ -\frac{\kappa}{2} \sum_{k=1}^{K} (\mu_k - M)^2 - \beta \left(h + \sum_{k=1}^{K} \lambda_k \right) \right\} \\
&\times \prod_{i=1}^{N} \left(\sum_{k=1}^{K} q_k \lambda_k^{\frac{1}{2}} \exp\left\{ \frac{\lambda_k}{2} (y_i - \mu_k)^2 \right\} \right),
\end{aligned}$$

where $\theta = (\phi, \beta)$ denotes the vector of all parameters to be inferred, including the hyperparameter β.

Several factors complicate inference based on this model: First, the mixture components are not identifiable; that is, the posterior distribution is invariant under permutation of the labels of the mixture components. Consequences of non-identifiability are discussed at length in [41, 42]. In our computations in Section 2.8.3, we impose the constraint

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_K$$

to ensure that the components are identifiable. Second, in Lemma 2.8.1, we show that the posterior density may be unbounded, introducing spurious modes with infinite density. Finally, even with identifiability constraints, the posterior distribution may have multiple modes of finite posterior density. For example, see the modes reported in [42]. These difficulties put the validity of point estimates such as the posterior mean in question. In Section 2.8.3, we use EMUS to efficiently visualize the posterior, assessing the effects of multimodality and unboundedness.

We suspect that the unboundedness of the posterior for this model is well known. However, we are unable to find a reference, so we now explain. It is certainly well known that the likelihood of a Gaussian mixture model is unbounded: Roughly speaking, the likelihood is infinite when any mixture component is collapsed on a single data point [43]. Nonetheless, one might expect the posterior density $p(\theta|\mathbf{y})$ to be bounded, since the prior penalizes large values of the precisions λ_i. This is not always the case when the data vector contains repeated entries:

Lemma 2.8.1. *If any datum y_i has frequency N_i greater than*

$$2g + 2(K-1)\alpha,$$

then the posterior density $p(\theta|\mathbf{y})$ is unbounded.

Proof. Take the limit of $p(\theta|\mathbf{y})$ as $\lambda_1 \to \infty$ with $\mu_1 = y_i$, $\beta = \lambda_1^{-1}$, and all other variables held fixed. □

The reader will observe that under the model, the set of data vectors with repeated entries has probability zero. However, in practice, the data consist of measurements with finite precision, and therefore repeated entries occur commonly, cf. the Hidalgo stamp data used in Section 2.8.3.

2.8.3 Numerical experiments: Choosing strata, computing tails, diagnosis of problems

In this section, we explain how to recognize and correct problems related to poor choices of strata, and we demonstrate the use of EMUS to investigate the multimodality and unboundedness of the posterior in the mixture model. We first compute two one-dimensional marginals of the high-dimensional posterior density $p(\theta|\mathbf{y})$ using the natural stratification described in (2.58). The natural stratification works in one case but not the other. In the case where the natural stratification does not work, preliminary calculations based on the natural stratification suggest a better choice of strata.

Here, we let $\mathbf{y}$ be the Hidalgo stamp data set first studied in [44], consisting of the thicknesses of 485 stamps, ranging between 60μm and 130μm. We let there be three mixture components ($K = 3$), following previous computational studies ˆcite[]Chopin2012,jasra2005. In our first, calculation, we estimated the marginal in μ_2 using the natural stratification with a grid of 201 bias functions covering the range $[7, 11]$, with the support of the leftmost and rightmost basis functions reaching to $-\infty$ and ∞, respectively. For the middle strata, define $\phi_1 : \mathbb{R} \to \mathbb{R}$ by

$$\phi_1(x) := \max\{0, 1 - |x|\}. \tag{2.60}$$

We used the bias functions

$$\psi_i(\theta) = \phi_1\left(\frac{\mu_2 - (7 + (i-1)h)}{h}\right), \text{ where } h := 0.02, \tag{2.61}$$

for $i = 2, \ldots, 200$. Now, define $\phi_2 : \mathbb{R} \to \mathbb{R}$ by

$$\phi_2(x) := \min\{\max\{0, 1 - x\}, 1\} \tag{2.62}$$

We use the strata

$$\psi_1(\theta) = \phi_2\left(\frac{\mu_2 - 7}{h}\right) \tag{2.63}$$

$$\psi_{201}(\theta) = \phi_2\left(\frac{(7 + 200h) - \mu_2}{h}\right) \tag{2.64}$$

for the first and last strata, where $h = 0.02$ as before.

We chose the total number of bias functions based on the sizes of the off-diagonal entries in the overlap matrix. For any bias functions of the form (2.61), the overlap matrix is tridiagonal. If the superdiagonal and subdiagonal entries $F_{i,i+1}$ and $F_{i,i-1}$ are sufficiently large, then the EMUS estimator is not too sensitive to statistical errors in $\bar{F}$ [16]. For our choice of bias functions, the smallest subdiagonal entry of $\bar{F}$ is approximately 0.004 and the smallest superdiagonal entry is 0.01.

We sampled the biased distributions using the affine invariant ensemble sampler with 100 walkers, as implemented in the emcee package [45]. Due to computational restrictions on memory, only every tenth sample point was saved. As a check on the sampling, the average acceptance probability over all walkers in the ensemble sampler was calculated for each biased distribution. Averaging over biased distributions gave a total average acceptance probability of 0.31. The minimum acceptance probability over all distributions was 0.12.

To initialize sampling, we computed an unbiased test trajectory. We then started by sampling a single biased distribution π_k, initializing with points drawn randomly from the unbiased trajectory. We sampled the other biased distributions in sequence, initializing with points drawn randomly from samples of adjacent biased distributions. Thus, we sampled π_k first, then π_{k-1} and π_{k+1}, then π_{k-2} and π_{k+2}, etc. We equilibrated the sampler in each π_i for 3000 Monte Carlo steps, and collected data for an additional 100000 steps.

We computed the marginal in μ_2 using a grid of 200 histogram bins, covering the region $[7, 11]$; this corresponds to taking $h = 0.01$ in (2.59). The result is the curve labeled EMUS in Figure 2.6. The marginal in μ_2 has two modes, labeled 1 and 2 in Figure 2.6. We plot

the mixture distributions corresponding to these modes in Figure 2.7. (To be precise, the distributions in Figure 2.7 correspond to means over histogram bins centered at the labeled points.)

For comparison, we also estimated the marginal in μ_2 from multiple long, unbiased trajectories. We computed 100 unbiased trajectories of the affine invariant ensemble sampler in parallel. For each trajectory, the ensembles were first equilibrated for 10000 Monte Carlo steps, and then data were collected for 100000 steps. These trajectories were combined and binned to produce the density labeled Unbiased in Figure 2.7. We estimated the relative asymptotic variance of the marginal density for the unbiased calculation using ACOR [1]. We present the results in Figure 2.6. Note that near the mode, unbiased MCMC performs slightly better than EMUS, but in the tails, EMUS performs dramatically better.

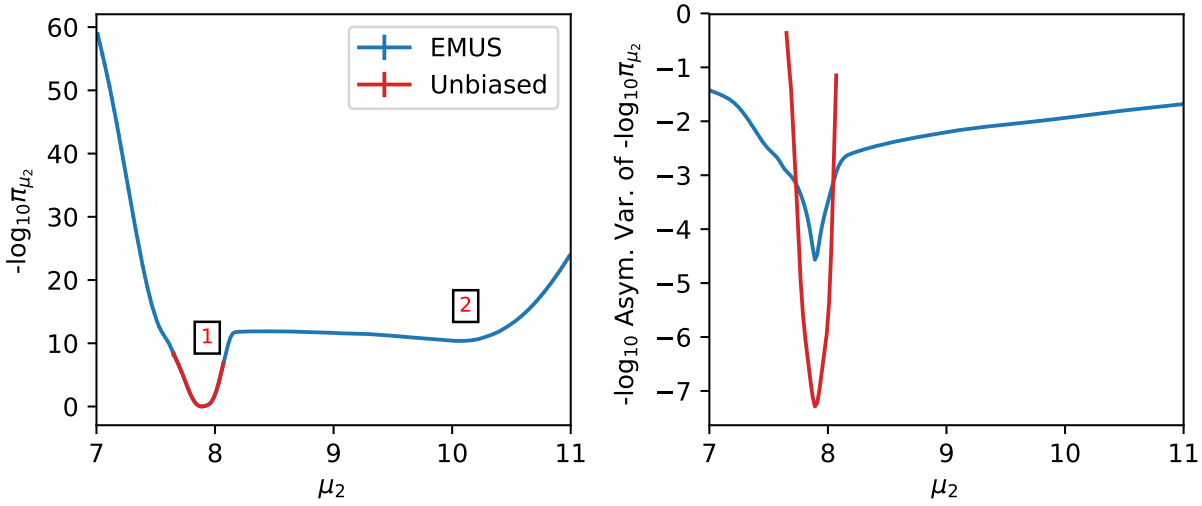


Figure 2.6: Estimates of the logarithm of the marginal density in μ_2 and the asymptotic variances of those estimates. The left subplot displays estimates of the marginal in μ_2 computed by EMUS using three different methods of initializing sampling in the biased distributions. Observe that the difference between the various methods is smaller than the line width. The right subplot displays the asymptotic variance of the marginal density in μ_2 for the unbiased and center-initialized EMUS calculations. We note that while the unbiased calculation has greater accuracy near the mode, the EMUS calculation has greater accuracy in the tails.

After computing the marginal in μ_2, we tried computing the marginal in $\log_{10} \lambda_1$ using

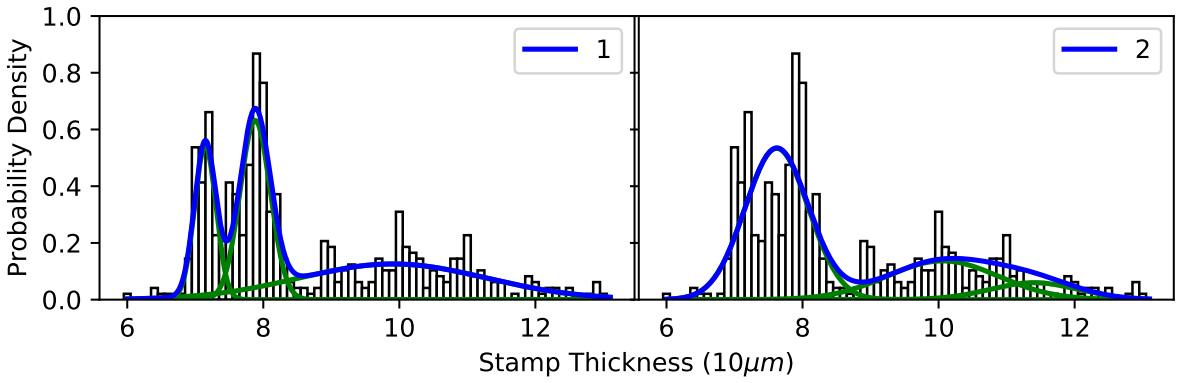
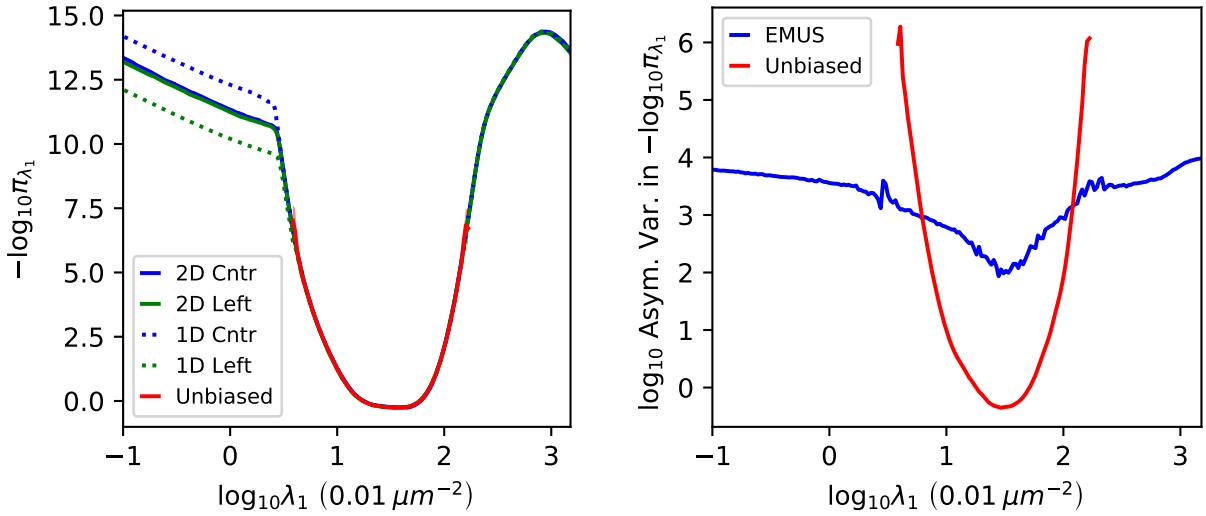


Figure 2.7: Gaussian mixtures corresponding to modes of the marginal in μ_2. Mixtures 1 and 2 correspond to the labeled points in Figure 2.6. To be precise, the blue curve in each plot is the mixture distribution corresponding to the mean of a histogram bin centered at the point labeled in Figure 2.6. The green curves are the individual mixture components. The black histogram is the Hidalgo stamp data.

the natural stratification described before. We used the same initialization scheme as for the marginal in μ_2, beginning with a single biased distribution initialized from an unbiased test trajectory. We call this the *center* sample. The result of this calculation was the density labeled "1D Center" in Figure 2.8(a). When we tried to compute the asymptotic variance of this density estimate, we noticed very slow convergence of the sampler for some biased distributions. To investigate, we performed another EMUS calculation using a similar initialization procedure, but starting from π_1, the biased distribution at the extreme left, covering the lowest values of λ_1. We call this the *left* sample. The result of this second calculation was the density labeled "1D Left" in Figure 2.8(a). For both the center and left samples, the strata were equilibrated for 3000 steps and sampled for another 200000. We observe that the two densities differ significantly in the region $-1 \leq \log_{10} \lambda_1 \leq 0.5$. They should be the same up to sampling errors; for example, we observe that different initializations have no effect on the calculation of the marginal in μ_2, cf. Figure 2.6.

Figure 2.9 explains the problem and suggests a solution: In the region $0.2 \leq \log_{10} \lambda_1 \leq 0.7$, the center and left samples cover entirely different ranges of $\log_{10} \lambda_2$. This suggests that

the biased distributions corresponding to the range $0.2 \leq \log_{10} \lambda_1 \leq 0.7$ are multimodal, with barriers in λ_2 impeding sampling.



(a) (b)

Figure 2.8: Estimates of the logarithm of the marginal density in $\log_{10} \lambda_1$ and the asymptotic variances of those estimates. Figure 2.8(a) displays the estimates of the marginal in $\log_{10} \lambda_1$ computed by various methods. The error bars are twice the estimated asymptotic standard deviation in each histogram bin. For both the unbiased calculation asymptotic variances were estimated using ACOR [1]. No error bars are given for the two one-dimensional calculations, as the barrier depicted in Figure 2.10 makes accurate estimation of the asymptotic variance impossible. A clear error is visible in the two one-dimensional umbrella sampling calculations, due to initialization along either side of the barrier in Figure 2.10. Figure 2.8(b) displays the asymptotic variance of the marginal density in $\log_{10} \lambda_1$ for the unbiased and the two-dimensional EMUS calculations. We note that while the unbiased calculation achieves greater accuracy near the mode, the EMUS calculation achieves greater accuracy in the tails.

To confirm the hypothesis that barriers in λ_2 were responsible for the poor convergence observed in the center and left samples, we performed a third calculation, stratifying in both $\log_{10} \lambda_1$ and $\log_{10} \lambda_2$. We used a 50×50 grid of bilinear bias functions, with maxima equally spaced between -1 and 3.2. To be precise, for $i, j = 1, \ldots, 50$, we defined the bias functions

$$\psi_{ij}(\theta) = \phi\left(\frac{-1 + h(i-1) - \log_{10} \lambda_1}{h}\right) \phi\left(\frac{-1 + h(j-1) - \log_{10} \lambda_2}{h}\right),$$

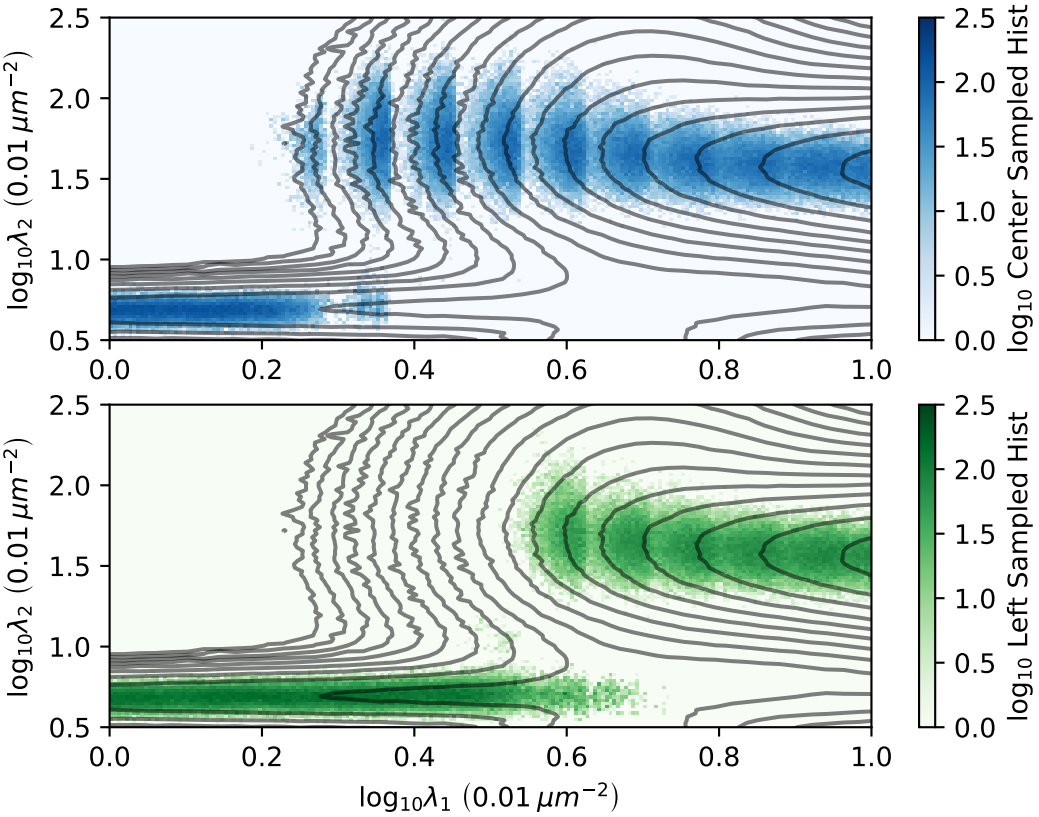


Figure 2.9: To generate Figure 2.9, we binned the samples for the one-dimensional left and center EMUS calculations, and we plotted the difference in the histograms. The contour lines are contours of the log marginal density, as in Figure 2.11(a). Figure 2.9 shows that while the two calculations largely sample the same regions, near $\log_{10}\lambda_1 = 0.45$ they become trapped on opposite sides of a barrier. This leads to poor sampling, causing a slowly decaying error in the estimates of the marginal density, cf. Figure 2.8(a)

where

$$h = \frac{3.2 - (-1)}{49}.$$

Let η_{ij} denote the biased distribution corresponding to ψ_{ij}.

We performed the two-dimensional EMUS calculation twice, initializing from the center and left samples drawn from the natural stratification in $\log_{10}\lambda_1$. For each $i = 1, \ldots, L$, to sample the row $\{\eta_{ij} : j = 1, \ldots, 50\}$ of biased distributions, we began by initializing sampling of a single biased distribution η_{ik} with points from the either the center or left sample of

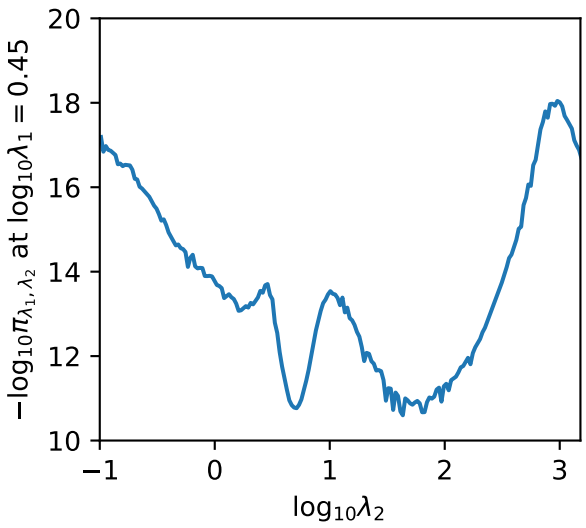
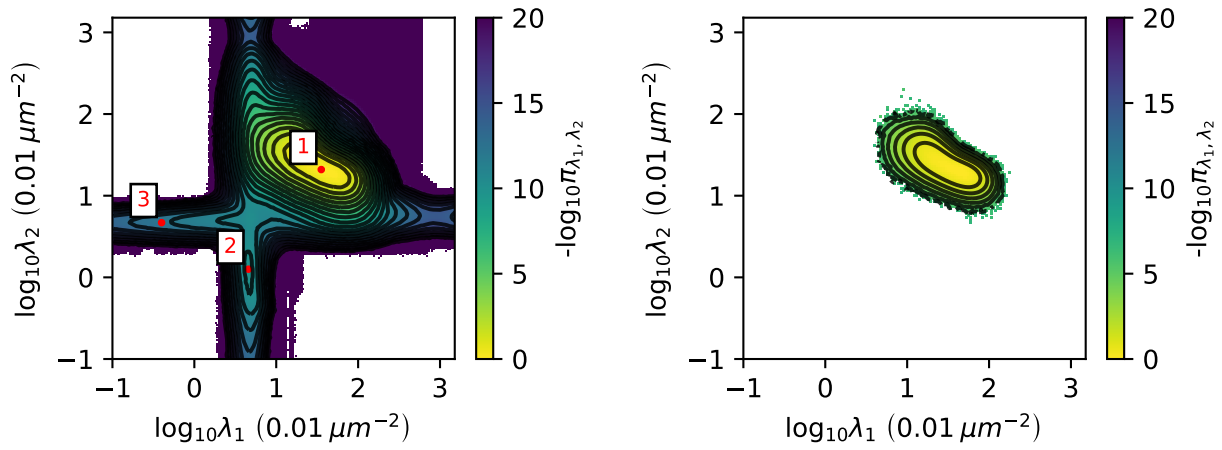


Figure 2.10: Here we give an estimate of the conditional distribution of $\log_{10}\lambda_2$ with $\log_{10}\lambda_1 = 0.45$ calculated from the two-dimensional marginal seen in Figure 2.11(a). The conditional distribution is multimodal. The mode on the left corresponds to mixtures with the data from thicknesses of 60 to 85 μm covered by a single Gaussian similar to mode 2 in Figure 2.12. The mode on the right corresponds to mixtures with these data covered by two Gaussians similar to mode 1 in Figure 2.12.

π_i. We then sampled the other distributions η_{ij} for $j \neq k$ in sequence, again initializing with points from samples of adjacent distributions, either $\eta_{i,j+1}$ or $\eta_{i,j-1}$ in this case. If no samples were found inside the support of a biased distribution, that distribution was ignored. For each biased distribution, sampling was burned in for 4500 steps, and samples were collected for an additional 2500 steps. Ultimately, 1397 of the 2500 biased distributions were sampled; the unsampled distributions correspond to the white space in Figure 2.11(a).

We computed the marginal in $\log_{10}\lambda_1$ and $\log_{10}\lambda_2$ using a 200×200 grid of histogram bins, covering the region $-1 \leq \log_{10}\lambda_1 \leq 3.2$ and $-1 \leq \log_{10}\lambda_2 \leq 3.2$; this corresponds to taking $h = (3.2 - (-1))/200$ in (2.59); the result from the center calculation appears in Figure 2.11(a). The two-dimensional marginals were essentially the same for the center and left initializations; see Figure 2.13. We also estimated the one-dimensional marginal in $\log_{10}\lambda_1$ using the two-dimensional stratification; see the results labeled "2D Center" and "2D Left " in Figure 2.8(a). Finally, we estimated the relative asymptotic variance of the



(a) (b)

Figure 2.11: Logarithm of marginal density in $\log_{10}\lambda_1$ and $\log_{10}\lambda_2$ as estimated by EMUS and unbiased MCMC. Contour lines in both figures are every unit change in the estimated $\log_{10}$ marginal density. Figure 2.11(a) is the EMUS estimate. The numbers 1, 2, and 3 on this figure correspond to the mixture densities in Figure 2.12. Note that at values of $\log_{10}\lambda$ near 3.0 we begin to see the modes corresponding to singularities of the posterior. Figure 2.11(b) is the marginal density estimated from a long unbiased trajectory of the ensemble sampler. Note that the entire trajectory lies in a small neighborhood of the mode labeled 1 in Figure 2.11(a).

marginal in $\log_{10}\lambda_1$ computed by two-dimensional stratification. Again, we observe that EMUS performs much better than unbiased sampling in the tails, cf. Figure 2.8(b).

The marginal in $\log_{10}\lambda_1$ and $\log_{10}\lambda_2$ confirms that barriers in λ_2 caused the problems observed in calculating the marginal in $\log_{10}\lambda_1$ using the natural stratification. In fact, we see that computing the marginal in either λ_1 or λ_2 requires stratifying both variables, as stratifying only one leads to barriers that impede sampling in the other. In particular, there are barriers in λ_2 along the line $\log_{10}\lambda_1 = 0.45$ and a barrier in λ_1 along $\log_{10}\lambda_2 = 0.6$: In Figure 2.10, we plot an estimate of the conditional distribution of $\log_{10}\lambda_2$ with $\log_{10}\lambda_1 = 0.45$ fixed. This distribution is multimodal with a region of very low probability separating the modes, which explains the poor sampling depicted in Figure 2.9.

To conclude, we have confirmed that EMUS can be extremely efficient for computing tails. However, one must exercise care in the choice of strata. The natural stratification often

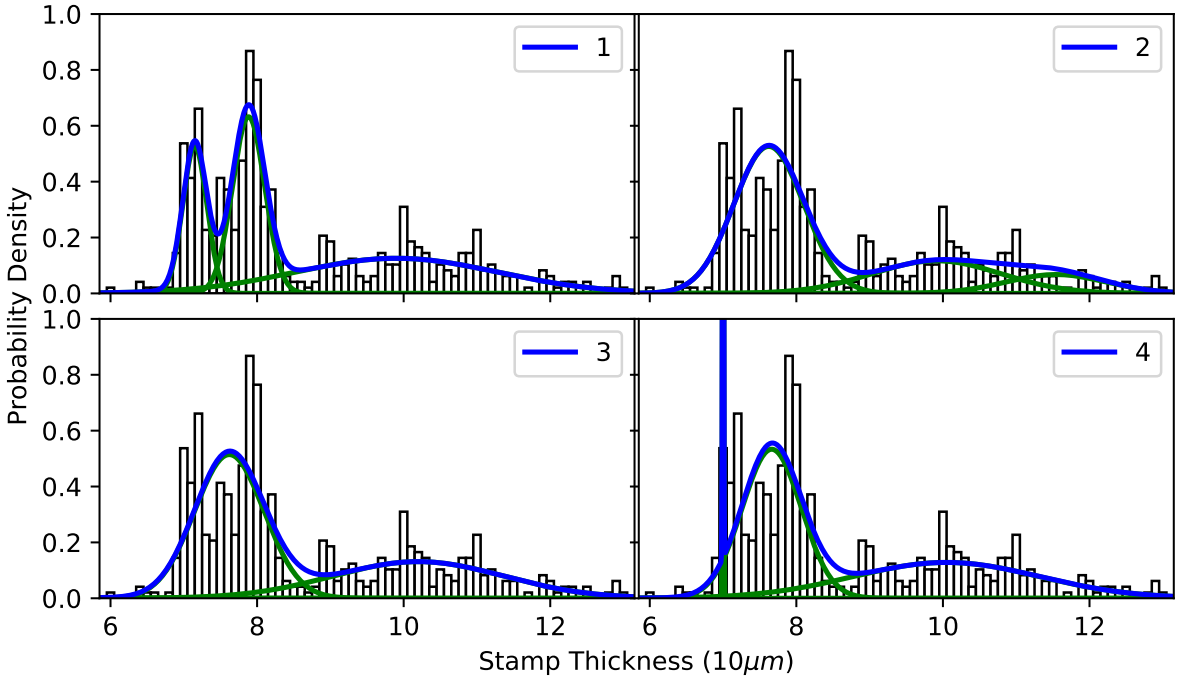


Figure 2.12: Gaussian mixtures corresponding to means of histogram bins. Mixtures one through three correspond to the labeled points on Figure 2.11(a), mixture four corresponds to a distribution near a singularity of the posterior, with $\log_{10}\lambda_1 = 4.34$ and $\log_{10}\lambda_2 = 0.79$. To be precise, the blue curve in each plot is the mixture distribution corresponding to the mean of a histogram bin centered at the point labeled in Figure 2.11(a). The green curves are the individual mixture components. The black histogram is the Hidalgo stamp data.

suffices, but in some cases, like computing the marginal in $\log_{10}\lambda_1$, the biased distributions of the natural stratification may be very difficult to sample. We propose the use of different initializations, like the center and left samples, as a method of identifying problems related to poorly chosen strata. Careful inspection of simulations performed with these different initializations can identify problems and suggest better strata.

2.9 Conclusions

The success of an umbrella sampling simulation depends on the choice of windows (i.e., how the system is biased) and the estimator used to determine the normalization constants of the windows from trajectory data. Here, we show that the normalization constants can be

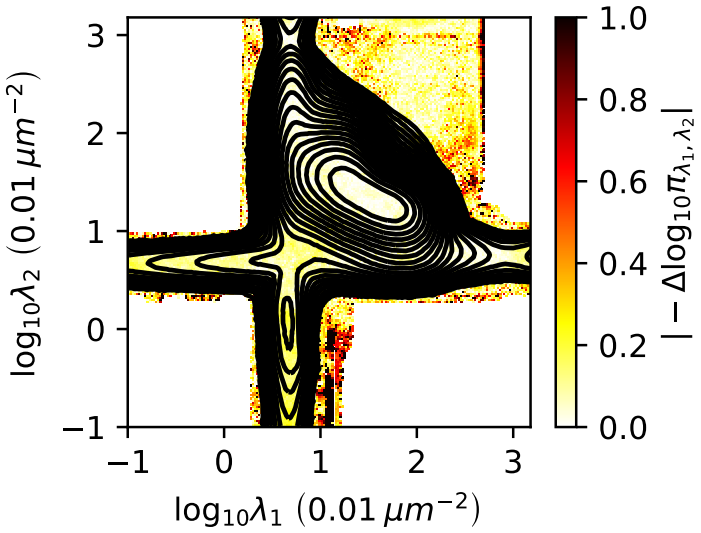


Figure 2.13: The difference between the free energy surfaces of the two-dimensional umbrella sampling runs. The center calculation was initialized from the center one-dimensional calculation, and the left calculation from the left one-dimensional calculation. In general the difference is small, roughly a tenth of an order of magnitude in the log marginal.

obtained from an eigenvector of a stochastic matrix. This eigenvector method for umbrella sampling (EMUS) can be viewed as the first step in an implementation of the MBAR estimator. In our experience, this first step is nearly converged, and machine precision is reached in only a few iterations. Moreover, each iteration yields a consistent estimate. Most importantly, error analysis is considerably easier for EMUS than MBAR because the elements of the stochastic matrix do not depend on the normalization constants.

Within this framework, we revisited a common scaling argument for justifying umbrella sampling and showed that once the number of windows becomes sufficiently large, the scheme does not benefit from the addition of more windows (i.e., the variance is not further reduced for a fixed computational effort). We show that an alternative scaling regime in which temperature decreases (or, equivalently, free-energy barrier heights increase) as the number of windows increases best demonstrates the potential benefits of the umbrella sampling strategy; in that regime the efficiency improvement over direct simulation is exponential in

the (inverse) temperature.

Our main theoretical result is a central limit theorem for the statistical averages obtained from EMUS. This result relies on the delta method, which we use to characterize the propagation of the asymptotic error through the solution of a stochastic matrix eigenproblem. The central limit theorem provides an expression for the asymptotic variance of the averages of interest. It is a sum of contributions from individual windows, and we use it to develop a prescription for estimating the relative importances of windows for averages from the trajectory data. For free energy differences of states of the alanine dipeptide, we find numerically that the importances are largest for low-free energy pathways that connect the specific states of interest. These results suggest that the importances could serve as the basis for adaptive schemes that focus computational effort on the windows of most importance. Even more interesting would be to adjust the bias functions as the simulation progresses. How best to do this remains an open area of investigation.

2.10 Supplement: Consistency of Iterative EMUS

Here, we prove that for fixed finite m, z^m is a consistent estimator of the vector of normalization constants z. With the initial guess $z^0 = n$, the result, z^1, of the first iteration is the EMUS estimator. We now show that z^2 is also consistent in the sense that if the trajectory averages defining $\bar{F}(w)$ converge then z^2 converges to z. Because the various sequences in question are sequences of random variables one must specify what is meant by convergence. The argument below applies when convergence refers either to convergence in probability or convergence with probability one (almost sure convergence) as long as the notion of convergence is consistent throughout. Consistency of z^m follows by induction on m using a similar argument.

For any positive vector w, we define

$$u_k = \prod_{j \neq k} \frac{w_j}{n_j},$$

and we write

$$\bar{F}_{ij}(w) = \frac{1}{N_i} \sum_{t=0}^{N_i - 1} h_{ij}(u, x),$$

where

$$h_{ij}(u, x) = \frac{u_i \psi_j(x)}{\sum_k u_k \psi_k(x)}.$$

We then observe that

$$\partial_{u_k} h_{ij}(u, x) = \begin{cases} \frac{1}{u_i} h_{ij}(u, x)[1 - h_{ii}(u, x)], & \text{if } k = i \\ -\frac{1}{u_i} h_{ij}(u, x) h_{ik}(u, x), & \text{if } k \neq i. \end{cases}.$$

Because $h_{ij}(u, x) \leq u_i / u_j$,

$$|\partial_{u_k} h_{ij}(u, x)| \leq \frac{1}{u_j} \max \left\{ 1, \frac{u_i}{u_k} \right\}.$$

Therefore,

$$|h_{ij}(\tilde{u}, x) - h_{ij}(u, x)| \leq \gamma(u, \tilde{u}), \tag{2.65}$$

for a continuous function γ defined for positive vectors u and $\tilde{u}$ and such that $\gamma(u, u) = 0$ for any u. The function $\gamma(u, \tilde{u})$ must explode when the entries of u or w approach 0. Now define

$$u_k = \prod_{j \neq k} \frac{z_j^1}{n_j} \text{ and } \tilde{u}_k = \prod_{j \neq k} \frac{z_j}{n_j},$$

where z is the exact vector of normalization constants. By (2.65), we have

$$|\bar{F}_{ij}(z^1) - \bar{F}_{ij}(z)| \leq \gamma(u, \tilde{u}). \tag{2.66}$$

As the number of samples N increases, $\bar{F}(z)$ converges to $F(z)$. Moreover, since z^1 is the EMUS estimate of z, z^1 converges to z. Therefore, u converges to $\tilde{u}$, and (2.66) implies that $\bar{F}_{ij}(z^1)$ converges to $F_{ij}(z)$. Finally, since the function mapping an irreducible, stochastic matrix to its invariant vector is continuous, it follows that z^2 converges to the invariant vector of $F(z)$, which is z. This verifies the consistency of z^2.

CHAPTER 3

DYNAMICAL GALERKIN APPROXIMATION

Understanding chemical mechanisms requires estimating dynamical statistics such as expected hitting times, reaction rates, and committors. Here, we present a general framework for calculating these dynamical quantities by approximating boundary value problems using dynamical operators with a Galerkin expansion. A specific choice of basis set in the expansion corresponds to estimation of dynamical quantities using a Markov state model. More generally, the boundary conditions impose restrictions on the choice of basis sets. We demonstrate how an alternative basis can be constructed using ideas from diffusion maps. In our numerical experiments, this basis gives results of comparable or better accuracy to Markov state models. Additionally, we show that delay embedding can reduce the information lost when projecting the system's dynamics for model construction; this improves estimates of dynamical statistics considerably over the standard practice of increasing the lag time.

3.1 Introduction

Molecular dynamics simulations allow chemical mechanisms to be studied with atomistic detail. By averaging over trajectories, one can estimate dynamical statistics such as mean first-passage times or committors. These quantities are integral to chemical rate theories [46, 47, 48]. However, events of interest often occur on timescales several orders of magnitude longer than the timescales of microscopic fluctuations. In such cases, collecting chemical-kinetic statistics by integrating the system's equations of motion and directly computing averages (sample means) requires prohibitively large amounts of computational resources.

The traditional way to address this separation in timescales was through theories of activated processes [47, 49]. By assuming that the kinetics are dominated by passage through a single transition state, researchers were able to obtain approximate analytical forms for reaction rates and related quantities. These expressions can be connected with microscopic

simulations by evaluating contributing statistics such as the potential of mean force and the diffusion tensor [50, 51, 52]. However, many processes involve multiple reaction pathways, such as the folding of larger proteins [53, 54]. In these cases it may not be possible in practice, or even in principle, to represent the system in a way that the assumptions underlying theories of activated processes are reasonable.

More recently, transition path sampling algorithms, which focus sampling on the pathways connecting metastable states, have been used to estimate rates [55, 56]. Given such trajectories, other dynamical statistics, such as commitors, can be learned [57, 58]. Short trajectories reaching the metastable states can be harvested efficiently, but sampling long trajectories, especially those including multiple intermediates, becomes difficult [59, 60]. Another approach is to use splitting schemes, which aim to efficiently direct sampling by intelligently splitting and reweighting short trajectory segments [61, 62, 63, 64, 65, 66, 67, 68, 69, 3]. Some of these methods can yield results that are exact up to statistical precision, with minimal assumptions about the dynamics [65, 66, 67, 68, 69, 3]. However, the efficiency of these schemes is generally dependent on a reasonable choice of low-dimensional *collective variable* (CV) space: a projection of the system's phase space. Not only can this choice be nonobvious [57], it can be statistic specific. Moreover, starting and stopping the molecular dynamics many times based on the values of the CVs may be impractical depending on the implementation of the molecular dynamics engine and the overhead associated with computational communication.

A third approach is the construction of Markov state models (MSMs) [70, 71, 72]. Here, the dynamics of the system are modeled as a discrete-state Markov chain with state-to-state transition probabilities estimated from previously sampled data. Projecting the dynamics onto a finite-dimensional model introduces a systematic bias, although this bias goes to zero in the correct limit of infinitely many states [73]. While MSMs were initially developed as a technique for approximating the slowest eigenmodes of a system's dynamics [70], MSMs can also be used to calculate dynamical statistics for the study of kinetics [74, 75, 76]. Since MSM

construction only requires time pairs separated by a single lag time, one has more freedom in how one generates the molecular dynamics data. In particular, if the lag time is sufficiently short, MSMs can be used to estimate rates even in the absence of full reactive trajectories. Constructing an efficient MSM requires projection onto CVs, and the systematic error in the resulting estimates can depend strongly on how they are defined. However, the CV space can generally be higher dimensional since it is only used to define Markov states.

It has been shown that calculating the system's eigenmodes with MSMs can be generalized to a basis expansion of the eigenmodes using an arbitrary basis set [77, 73, 78]. In this paper, we show that a similar generalization is possible for other dynamical statistics. Rather than solving eigenproblems, these quantities solve linear boundary value problems. This raises additional challenges: not only do the solutions obey specific boundary conditions, the resulting approximations are sensitive to the choice of lag time. We provide numerical schemes to address these difficulties.

We organize our work as follows. In Section 3.2 we give background on the transition operator and review both MSMs and more general schemes for data-driven analysis of the spectrum of dynamical operators. We then continue our review with the connection between operator equations and chemical kinetics in Section 3.3. In Section 3.4 we present our formalism. We discuss the choice of basis set in Section 3.5 and introduce a new algorithm for constructing basis sets that obey the boundary conditions our formalism requires. In Section 3.6 we show that delay embedding can recover information lost in projecting the system's dynamics onto a few degrees of freedom, negating the need for increasing the scheme's lag time to enforce Markovianity. We then demonstrate our algorithm on a collection of long trajectories of the Fip35 WW domain dataset in Section 3.7, and conclude in Section 3.8. This work was previously published in [79].

3.2 Background

Many key quantities in chemical kinetics can be expressed through solutions to linear operator equations. Key to this formalism is the *transition operator*. We begin by assuming that the system's dynamics are given by a Markov process $\xi^{(t)}$ that is time-homogeneous, i.e. that the dynamics are time-independent. We do not put any restrictions on the nature of the system's state space. For example, if ξ is a diffusion process, the state space could be the space of real coordinates, $\mathbb{R}^n$. Similarly, for a finite-state Markov chain, it would be a finite set of configurations. We also do not assume that the dynamics are reversible or that the system is in a stationary state unless specifically noted.

The transition operator at a lag time of s is defined as

$$\mathcal{K}_s f(x) = \mathbf{E}\left[f\left(\xi^{(s)}\right)|\xi^{(0)} = x\right], \tag{3.1}$$

where f is a function on the state space, and $\mathbf{E}$ denotes expectation. Note that due to time-homogeneity, we could just as easily have defined the transition operator with the time pair $\left(\xi^{(t)}, \xi^{(t+s)}\right)$ in place of $\left(\xi^{(0)}, \xi^{(s)}\right)$. Depending on the context in question, $\mathcal{K}_s$ may also be referred to as the Markov or Koopman operator [80, 4]. We use the term transition operator as it is well established in the mathematical literature and stresses the notion that $\mathcal{K}_s$ is the generalization of the transition matrix for finite-state Markov processes. For instance, the requirement that the rows of a transition matrix sum to one generalizes to

$$\mathcal{K}_s 1 = \mathbf{E}\left[1|\xi^{(0)} = x\right] = 1. \tag{3.2}$$

Studying the transition operator provides, in principle, a route to analyzing the system's dynamics. Unfortunately $\mathcal{K}_s$ is often either unknown or too complicated to be studied directly. This has motivated research into data-driven approaches that instead treat $\mathcal{K}_s$ indirectly by analyzing sampled trajectories.

3.2.1 Markov State Modeling

One approach to studying chemical dynamics through the transition operator is the construction of Markov state models [70, 71, 72]. In this technique one constructs a Markov chain on a finite state space to model the true dynamics of the system. The transition matrix of this Markov chain is then taken as a model for the true transition operator.

To construct an MSM from trajectory data, we partition the system's state space into M nonoverlapping sets. We refer to these sets as Markov states and denote them as S_i. Now, let μ be an arbitrary probability measure. If the system is initially distributed according to μ, the probability of transitioning from a set S_i to S_j after a time s is given by

$$P_{ij} = \frac{\int \mathbb{1}_{S_i}(x)\mathcal{K}_s\mathbb{1}_{S_j}(x)\mu(dx)}{\int \mathbb{1}_{S_i}(x)\mu(dx)}, \tag{3.3}$$

where $\mathbb{1}_{S_i}$ is the indicator function

$$\mathbb{1}_{S_i}(x) = \begin{cases} 1 \text{ for } x \text{ in } S_i \\ 0 \text{ otherwise.} \end{cases} \tag{3.4}$$

Here $\int f(x)\mu(dx)$ is the expectation with respect to the probability measure μ [81]. When μ has a probability density function this integral is the same as the integral against the density, and in a finite state space it would be a weighted average over states. This formalism lets us treat both continuous and discrete state spaces with one notation.

Because the sets S_i partition the state space, a simple calculation shows that the elements in each row of P_{ij} sum to one. P_{ij} therefore defines a transition matrix for a finite-state Markov process where state i corresponds to the set S_i. The dynamics of this process are a model for the true dynamics, and P_{ij} is a model for the transition operator.

To build this model, we construct an estimate of P_{ij} from sampled data. A simple approach is to collect a dataset consisting of N time pairs, (X_n, Y_n). Here the initial point

X_n is drawn from μ, and Y_n is collected by starting at X_n and propagating the dynamics for time s. Note that since the choice of μ in (3.3) is relatively arbitrary, it can be defined implicitly through the sampling procedure. For instance, one can construct a dataset by extracting all pairs of points separated by the lag time s from a collection of trajectories; since we have assumed the dynamics are time-homogeneous, the actual physical time at which X_n was collected does not matter. We then define μ to be the measure from which our initial points $X_n^{(0)}$ were sampled. With this dataset, P_{ij} is now approximated as

$$\bar{P}_{ij} = \frac{\sum_{n=1}^{N} \mathbb{1}_{S_j}(Y_n)\, \mathbb{1}_{S_i}(X_n)}{\sum_{n=1}^{N} \mathbb{1}_{S_i}(X_n)} \tag{3.5}$$

Like P_{ij}, (3.5) defines a valid transition matrix. This is not the only approach for constructing estimates of P_{ij}. One commonly used approach modifies this procedure to ensure that $\bar{P}_{ij}$ gives reversible dynamics. In this approach, one adds a self-consistent iteration that seeks to find the reversible transition matrix with the maximum likelihood given the data [82, 83].

The MSM approach has many attractive features. Since P_{ij} defines a valid transition matrix, the MSM defines a Markov chain that can be used as a general model for the dynamics. This model can then be simplified by merging the Markov sets to improve interpretability [84, 72, 85]. MSMs can also be used to estimate spectral information associated with the transition operator such as its eigenvalues and eigenvectors, as we discuss in further detail in Section IIB [70, 73, 71, 83]. Finally, MSMs can be used to calculate a wide class of dynamical quantities, including committors, reaction rates, and expected hitting times [74, 75, 76]. Importantly, as constructing MSMs only requires datapoints separated by a short lag time, these long-time dynamical quantities can be evaluated using a collection of short trajectories [86]. In this paper, we focus exclusively on the latter application and consider MSMs as a technique for calculating the dynamical quantities required in rate theories.

The accuracy with which P_{ij} approximates $\mathcal{K}_s$ depends strongly on the choice of the

sets S_i, and choosing good sets is a nontrivial problem in high-dimensional state spaces [87, 88, 89, 90, 91]. To address this issue, states are generally constructed by projecting the system's state space onto a CV space. Sets are then defined by either gridding the CV space or clustering sampled configurations based on the values of their CVs. Unfortunately, when gridding, the number of states grows exponentially with the dimension of the CV space. This is not necessarily the case for partitioning schemes based on data clustering and recent work in this direction appears promising [83, 92, 93, 94, 95, 96]. In particular, recent approaches have used variational principles associated with the spectrum of $\mathcal{K}_s$ to give a quantitative notion of approximation quality across clustering procedures. [78, 97, 98, 99, 100, 101] However, effectively clustering high-dimensional data is a nontrivial problem [102, 103], and constructing an MSM that accurately reflects the dynamics may still require knowledge of a good, relatively low-dimensional CV space [87, 88, 104].

3.2.2 Data-driven Solutions to Eigenfunctions of Dynamical Operators

A related approach to characterizing chemical systems is to estimate the eigenfunctions and eigenvalues of operators associated with the system's dynamics from sampled data [4]. These separate the dynamics by timescale: eigenfunctions with larger eigenvalues correlate with the system's slower degrees of freedom. These eigenfunctions and eigenvalues can often be approximated from trajectory data, even when the transition operator is unknown. Multiple schemes that attempt this have been proposed, often independently, in different fields [105, 106, 107, 70, 108, 77, 78, 109, 110, 111]. We will refer to the family of these techniques using the umbrella term *Dynamical Operator Eigenfunction Analysis (DOEA)* for brevity and convenience. Below, we summarize a simple DOEA scheme for the transition operator for the reader's convenience, largely following work in Reference [107]. Other schemes exist and we refer the reader to Reference [4] for further reading.

Here, we consider the solution to the eigenproblem

$$\mathcal{K}_s \psi_l(x) = \lambda_l \psi_l(x). \tag{3.6}$$

We approximate ψ_l as a sum of basis functions ϕ_j with unknown coefficients a_j,

$$\psi_l(x) = \sum_{j=1}^{M} a_j \phi_j(x). \tag{3.7}$$

This is an example of Galerkin approximation of (3.6) [70], a formalism we cover more closely in Section 3.4. We now assume our data takes the form discussed in Section 3.2.1. Substituting the basis expansion into (3.6), multiplying by $\phi_i(x)$, and taking the expectation against μ, we obtain the matrix equation

$$\sum_{j=1}^{M} K_{ij} a_j = \lambda_l \sum_{j=1}^{M} S_{ij} a_j \tag{3.8}$$

where K_{ij} and S_{ij} are defined as

$$K_{ij} = \int \phi_i(x) \mathcal{K}_s \phi_j(x) \mu(dx) \tag{3.9}$$

$$S_{ij} = \int \phi_i(x) \phi_j(x) \mu(dx) \tag{3.10}$$

respectively. The matrix elements can be approximated as

$$\bar{K}_{ij} = \frac{1}{N} \sum_{n=1}^{N} \phi_i(X_n) \phi_j(Y_n) \tag{3.11}$$

$$\bar{S}_{ij} = \frac{1}{N} \sum_{n=1}^{N} \phi_i(X_n) \phi_j(X_n). \tag{3.12}$$

We substitute these approximations into (3.8) and solve for estimates of a_i and λ_l. Equation (3.7) can then be used to give an approximation for ψ_l.

DOEA schemes are closely linked to MSMs. Using the indicator functions from Section 3.2.1 is mathematically equivalent to solving for the eigenfunctions of P_{ij}. Indeed, one of the first uses for MSMs was for approximating the eigenfunctions and eigenvalues of the transition operator [70, 73].

The use of more general basis sets in DOEA allows information to be more easily extracted from high-dimensional CV spaces and gives added flexibility in algorithm design [105, 87, 88, 112, 113]. Time-lagged independent component analysis (TICA) corresponds to a basis of linear functions and is commonly applied as a preprocessing step to generate CVs for MSM construction [105, 87, 88]. Variational principles can be exploited to obtain the eigenfunctions of $\mathcal{K}_s$ for reversible dynamics (variational approach of conformation dynamics, VAC) [78] and, more generally, for the singular value decomposition of $\mathcal{K}_s$ (the variational approach for Markov processes, VAMP) [99, 98]. These principles can allow the creation of cost functions that can be used to assess how well a basis recapitulates the spectral properties of $\mathcal{K}_s$ [78, 114, 99]. Furthermore, by directly minimizing these cost functions, one can construct nonlinear basis sets using complex machine learning approaches such as tensor-product algorithms or neural networks [115, 98].

While attempts have been made to define a theory of chemical dynamics purely in terms of the transition operator's eigenfunctions and eigenvalues [97], most chemical theories require dynamical quantities such as committors and mean first-passage times. In this work, we show that it is possible to construct estimates of these quantities using a general basis expansion. Just as DOEA schemes extend MSM estimates of spectral properties to general basis functions, our formalism generalizes MSM estimation of the quantities used in rate theory.

3.3 The Generator and Chemical Kinetics

Many key quantities in chemical kinetics solve operator equations acting on functions of the state space. Below, we give a quick review of this formalism, detailing a few examples of

chemically relevant quantities that can be expressed in this manner. These include statistics such as the mean first-passage time, forwards and backwards committors, and autocorrelation times. In particular, many of these operator equations are examples of Feynman-Kac formulas. For an in-depth treatment of this formalism, we refer the reader to references [116, 117].

In this work, we focus on analyzing data gathered from experiment or simulation. We expect the data to consist of a series of measurements collected at a fixed time interval. Therefore, rather than considering the dynamics of $\xi^{(t)}$, we will consider the dynamics of a discrete-time process $\Xi^{(t)}$ constructed by recording ξ every Δt units of time. If Δt is sufficiently small, this should not appreciably change any kinetic quantities.

In the discussion that follows, we choose to work with the generator of $\Xi^{(t)}$, defined as

$$\mathcal{L}f(x) = \frac{\mathcal{K}_{\Delta t}f(x) - f(x)}{\Delta t}, \tag{3.13}$$

instead of the transition operator. This makes no mathematical difference, but using $\mathcal{L}$ simplifies the presentation. We also stress that, with the exception of (3.24) below, the equations that follow hold only for a lag-time of $s = \Delta t$. For larger lag times, i.e. $s > \Delta t$, these equations only hold approximately. This is discussed further in Section 3.6.

3.3.1 Equations using the Generator

We begin by considering the mean first-passage time and forward committor, two central quantities in chemical kinetics [47, 118, 119]. Let A and B be disjoint subsets of state space and let τ_A be the first time the system enters A:

$$\tau_A = \min\left\{t \geq 0 | \Xi^{(t)} \in A\right\}. \tag{3.14}$$

The *mean first-passage time* is the expectation of τ_A, conditioned on the dynamics starting at x:

$$m_A(x) = \mathbf{E}\left[\tau_A | \Xi^{(0)} = x\right]. \tag{3.15}$$

Note that $1/m_A(x)$ is a commonly used definition of the rate [47]. The *forward committor* is defined as the probability of entering B before A, conditioned on starting at x:

$$q_+(x) = \mathbf{P}\left[\tau_B < \tau_A | \Xi^{(0)} = x\right]. \tag{3.16}$$

Both of these quantities solve operator equations using the generator. The mean first-passage obeys the operator equation

$$\begin{aligned} \mathcal{L} m_A(x) &= -1 \text{ for } x \text{ in } A^c \\ m_A(x) &= 0 \text{ for } x \text{ in } A. \end{aligned} \tag{3.17}$$

Here A^c denotes the set of all state space configurations not in A. Equation (3.17) can be derived by conditioning on the first step of the dynamics. For all x in A^c we have

$$\begin{aligned} m_A(x) &= \mathbf{E}\left[\tau_A | \Xi^{(0)} = x\right] \\ &= \mathbf{E}\left[m_A\left(\Xi^{(\Delta t)}\right) + \Delta t \middle| \Xi^{(0)} = x\right] \\ &= \mathbf{E}\left[m_A\left(\Xi^{(\Delta t)}\right) \middle| \Xi^{(0)} = x\right] + \Delta t \\ &= \mathcal{K}_{\Delta t} m_A(x) + \Delta t \end{aligned}$$

where the second line follows from the time-homogeneity of Ξ. Rearranging then gives (3.17).

We can show that the forward committor obeys

$$\begin{aligned} \mathcal{L}q_+(x) &= 0 \text{ for } x \text{ in } (A \cup B)^c \\ q_+(x) &= 0 \text{ for } x \text{ in } A \\ q_+(x) &= 1 \text{ for } x \text{ in } B \end{aligned} \tag{3.18}$$

by similar arguments. We introduce the random variable

$$\mathbf{1}_{\tau_B<\tau_A} = \begin{cases} 1 \text{ if } \tau_B < \tau_A \\ 0 \text{ otherwise.} \end{cases} \tag{3.19}$$

For all x outside A and B, we can then write

$$\begin{aligned} q_+(x) &= \mathbf{E}\left[\mathbf{1}_{\tau_B<\tau_A} \middle| \Xi^{(0)} = x\right] \\ &= \mathbf{E}\left[q_+\left(\Xi^{(\Delta t)}\right) \middle| \Xi^{(0)} = x\right] \\ &= \mathcal{K}_{\Delta t} q_+(x), \end{aligned}$$

which gives (3.18) on rearranging.

3.3.2 Expressions using Adjoints of the Generator

Additional quantities can be characterized using adjoints of the generator. We reintroduce the sampling measure μ from Section 3.2, and define the inner product

$$\langle u, v \rangle = \int u(x)v(x)\mu(dx). \tag{3.20}$$

Equipped with this inner product, the space of all functions that are square-integrable against μ forms a Hilbert space that that we denote as L^2_μ. The unweighted adjoint of $\mathcal{L}$ is the

operator $\mathcal{L}^\dagger$ such that for all u and v in the Hilbert space,

$$\left\langle \mathcal{L}^\dagger u, v \right\rangle = \langle u, \mathcal{L} v \rangle . \tag{3.21}$$

We now assume that the system has a unique stationary measure. The change of measure from μ to the stationary measure is defined as the function π such that

$$\int \mathbf{E}\left[f\left(\Xi^{(t)}\right) | \Xi^{(0)} = x \right] \pi(x)\mu(dx) = \int f(x)\pi(x)\mu(dx) \tag{3.22}$$

or equivalently,

$$\int \mathcal{L} f(x)\pi(x)\mu(dx) = 0 \tag{3.23}$$

holds for all functions f. As an example, if the system's state space is Euclidean space and the dynamics are stationary at thermal equilibrium, we would have

$$\pi(x)\mu(dx) \propto e^{\frac{H(x)}{k_B T}} dx$$

where $H(x)$ is the system's Hamiltonian, T is the system's temperature, and k_B is the Boltzmann constant. However, this relation is not necessarily true for general state spaces or for nonequilibrium stationary states.

The change of measure to the stationary measure can be written as the solution to an expression with $\mathcal{L}^\dagger$. Interpreting (3.23) as an inner product, the definition of the adjoint implies

$$0 = \langle \pi, \mathcal{L} f \rangle = \left\langle \mathcal{L}^\dagger \pi, f \right\rangle$$

for all f, or equivalently

$$\mathcal{L}^\dagger \pi(x) = 0. \tag{3.24}$$

Other equations may use weighted adjoints of $\mathcal{L}$. Let p be the change of measure from μ to another, currently unspecified measure. The *p-weighted adjoint* of $\mathcal{L}$ is the operator $\mathcal{L}_p^\dagger$

such that

$$\langle u, p\mathcal{L}v\rangle = \left\langle \mathcal{L}_p^\dagger u, pv \right\rangle. \tag{3.25}$$

A few manipulations show that the weighted adjoint can be expressed as

$$\mathcal{L}_p^\dagger f(x) = \frac{1}{p(x)}\mathcal{L}^\dagger\left(fp\right)(x). \tag{3.26}$$

This reduces to the unweighted adjoint when $p(x) = 1$.

One example of a formula that uses a weighted adjoint is a relation for the *backwards committor*. The backwards committor is the probability that, if the system is observed at configuration x and the system is in the stationary state, the system exited state A more recently than state B. It satisfies the equation

$$\begin{aligned} \mathcal{L}_\pi^\dagger q_-(x) &= 0 \text{ for } x \text{ in } (A \cup B)^c \\ q_-(x) &= 1 \text{ for } x \text{ in } A \\ q_-(x) &= 0 \text{ for } x \text{ in } B. \end{aligned} \tag{3.27}$$

Finally, we note that some quantities in chemical dynamics require the solution to multiple operator equations. For instance, in transition path theory [48] the *total reactive current* and *reaction rate* between A and B require evaluating the backwards committor and the forward committor, followed by another application of the generator. The total reactive current between the two sets is given by

$$I_{AB} = \int q_-(x)\mathbb{1}_C(x)\mathcal{L}\left(\mathbb{1}_{C^c}q_+\right)(x)\pi(x)\mu(dx) - \int q_-(x)\mathbb{1}_{C^c}(x)\mathcal{L}\left(\mathbb{1}_C q_+\right)(x)\pi(x)\mu(dx). \tag{3.28}$$

Here C is a set that contains B but not A. The reaction rate constant is then given by

$$k_{AB} = \frac{I_{AB}}{\int q_-(x)\pi(x)\mu(dx)}. \tag{3.29}$$

We derive these expressions in the Supplementary Material in Section 3.9.3 through arguments very similar to those presented in Reference [120].

Evaluating the *integrated autocorrelation time* (IAT) of a function requires estimating π, as well as solving an equation using the generator. For a function with $\int f(x)\pi(x)\mu(dx) = 0$, the IAT is the sum over the correlation function

$$t_f = \left(2\sum_{i=0}^{\infty} \frac{\int f(x)\mathcal{K}_{i\Delta t} f(x)\pi(x)\mu(dx)}{\int \left(f(x)\right)^2 \pi(x)\mu(dx)} - 1 \right) \Delta t \tag{3.30}$$

and, using the Neumann series representation [121] of the appropriate pseudo-inverse of $\mathcal{L}$, can be expressed as

$$t_f = 2\frac{\int f(x)\omega(x)\pi(x)\mu(dx)}{\int f(x)^2\pi(x)\mu(dx)} - \Delta t, \tag{3.31}$$

where ω is the solution to the equation

$$\mathcal{L}\omega(x) = f(x) \tag{3.32}$$

constrained to have $\int \omega(x)\pi(x)\mu(dx) = 0$.

Note that although the quantities above give us information about the long-time behavior of the system, the formalism introduced here only requires information over short time intervals. This suggests that solving these equations directly could lead to a numerical strategy for estimating these long-time statistics from short-time data.

3.4 Dynamical Galerkin Approximation

Inspired by the theory behind DOEA and MSMs, we seek to solve the equations in Section 3.3 in a data-driven manner. We first note that the equations follow the general form

$$\begin{aligned} \mathcal{L}g(x) =& h(x) \text{ for } x \text{ in } D \\ g(x) =& b(x) \text{ for } x \text{ in } D^c \end{aligned} \tag{3.33}$$

or

$$\begin{aligned} \mathcal{L}_p^\dagger g(x) =& h(x) \text{ for } x \text{ in } D \\ g(x) =& b(x) \text{ for } x \text{ in } D^c. \end{aligned} \tag{3.34}$$

Here D is a set in state space that constitutes the *domain*, g is the unknown solution, and h and b are known functions. If the generator and its adjoints are known, these equations can in principle be solved numerically [122, 123, 124]. However, this is generally not the case, and even if the operators are known, the dimension of the full state space is often too high to allow numerical solution. In our approach, we use approximations similar to (3.11) and (3.12) to estimate these quantities from trajectory data. This procedure only requires collections of short trajectories of the system, and works when the dynamical operators are not known explicitly.

We first discuss operator equations using the generator; equations using an adjoint require only slight modification and are discussed at the end of the subsection. We construct an approximation of the operator equation through the following steps.

1. *Homogenize boundary conditions:* If necessary, rewrite (3.33) as a problem with homogeneous boundary conditions using a guess for g.

2. *Construct a Galerkin scheme:* Approximate the solution as a sum of basis functions and convert the result of step 1 into a matrix equation.

3. *Approximate inner products with trajectory averages:* Approximate the terms in the Galerkin scheme using trajectory averages and solve for an estimate of g.

Since we use dynamical data to estimate the terms in a Galerkin approximation, we refer to our scheme as *Dynamical Galerkin Approximation* (DGA).

3.4.1 Homogenizing the Boundary Conditions

First, we rewrite (3.33) as a problem with homogeneous boundary conditions. This allows us to enforce the boundary conditions in step 2 by working within a vector space where every function vanishes at the boundary of the domain. If the boundary conditions are already homogeneous, either because b is explicitly zero or because D includes all of state space, this step can be skipped. We introduce a guess function r that is equal to b on D^c. We then rewrite (3.33) in terms of the difference between the guess and the true solution:

$$\gamma(x) = g(x) - r(x). \tag{3.35}$$

This converts (3.33) into a problem with homogeneous boundary conditions:

$$\mathcal{L}\gamma(x) = h(x) - \mathcal{L}r(x) \text{ for } x \text{ in } D \tag{3.36}$$

$$\gamma(x) = 0 \text{ for } x \text{ in } D^c. \tag{3.37}$$

A naive guess can always be constructed as

$$r^{naive}(x) = \mathbb{1}_{D^c}(x)b(x), \tag{3.38}$$

but if possible, one should attempt to choose r so that γ can be efficiently expressed using the basis functions introduced in step 2.

3.4.2 Constructing the Galerkin Scheme

We now approximate the solution of (3.36) and (3.37) via basis expansion using the formalism of Galerkin approximation. Equation (3.36) implies that

$$\langle u\mathbb{1}_D, \mathcal{L}\gamma\rangle = \langle u\mathbb{1}_D, h\rangle - \langle u\mathbb{1}_D, \mathcal{L}r\rangle \tag{3.39}$$

holds for all u in the Hilbert space L^2_μ. This is known as the *weak formulation* of (3.36) [125].

The space L^2_μ is typically infinite dimensional. Consequently, we cannot expect to ensure that (3.39) holds for every function in L^2_μ. We therefore attempt to solve (3.39) only on a finite-dimensional subspace of L^2_μ. To do this, we introduce a set of M linearly independent functions denoted $\{\phi_1, ..., \phi_M\}$ that obey the homogeneous boundary conditions; we refer to these as the *basis functions*. The space of all linear combinations of the basis functions forms a subspace in L^2_μ which we call the *Galerkin subspace*, G. By construction, every function in G obeys the homogeneous boundary conditions. We now project (3.39) onto this subspace, giving the approximate equation

$$\langle \tilde{u}, \mathcal{L}\tilde{\gamma} \rangle = \langle \tilde{u}, h \rangle - \langle \tilde{u}, \mathcal{L}r \rangle \tag{3.40}$$

for all $\tilde{u}$ in G. Here $\tilde{\gamma}$ is the projection of γ onto G. Constructing G using a linear combination of basis functions that obey the homogeneous boundary conditions ensures that $\tilde{\gamma}$ obeys the homogeneous boundary conditions as well. If we had constructed G using arbitrary basis functions, this would not be true. As we increase the dimensionality of G, we expect the error between γ and $\tilde{\gamma}$ to become arbitrarily small. Since $\tilde{u}$ is in G, it can be written as a linear combination of basis functions. Consequently, if

$$\langle \phi_i, \mathcal{L}\tilde{\gamma} \rangle = \langle \phi_i, h \rangle - \langle \phi_i, \mathcal{L}r \rangle$$

holds for all ϕ_i, then (3.40) holds for all $\tilde{u}$. Moreover, the construction of G implies that there exist unique coefficients a_j such that

$$\tilde{\gamma}(x) = \sum_{j=1}^{M} a_j \phi_j(x), \tag{3.41}$$

enabling us to write

$$\sum_{j=1}^{M} L_{ij} a_j = h_i - r_i \tag{3.42}$$

where

$$L_{ij} = \langle \phi_i, \mathcal{L}\phi_j \rangle \tag{3.43}$$

$$h_i = \langle \phi_i, h \rangle \tag{3.44}$$

$$r_i = \langle \phi_i, \mathcal{L}r \rangle . \tag{3.45}$$

If the terms in (3.43)-(3.45) are known, (3.42) can be solved for the coefficients a_j, and an estimate of g can be constructed as

$$\tilde{g}(x) = r(x) + \sum_{j=1}^{M} a_j \phi_j(x). \tag{3.46}$$

Since $\tilde{\gamma}$ is zero on D^c and r obeys the inhomogeneous boundary conditions by construction,

$$\tilde{g} = r(x) = b(x) \text{ for } x \text{ in } D^c. \tag{3.47}$$

Consequently, our estimate of g obeys the boundary conditions.

A similar scheme can be constructed for equations with a weighted adjoint $\mathcal{L}_p^\dagger$ by adding one additional step to the procedure. After homogenizing the boundary conditions, we multiply both sides of (3.36) by p. We then proceed as before, and obtain (3.42) with terms defined as

$$L_{ij} = \left\langle \phi_i, p\mathcal{L}_p^\dagger \phi_j \right\rangle = \langle \mathcal{L}\phi_i, p\phi_j \rangle \tag{3.48}$$

$$h_i = \langle \phi_i, ph \rangle \tag{3.49}$$

$$r_i = \left\langle \phi_i, p\mathcal{L}_p^\dagger r \right\rangle = \langle \mathcal{L}\phi_i, pr \rangle \tag{3.50}$$

instead of (3.43), (3.44), and (3.45) respectively.

3.4.3 Approximating Inner Products through Monte Carlo

Solving for a_j in (3.41) requires estimates of the other terms in (3.42). In general, these terms cannot be evaluated directly, due to the complexity of the dynamical operators and the high dimensionality of these integrals. However, we can estimate these terms using trajectory averages, in the style of the estimates in (3.11). Let $\rho_{\Delta t}$ be the joint probability measure of $\Xi^{(0)}$ and $\Xi^{(\Delta t)}$, such that for two sets X and Y in state space,

$$\int_{X,Y} \rho_{\Delta t}(dx, dy) = \mathbf{P}[\Xi^{(0)} \in X, \Xi^{(\Delta t)} \in Y]. \tag{3.51}$$

We observe that

$$\begin{aligned} \langle u, \mathcal{L}v \rangle &= \int u(x) \frac{\mathbf{E}\left[v\left(\Xi^{(\Delta t)}\right) | \Xi^{(0)} = x\right] - v(x)}{\Delta t} \mu(dx) \\ &= \int u(x) \frac{v(y) - v(x)}{\Delta t} \rho_{\Delta t}(dx, dy). \end{aligned} \tag{3.52}$$

We now assume that we have a dataset of the form described in Section 3.2.1, with a lag time of Δt. Since each pair (X_n, Y_n) is a draw from $\rho_{\Delta t}$, (3.52) can be approximated using the Monte Carlo estimate

$$\overline{\langle u, \mathcal{L}v \rangle} = \frac{1}{N} \sum_{n=1}^{N} u(X_n) \frac{v(Y_n) - v(X_n)}{\Delta t}. \tag{3.53}$$

Similarly, inner products of the form $\langle u, v \rangle$ can be estimated as

$$\overline{\langle u, v \rangle} = \frac{1}{N} \sum_{n=1}^{N} u(X_n) v(X_n). \tag{3.54}$$

If the Galerkin scheme arose from an equation with a weighted adjoint, evaluating the expectations in (3.48) and (3.50) may require knowing p a priori. However, if $p = \pi$, one can

construct an estimate of π by applying the DGA framework to equation (3.24).

3.4.4 Pseudocode

The DGA procedure can thus be summarized as follows.

1. Sample N pairs of configurations (X_n, Y_n), where Y_n is the configuration resulting from propagating the system forward from X_n for time Δt.

2. Construct a set of M basis functions ϕ_i obeying the homogeneous boundary conditions and, if needed, the guess function r.

3. Estimate the terms in (3.42) using the expressions in (3.4.3).

4. Solve the resulting matrix equations for the coefficients and substitute them into (3.46) to construct an estimate of the function of interest.

Some DGA estimates may require additional manipulation to ensure physical meaning. For instance, change of measures and expected hitting times are nonnegative, and committors are constrained to be between zero and one. These bounds are not guaranteed to hold for estimates constructed through DGA. To correct this, we apply a simple postprocessing step, and round the DGA estimate to the nearest value in the range. Alternatively, constraints on the mean of the solution (e.g., that for ω below (3.32)) can be applied by subtracting a constant from the estimate.

Finally, many dynamical quantities require evaluation of additional inner products. For instance, to estimate the autocorrelation time, t_f, one must construct approximations to ω and π and set ω to have zero mean against $\pi(x)\mu(dx)$. One would then evaluate the numerator and denominator of (3.31) using (3.54).

To aid the reader in constructing estimates using this framework, we have written a Python package for creating DGA estimates [126] This package also contains code for constructing the basis set we introduce in Section 3.5.. As part of the documentation, we have included Jupyter notebooks to aid the reader in reproducing the calculations in this work.

3.4.5 Connection with Other Schemes

As we have previously discussed, this formalism is closely related to DOEA. Rather than considering the solution for a linear system, we could construct a Galerkin scheme for the eigenfunctions of $\mathcal{L}$. Since $\mathcal{L}$ and $\mathcal{K}_s$ have the same eigenfunctions, in the limit of infinite sampling this would give equivalent results to the scheme in 3.2.2. DOEA techniques have also been extended to solve (3.24) [127]. A similar algorithm for addressing boundary conditions has also been suggested in the context of the data-driven study of partial differential equations and fluid flows [128].

Our scheme is also closely related to Markov state modeling. Let ϕ_i be a basis set of indicator functions on disjoint sets S_i covering the state space. Under minor restrictions, applying DGA with this basis is equivalent to estimating the quantities in 3.3 with an MSM. We give a more thorough treatment in the Supplementary Material in Section 3.9.1; here we quickly motivate this connection by examining (3.43) for this particulary choice of basis. We note that we can divide both sides of (3.42) by $\int \phi(x)\mu\,(dx)$ without changing the solution. For this choice of basis, we would then have

$$\frac{L_{ij}}{\int \phi(x)\mu(dx)} = \frac{1}{\Delta t}\,(P - I)_{ij} \tag{3.55}$$

where P is the MSM transition matrix defined in (3.3) and I is the identity matrix. Because of this similarity, we refer to a basis set constructed in this manner as an "MSM" basis.

3.5 Basis Construction using Diffusion Maps

One natural route to improving the accuracy of DGA schemes is to improve the set of basis functions ϕ, thus reducing the error caused by projecting the operator equation onto the finite-dimensional subspace. Various approaches have been used to construct basis sets for describing dynamics in DOEA schemes [105, 87, 88, 112, 113]. However, if D^c is nonempty these functions cannot be used in DGA. In particular, the linear basis in TICA cannot be

used. Here, we provide a simple method for constructing basis functions with homogeneous boundary conditions based on the technique of diffusion maps [129, 130].

Diffusion maps are a technique shown to have success in finding global descriptions of molecular systems from high-dimensional input data [131, 132, 133, 134, 135, 136]. A simple implementation proceeds by constructing the transition matrix

$$P_{mn}^{\mathrm{DMAP}} = \frac{K_\varepsilon(x_m, x_n)}{\sum_n K_\varepsilon(x_m, x_n)}, \tag{3.56}$$

where K_ε is a kernel function that decays exponentially with the distance between datapoints x_m and x_n at a rate set by ε. Multiple choices of K_ε exist; we give the algorithm used to construct the kernel in the Supplementary Material in Section 3.9.2. The eigenvectors of P^{DMAP} with M highest positive eigenvalues were historically used to define a new coordinate system for dimensionality reduction. They can also be used as a basis set for DOEA and similar analyses [112, 137, 110]. Here we extend this line of research, showing that diffusion maps can also be used to construct basis functions that obey homogeneous boundary conditions on arbitrary sets as required for use in DGA. We note that the diffusion process represented by P^{DMAP} is not intended as an approximation of the dynamics, but rather as a tool for building the basis functions ϕ_i. In particular, while the P^{DMAP} matrix is typically reversible, this imposes no reversible constraint in the DGA scheme using the basis derived from P^{DMAP}.

To construct a basis set that obeys nontrivial boundary conditions, we first take the submatrix of P^{DMAP} such that $x_m, x_n \in D$. We then calculate the eigenvectors φ_i of this submatrix that have the M highest positive eigenvalues, and take as our basis

$$\phi_i(x) = \begin{cases} \varphi_i(x) \text{ for } x \text{ in } D, \\ 0 \text{ otherwise.} \end{cases} \tag{3.57}$$

In addition to allowing us to define a basis set, P^{DMAP} gives a natural way of constructing

guess functions that obey the boundary conditions. Since (3.56) is a transition matrix, it corresponds to a discrete Markov chain on the data. Therefore, we can construct guesses by solving analogs to (3.33) using the dynamics specified by the diffusion map. For equations using the generator, we solve the problem

$$\sum_n (P^{\mathrm{DMAP}} - I)_{mn} r_n = h(x_m) \text{ for } m \text{ in } D \tag{3.58}$$

$$r_m = b(x_m) \text{ for } m \text{ in } D^c \tag{3.59}$$

where I is the identity matrix. Here the sum runs over all datapoints, not just those in D. The resulting estimate obeys the boundary conditions for all datapoints sampled in D^c.

Equation (3.58) can also be used to construct guesses for equations using weighted adjoints. In principle, one could replace P^{DMAP} with its weighted adjoint against p and solve the corresponding equation. However, r_n still obeys the boundary conditions irrespective of the weighted adjoint used. We therefore take the adjoint of P^{DMAP} with respect to its stationary measure. Since the Markov chain associated with the diffusion map is reversible [129], P^{DMAP} is self-adjoint with respect to its stationary measure and we again solve (3.58). We discuss how to perform out-of-sample extension on the basis and the guess functions in the Supplementary Material in Section 3.9.2.

To help the reader visualize a diffusion-map basis, we analyze a collection of datapoints sampled from the Müller-Brown potential [138], scaled by 20 so that the barrier height is about 7 energy units; we set $k_B T = 1$. This potential is sampled using a Brownian particle with isotropic diffusion coefficient of 0.1 using the BAOAB integrator for overdamped dynamics with a time step of 0.01 time units [139]. Trajectories are initialized out of the stationary measure by uniformly picking 10000 starting locations on the interval $x \in (-2.5, 1.5), y \in (-2.5, 1.5)$. Initial points with potential energies larger than 100 are rejected and resampled to avoid numerical artifacts. Each trajectory is then constructed by simulating the dynamics for 500 steps, saving the position every 100 steps. We then

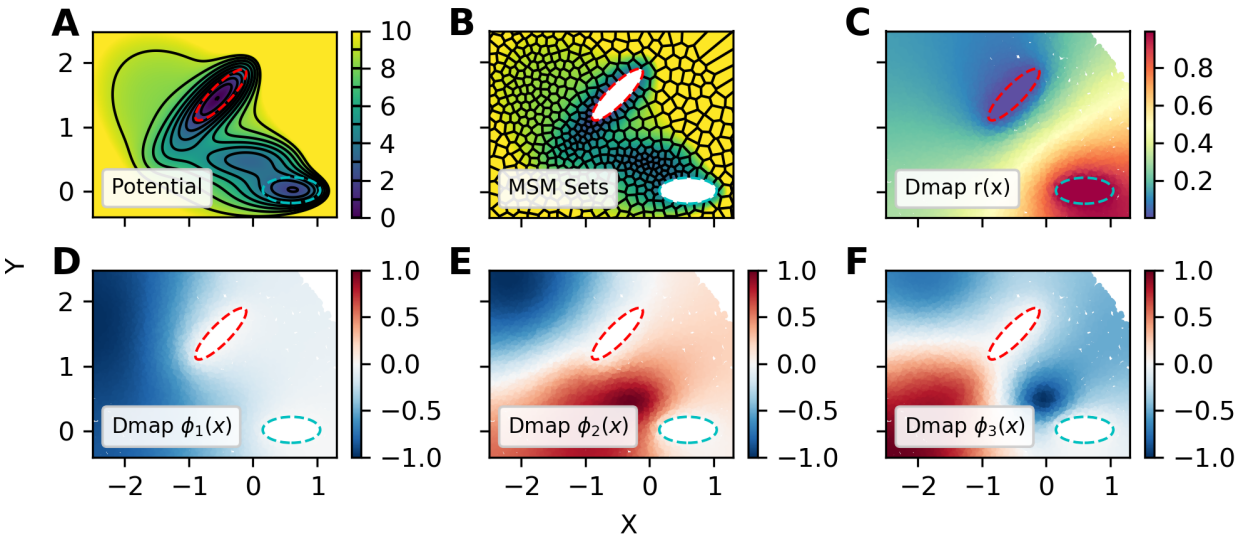


Figure 3.1: Example basis and guess functions constructed by the diffusion-map basis on the scaled Müller-Brown potential. (A) The potential energy surface. Black contour lines indicate the potential energy in units of $k_B T$, red and cyan dotted contours indicate the boundaries of states A and B respectively. (B) An MSM clustering with 500 sets on the domain; the color scale is the same as in (A). Each MSM basis function is one inside a cell and zero otherwise. Sets inside states A and B are not shown to emphasize the boundary conditions. (C) Scatter plot of the guess function for the committor for hitting B before A, constructed using (3.58). (D–F) Scatter plots of the first three basis functions constructed according to (3.57).

define two states A and B (red and cyan dashed contours in Figure 3.1, respectively) and construct the basis and guess functions required for the committor. The results, plotted in Figure 3.1, demonstrate that the diffusion-map basis functions are smoothly varying with global support.

3.5.1 Basis Set Performance in High-Dimensional CV spaces.

We now test the effect of dimension on the performance of the basis set by attempting to calculate the forward committor and total reactive flux for a series of toy systems based on the model above. To be able to vary the dimensionality of the system, we add up to 18

harmonic "nuisance" degrees of freedom. Specifically,

$$U\left(x, y, z_3, ..., z_d\right) = U_{\mathrm{MB}}(x, y) + \sum_{l=3}^{d} z_l^2, \tag{3.60}$$

where U_{MB} is the scaled Müller-Brown potential discussed above. We compare our results with references computed by a grid-based scheme described in the Supplementary Material. Our reference for the committor is plotted in Figure 3.3A. We initialize the x and y dimensions as above; the initial values of the nuisance coordinates were drawn from their marginal distributions at equilibrium. We then sampled the system using the same procedure as before.

Throughout this section and all subsequent numerical comparisons, we compare the diffusion-map basis with a basis of indicator functions. Since, with minor restrictions, using a basis of indicator functions is equivalent to calculating the same dynamical quantities using a MSM, we estimate committors, mean first-passage times, and stationary distributions by constructing a MSM in PyEMMA and using established formulas [120, 75, 76]. In general, it is not our intention to compare an optimal diffusion-map basis to an optimal MSM basis. Multiple diffusion-map and clustering schemes exist and performing an exhaustive comparison would require comparison over multiple methods and hyperparameters. We leave such a comparison for future work, and only seek to present reasonable examples of both schemes.

MSM clusters are constructed using k-means, as implemented in PyEMMA [104]. While MSMs are generally constructed by clustering points globally, this does not guarantee that a given clustering satisfies a specific set of boundary conditions. Consequently, we modify the set definition procedure slightly.

We first construct M clusters on the domain D, and then cluster D^c separately. The number of states inside D^c is chosen so that states inside D^c have approximately the same number of samples on average as states in the domain. For the current calculation, this corresponded to approximately one state inside set A or B for every five states outside the

domain; we round to a ratio of 1/5 for numerical simplicity. We note that clustering on the interior of D^c does not affect calculated committors or mean first-passage times. We use 500 basis functions for both the MSM and diffusion-map basis sets. We give plots supporting this choice in Section 3.9.5 of the Supplementary Material.

In modern Markov state modeling, one commonly constructs the transition matrix only over a well-connected subset of states named the active set [83, 140]. We have follow this practice, and exclude points outside the active set from any error analyses of the resulting MSMs. We believe this gives the MSM basis an advantage over the diffusion-map basis in our comparisons, as we are explicitly ignoring points where it fails to provide an answer and would presumably give poor results.

It is also common to ensure that the resulting matrix obeys detailed balance through a maximum likelihood procedure [82, 83]. We choose not to do this because we do not wish to assume reversibility in our formalism. Moreover, our calculations have also shown that enforcing reversibility can introduce a statistical bias that dominates the error in any estimates. We give numerical examples of this phenomenon in Section 3.9.6 of the Supplementary Material.

In Figure 3.2A, we plot root-mean-square error (RMSE) between the estimated and reference forward committors as a function of the number of nuisance degrees of freedom. While for low-dimensional systems the MSM and the diffusion-map basis give comparable results, as we increase the dimensionality, the MSM gives increasingly worse answers. To aid in understanding these results, we plot example forward committor estimates for the 20-dimensional system in Figure 3.3. We see that the diffusion-map basis manages to capture the general trends in the reference in Figure 3.3A. In contrast, the MSM basis gives considerably noisier results.

We also estimate the total reactive flux across the same dataset, setting C and C^c in (3.82) to be the sets on either side of the calculated isocommittor (Figure 3.2B). The large errors that we observe in the reactive flux occur due to the nature of the dataset. If data

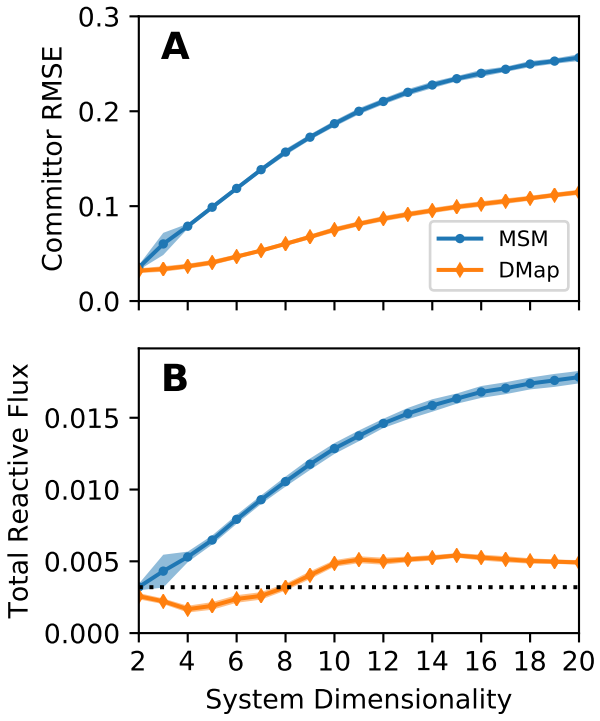


Figure 3.2: Comparison of basis performance as the dimensionality of the toy system increases. (A) The average error in the forward committor between states B and A in Figure 3.3 for both the MSM and the diffusion-map basis functions, as a function of the number of nuisance degrees of freedom. (B) Estimated reactive flux using both MSM and the diffusion-map basis functions as function of the same. In both plots shading indicates the standard deviation over 30 datasets. The dotted line in (B) is the reactive flux as calculated by an accurate reference scheme.

were collected from a long equilibrium trajectory, it would not be necessary to estimate $\pi(x)$ separately, and we could set $\pi(x) = 1$. In that case, provided the number of MSM states was sufficient, the MSM reactive flux reverts to direct estimation of the number of reactive trajectories per unit time. This would give an accurate reactive flux regardless of the quality of the estimated forwards or backwards committors.

3.6 Addressing Projection Error Through Delay Embedding

Our results suggest that improving basis set choice can yield DGA schemes with better accuracy in higher-dimensional CV spaces. However, even large CV spaces are considerably

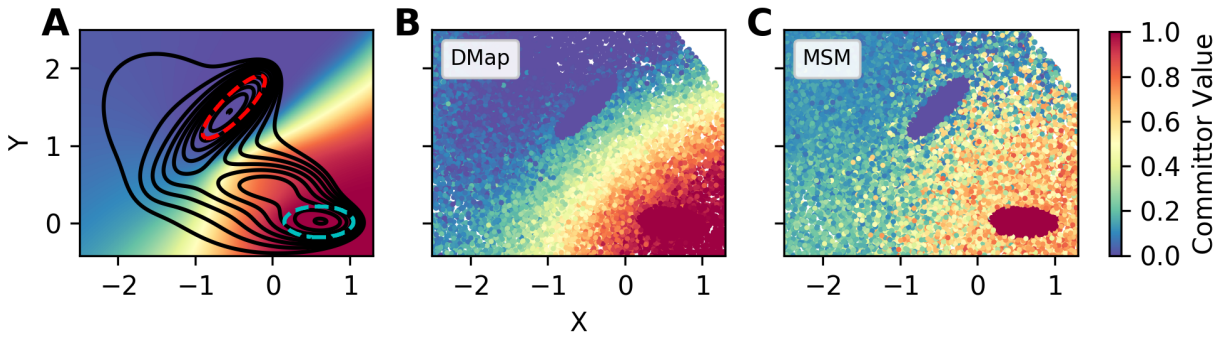


Figure 3.3: Example forward committors calculated using the diffusion-map and MSM bases on a high-dimensional toy problem. The system is the same as in Figure 3.1, with 18 additional nuisance dimensions. (A) Forward committor function calculated using an accurate grid-based scheme. The black lines indicate the contours of free energy in the x and y coordinates, and the red and cyan dashed contours indicate the two states. Every subsequent dimension has a harmonic potential with force constant of 2. (B–C) Estimated forward committor constructed using the diffusion map and MSM bases, respectively.

lower-dimensional than the systems full state space. Consequently, they may still omit key degrees of freedom needed to describe the long-time dynamics. In both MSMs and DOEA, this projection error is often addressed by increasing the lag time of the transition operator [83, 141, 115, 98]. In the long-lag-time limit, bounds on the approximation error for DOEA show that the scheme gives the correct equilibrium averages up to projection [73, 142]. However, MSMs and DOEA cannot resolve dynamics on time scales shorter than the lag time. This is reflected in existing DOEA error bounds on the relative error of the estimate of the subdominant eigenvalue, which do not vanish with increasing lag time [142].

Moreover, whereas changing the lag time does not affect the eigenfunctions in (3.6), the equations in Section 3.3 hold only for a lag time of Δt. Using a longer time is effectively making the approximation

$$\mathcal{L}f(x) \approx \frac{\mathcal{K}_s f(x) - f(x)}{s}. \tag{3.61}$$

This causes a systematic bias in the estimates of the dynamical quantities discussed in Section 3.3. While for small lag times this bias is likely negligible, it may become large as the lag time increases. For instance, estimates of the mean first-passage time grow linearly

with s as the lag time goes to infinity [141].

Here, we propose an alternative strategy for dealing with projection error. Rather than looking at larger time lags, we use past configurations in CV space to account for contributions from the removed degrees of freedom. This idea is central to the Mori-Zwanzig formalism [143]. Here, we use *delay embedding* to include history information. Let $\zeta^{(t)}$ be the projection of $\Xi^{(t)}$ at time t. We define the delay-embedded process with d delays as

$$\theta^{(t)} = \left(\zeta^{(t)}, \zeta^{(t-\Delta t)}, \zeta^{(t-2\Delta t)}, \ldots, \zeta^{(t-d\Delta t)}\right). \tag{3.62}$$

Delay embedding has a long history in the study of deterministic, finite-dimensional systems, where it has been shown that delay embedding can recapture attractor manifolds up to diffeomorphism [144, 145]. Weaker mathematical results have been extended to stochastic systems [146, 147], although these are not sufficient to guarantee its effectiveness in all cases.

Delay embedding has been used previously with dimensionality reduction on both experimental [148] and simulated chemical systems [149, 150], and has also been used in applications of DOEA in geophysics [110]. In references [110, 151] it was argued that delay embedding can improve statistical accuracy for noise-corrupted and time-uncertain data. Other methods of augmenting the dynamical process with history information have been used in the construction of MSMs. In Reference [152], each trajectory was augmented with a labeling variable indicating its origin state. In Reference [141], it was suggested to write transition probabilities as a function of both the current and the preceding MSM state. This corresponds to a specific choice of basis on a delay embedded process.

Here we show that delay embedding can be used to improve dynamical estimates in DGA. To apply DGA to the delay-embedded process, we must extend the functions h and b in (3.33) and (3.33) to the delay-embedded space. We do this by using the value of the function on the central timepoint,

$$f\left(\theta^{(t)}\right) = f\left(\zeta^{(t-\lfloor d/2 \rfloor \Delta t)}\right) \tag{3.63}$$

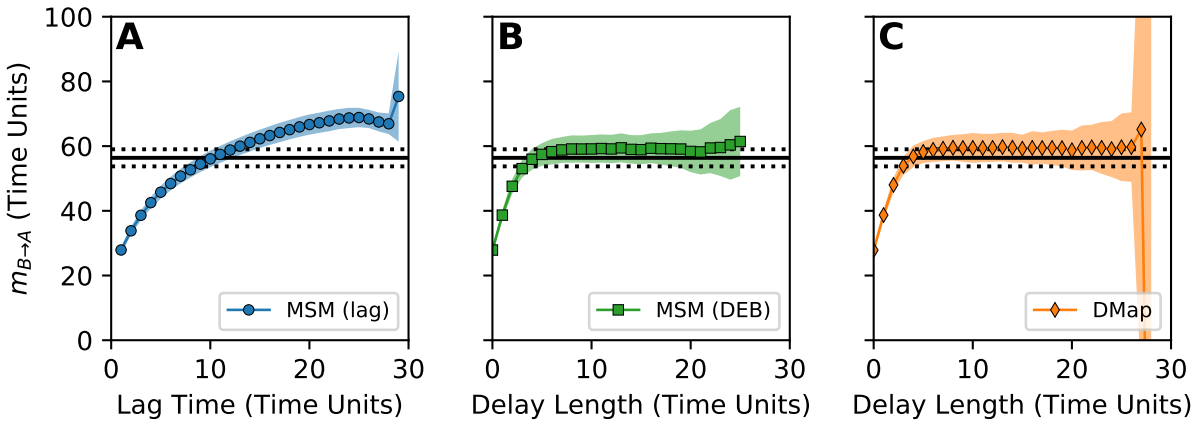


Figure 3.4: Comparison of methods for dealing with the projection error in an incomplete CV space. In all subplots we estimate the mean first-passage time from state $B = \{y < 0.15\}$ to state $A = \{y > 1.15\}$ using a DGA scheme on only the y coordinate of the Müller-Brown potential. (A) Estimate constructed using an MSM basis with increasing lag time in (3.61), as a function of the lag time. (B) Estimate constructed using an MSM basis, but applying delay embedding rather than increasing the lag time, as a function of the delay length. (C) Estimate constructed using the diffusion-map basis with delay embedding, as a function of the delay length. In each plot, the symbols show the mean over 30 identically constructed trajectories, and the shading indicates the standard deviation across trajectories. The black solid line is an estimate of the mean first-passage time calculated using the reference scheme in the Supplementary Material, and the dashed error bars represent the standard deviation of the mean first-passage time over state B.

where $\lfloor \ldots \rfloor$ denotes rounding down to the nearest integer. The states D and D^c in the delay-embedded space are extended similarly. One can easily show that this preserves dynamical quantities such as mean first-passage times and committors. The basis set is then constructed directly on θ, and the DGA formalism is applied as before.

We test the effect of delay embedding in the presence of projection error by constructing DGA schemes on the same system as in Section 3.5 and taking as our CV space only the y-coordinate. For this study we revise our dataset to include 2000 trajectories, each sampled for 3000 time steps. While using longer trajectories changes the density such that it is closer to equilibrium, it allows us to test longer lag times and delay lengths. To ensure that our states are well-defined in this new CV space, we redefine state A to be the set $\{y > 1.15\}$, and state B to be the set $\{y < 0.15\}$. We then estimate the mean first-passage time into

state A, conditioned on starting in state B at equilibrium.

$$m_{B\to A} = \frac{\int \mathbb{1}_B(x) m_A(x) \pi(x) \mu(dx)}{\int \mathbb{1}_B(x) \pi(x) \mu(dx)}.$$

We construct estimates using an MSM basis with varying lag time, an MSM basis with delay embedding, and a diffusion-map basis with delay embedding. In Figure 3.4, we plot the average mean first-passage time as a function of the lag time and the trajectory length used in the delay embedding. We compare the resulting estimates with an estimate of the mean first-passage time constructed using our grid-based scheme. In addition, an implied timescale analysis for the two MSM schemes is given in the Supplementary Material in Section 3.9.7.

The mean first-passage time estimated from the MSM basis with the lag time steadily increases as the lag time becomes longer (Figure 3.4A), as predicted in Reference [141]. In contrast, the estimates obtained from delay embedding both converge as the delay length increases, albeit to a value slightly larger than the reference. We believe this small error is because we treat the dynamics as having a discrete time step, while the reference curve approximates the mean first-passage time for a continuous-time Brownian dynamics. In particular, The latter includes events in which the system enters and exits the target state within the duration of a discrete-time time step, but such events are missing from the discrete-time dynamics.

In all three schemes, we see anomalous behavior as the length of the lag time or delay length increases. This is due to an increase in statistical error when the delay length or lag time becomes close to the length of the trajectory. If each trajectory has N datapoints, performing a delay-embedding with d delays means that each trajectory only gives $N - d$ samples. When N and d are of the same order of magnitude, this leads to increased statistical error in the estimates in 3.4.3, to the point of making the resulting linear algebra problem ill-posed. The diffusion-map basis fluctuates to unreasonable values at long delay lengths,

and the MSM basis fails completely, truncating the curve in Figures 3.4B and C. Similarly, the lagged MSM has an anomalous downturn in the average mean first-passage time near 26 time units. We give additional plots supporting this theory in the Supplementary Material in Section 3.9.7.

Finally, we observe that the delay length required for the estimate to converge is substantially smaller than the mean first-passage time. This suggests that delay embedding can be effectively used on short trajectories to get estimates of long-time quantities.

3.7 Application to the Fip35 WW Domain

To further assess our methods, we now apply them to molecular dynamics data and seek to evaluate committors and mean first-passage times. In contrast to the simulations above, we do not have accurate reference values and cannot directly calculate the error in our estimates. Instead, we observe that both the mean first-passage time and forward committor are conditional expectations, and obey the following relations [153]

$$\begin{aligned} m_A(x) &= \arg\min_{f(x)} \mathbf{E}\left[(\tau_A - f(x))^2\right] \\ q_+(x) &= \arg\min_{f(x)} \mathbf{E}\left[\left(\mathbf{1}_{\tau_B<\tau_A} - f(x)\right)^2\right]. \end{aligned}$$

This suggests a scheme for assessing the quality of our estimates. If we have access to long trajectories, each point in the trajectory has an associated sample of τ_A and $\mathbf{1}_{\tau_B<\tau_A}$. We define the two empirical cost functions

$$\mathrm{COST}_{m_A} = \frac{1}{N}\sum_{n=1}^{N}(\bar{m}_A(x_n) - \tau_{A,n})^2 \tag{3.64}$$

$$\mathrm{COST}_{q_+} = \frac{1}{N}\sum_{n=1}^{N}(\bar{q}_+(x_n) - \mathbf{1}_{\tau_B<\tau_A,n})^2. \tag{3.65}$$

Here x_n is a collection of samples from a long trajectory, $\tau_{A,n}$ is the time from x_n to A, and $\mathbf{1}_{\tau_B<\tau_{A,n}}$ is one if the sampled trajectory next reaches B and zero if it next reaches A. The numerical estimates of the mean first-passage time and committor are written as $\bar{m}_A$ and $\bar{q}$, respectively. In the limit of $N \to \infty$, the true mean first-passage time and committor would minimize (3.64) and (3.65). We consequently expect lower values of our cost functions to indicate improved estimates. For a perfect estimate, however, these cost functions would not go to zero. Rather, in the limit of infinite sampling, (3.64) and (3.65) would converge to the variances of τ_A and $\mathbf{1}_{\tau_B<\tau_A}$. For the procedure to be valid, it is important that the cost estimates are not constructed using the same dataset used to build the dynamical estimates. This avoids spurious correlations between the dynamical estimate and the estimated cost.

We applied our methods to the Fip35 WW domain trajectories described by D.E. Shaw Research in references [154, 155]. The dataset consists of six trajectories, each of length 100000 ns with frames output every 0.2 ns. Each trajectory has multiple folding and unfolding events, allowing us to evaluate the empirical cost functions. To avoid correlations between the DGA estimate and the calculated cost, we perform a test/train split and divide the data into two halves. We choose three trajectories to construct our estimate, and use the other three to approximate the expectations in (3.64) and (3.65). Repeating this for each possible choice of trajectories creates a total of 20 unique test/train splits.

To reduce the memory requirements in constructing the diffusion map kernel matrix, we subsampled the trajectories, keeping every 100th frame. This allowed us to test the scheme over a broad range of hyperparameters. We expect that in practical applications a finer time resolution would be used, and any additional computational expense could be offset by using landmark diffusion maps [156].

To define the folded and unfolded states, we follow reference [92] and calculate $r_{\beta 1}$ and $r_{\beta 1}$, the minimum root-mean-square-displacement for each of the two β hairpins, defined as amino acids 7-23 and 18-29, respectively [92]. We define the folded configuration as having both $r_{\beta 1} < 0.2$ nm and $r_{\beta 2} < 0.13$ nm and the unfolded configuration as having 0.4 nm

$< r_{\beta 1} <$ 1.0 nm and 0.3 nm $< r_{\beta 2} <$ 0.75 nm. For convenience, we refer to these states as A and B throughout this section. We then attempt to estimate the forward committor between the two states and the mean first-passage time into A using the same methods as in Section 3.6.

We take as our CVs the pairwise distances between every other α-carbon, leading to a 153-dimensional space. In previous studies, dimensionality-reduction schemes such as TICA have been applied prior to MSM construction. We choose not to do this, as we are interested in the performance of the schemes in large CV spaces. This also helps control the number of hyperparameters and algorithm design choices. Indeed, our tests suggest that, while using TICA with well-chosen hyperparameters can lead to improvements for both basis sets, the qualitative trends in our results remain unchanged. However, we think the interaction between dimensionality-reduction schemes and families of basis sets merits future investigation.

Our results are given in Figure 3.5. In panels A and B, we give the mean value of the cost for the mean first-passage time and forward committor over all test/train splits, as calculated using 200 basis functions for each algorithm. The number of basis functions was chosen to give the best result for the MSM scheme with increasing lag over any lag time, although we see only very minor differences in behavior for larger basis sets. The large standard deviations primarily reflect variation in the cost across different test/train splits, rather than any difference between the methods. This suggests the presence of large numerical noise in our results.

To get a more accurate comparison, we instead look at the expected improvement in cost between schemes for a given test/train split. To quantify whether an improvement occurs, we first determine the best parameter choice for the MSM basis with increasing lag. We estimate the cost for the MSM basis with delay embedding and for the diffusion-map basis, and calculate the difference in cost versus the lagged MSM scheme for each test/train split. We then average and calculate the standard deviation over pairs, and plot the results in

figures 3.5C through 3.5F. As the difference is calculated against the best parameter choice for the lagged MSM scheme, they are intrinsically conservative: in practice, one should not expect to have the optimal lagged MSM parameters.

In our numerical experiments, we see that the diffusion map seems to give the best results for relatively short delay lengths. However, the diffusion-map basis performs progressively worse as the delay length increases. The mechanism causing this loss in accuracy requires further analysis. This tentatively suggests the use of the diffusion-map basis for datasets consisting of very short trajectories, where using long delays may be infeasible. In contrast, our results with the delay-embedded MSM basis are more ambiguous. For the mean first-passage time, we do not see significant improvement over the results from the lagged MSM results. We do see noticeable improvement in the estimated forward committor probability as the delay length increases. However, we observe that the delay lengths required to improve upon the diffusion map result are comparable in magnitude to the average time required for the trajectory to reach either the A or B states. Indeed, we only see an improvement over the diffusion-map result at a delay length of 180 ns, and we observe that the longest the trajectory spends outside of both state A or state B is 223 ns. This negates any advantage of using datasets of short trajectories.

Caution is warranted in interpreting these results. We see large variances between different test/train splits, suggesting that despite having 300 μs of data in each training dataset, we are still in a relatively data-poor regime. Similarly, we cannot make an authoritative recommendation for any particular scheme for calculating dynamical quantities without further research. Such a study would not only require more simulation data, but also a comparison of multiple clustering and diffusion map schemes across several hyperparameters and their interaction with various dimensionality-reduction schemes. We leave this task for future work. However, our initial results are promising, suggesting that further development of DGA schemes and basis sets is warranted.

3.8 Conclusions

In this paper, we introduce a new framework for estimating dynamical statistics from trajectory data. We express the quantity of interest as the solution to an operator equation using the generator or one of its adjoints. We then apply a Galerkin approximation, projecting the unknown function onto a finite-dimensional basis set. This allows us to approximate the problem as a system of linear equations, whose matrix elements we approximate using Monte Carlo integration on dynamical data. We refer to this framework as *Dynamical Galerkin Approximation* (DGA). These estimates can be constructed using collections of short trajectories initialized from relatively arbitrary distributions. Using a basis set of indicator functions on nonoverlapping sets recovers MSM estimates of dynamical quantities. Our work is closely related to existing work on estimating the eigenfunctions of dynamical operators in a data-driven manner.

To demonstrate the utility of alternative basis sets, we introduce a new method for constructing basis functions based on diffusion maps. Results on a toy system shows that this basis has the potential to give improved results in high-dimensional CV spaces. We also combine our formalism with delay-embedding, a technique for recovering degrees of freedom omitted in constructing a CV space. Applying it to an incomplete, one-dimensional projection of our test system, we see that delay embedding can improve on the current practice of increasing the lag time of the dynamical operator.

We then applied the method to long folding trajectories of the Fip35 WW domain to study the performance of the schemes in a large CV space on a nontrivial biomolecule. Our results suggest that the diffusion-map basis gives the best performance for short delay times, giving results that are as good or better than the best time-lagged MSM parameter choice. Moreover, our results suggest that combining the MSM basis with delay embedding gives promising results, particularly for long delay lengths. However, long delay lengths are required to see an improvement over the diffusion-map basis, potentially negating any computational advantage in using short trajectories to estimate committors and mean first-

passage times.

We believe our work raises new theoretical and algorithmic questions. Most immediately, we hope our preliminary numerical results motivate the need for new approaches to building basis sets and guess functions obeying the necessary boundary conditions. Further theoretical work is also required to assess the validity of using delay embedding in our schemes. Finally, we believe it is worth searching for connections between our work, VAC and VAMP theory [78, 114, 115, 99, 98], and earlier approaches for learning dynamical statistics [57, 58, 133]. In particular, a variational reformulation of the DGA scheme would allow substantially more flexible representation of solutions. With these further developments, we believe DGA schemes have the potential to give further improved estimates of dynamical quantities for difficult molecular problems.

3.9 Appendix

3.9.1 Connection between DGA and Markov State Modeling

Here, we describe in detail the connection between DGA and certain dynamical estimates calculated using a MSM.

To map the general dynamics onto the state space of the Markov Chain, we make three assumptions.

Assumption 3.9.1. *Each Markov state S_i is contained entirely in either D or in D^c.*

Assumption 3.9.2. *The boundary conditions b can be expressed as*

$$b(x) = \sum_{l \in D^c}^{M'} b_l \mathbb{1}_{S_l}(x). \tag{3.66}$$

Assumption 3.9.3. *For any $\mathcal{L}_p^\dagger$ considered, p can be written as*

$$p(x) = \sum_{j \in D}^{M} \frac{p_j}{\langle \mathbb{1}_j \rangle} \mathbb{1}_j(x) + \sum_{l \in D^c}^{M'} \frac{p_l}{\langle \mathbb{1}_l \rangle} s \mathbb{1}_l(x). \tag{3.67}$$

The first assumption is necessary for the basis set to obey the homogeneous boundary conditions, and can be enforced explicitly in the construction of the MSM. The second two assumptions will be required to make the action of $\mathcal{L}$ representable as the action of matrices on vectors over the MSM states. While these assumptions should not be expected to hold for general b and p, in the correct limit of infinite sampling and sufficiently small Markov states, we expect (3.66) and (3.67) to be arbitrarily good approximations. In fact, for most b in Section 3.3, assumption 3.9.2 can hold exactly. We also note that the vector p_i sums to one, as

$$\begin{aligned} 1 &= \int p(x) \mu(dx) \\ &= \int \sum_{j \in D}^{M} \frac{p_j}{\langle \mathbb{1}_j \rangle} \mathbb{1}_j(x) + \sum_{l \in D^c}^{M'} \frac{p_l}{\langle \mathbb{1}_l \rangle} \mathbb{1}_l(x) \mu(dx) \\ &= \sum_{j \in D}^{M} p_j + \sum_{l \in D^c}^{M'} p_l. \end{aligned}$$

Consequently, p_i is a probability distribution over the MSM state-space.

Equations with the Transition Operator

We first consider equations that take the form of (3.33). As our guess, we will use (3.38). Substituting into (3.42), applying Assumption 3.9.2, and dividing by $\langle 1_{S_i} \rangle$, we arrive at

$$\sum_{j \in D}^{M} \frac{1}{\Delta t} (P - I)_{ij} a_j = \eta_i - \sum_{l \in D^c}^{M'} \frac{1}{\Delta t} (P - I)_{il} b_l. \tag{3.68}$$

Here P_{ij} is the MSM transition matrix defined in (3.3) with a time lag of Δt, and η_i is defined as

$$\eta_i = \frac{\langle \mathbb{1}_i, h \rangle}{\langle \mathbb{1}_i \rangle} \tag{3.69}$$

This can be rewritten as

$$\begin{aligned} \sum_j \frac{1}{\Delta t} \left(P - I\right)_{ij} a_j = \eta_i \text{ for } i \in D \\ a_i = b_i \text{ for } i \in D^c \end{aligned} \tag{3.70}$$

where the sum is over states on the entire domain. This is equivalent to (3.33) for the dynamics given by the MSM.

Equations with Transition Adjoints

For equations that take the form of (3.34) we again begin with (3.42), this time with terms defined by equations (3.48), (3.50), and (3.49). Substituting in our guess function and Assumptions 3.9.3 and 3.9.1, we have

$$\sum_{j \in D}^{M} \langle \mathcal{L}\mathbb{1}_i, \mathbb{1}_j \rangle \left(\frac{p_j}{\langle \mathbb{1}_j \rangle} \right) a_j = \left(\frac{p_i}{\langle \mathbb{1}_i \rangle} \right) \langle \mathbb{1}_i, h \rangle - \sum_{l \in D^c}^{M'} \langle \mathcal{L}\mathbb{1}_i, \mathbb{1}_l \rangle \, b_l \left(\frac{p_l}{\langle \mathbb{1}_l \rangle} \right) \tag{3.71}$$

We then divide both sides by p_i. Applying the definition of P_{ij}, we arrive at

$$\sum_{j \in D}^{M} p_i^{-1} \left(P - I\right)_{ij}^T p_j a_j = \eta_i - \sum_{l \in D^c}^{M'} p_i^{-1} \left(P - I\right)_{il} p_l b_l. \tag{3.72}$$

which, as before, is equivalent to solving

$$\begin{aligned} \sum_j p_i^{-1} \left(P - I\right)_{ij}^T p_j = \eta_i \text{ for } i \in D \\ a_i = b_i \text{ for } i \in D^c. \end{aligned} \tag{3.73}$$

Comparing with (3.26), we see that the matrix with elements $p_i^{-1}\,(P-I)_{ij}^T\,p_j$ is the weighted adjoint of the MSM generator against p_i. Consequently, (3.73) is equivalent to (3.34) for the MSM.

3.9.2 Details of Diffusion Map Construction

Here, we give the specific kernel and parameter choice used in our calculations used to construct the diffusion map in our calculations. Our procedure closely follows work in references [137] and [130]. Specifically, our algorithm corresponds to the parameter choice $\alpha = 0$ and $\beta = -1/d$ in reference [130] and not performing the bandwidth normalization in equation (5).

Kernel Construction

As in Section 3.5, let x_m be a collection of N datapoints. We define the initial bandwidth function

$$\varsigma_0(x_m) = \frac{1}{k_0}\sum_{l=1}^{k_0} ||x_m - x_{I(m,l)}||^2$$

where $I(m,l)$ is the index of the l'th nearest neighbor to point x_m (not including x_m). Here k_0 is a neighborhood parameter giving the number of nearest neighbors considered, we follow reference [130] and set it to 7. We then construct the kernel density estimate

$$q(x_m) = \frac{(2\pi\varepsilon_0)^{-d/2}}{N\varsigma_0(x_m)^d}\sum_{n=1}^{N} K_0(x_m, x_n; \varepsilon_0), \text{ where}$$
$$K_0(x_m, x_n; \varepsilon_0) = \exp\left(\frac{-||x_m - x_n||^2}{2\varepsilon_0\varsigma_0(x_m)\varsigma_0(x_n)}\right)$$

where d is the intrinsic dimensionality of the data manifold and ε_0 is a bandwidth parameter. To estimate d and a good choice for ε_0, we consider all possible choices of eps_0 of the form

2^k with $k = -40, -39, \ldots, 39, 40$. We then set

$$d = \frac{2}{\ln(2)} \max_k \left[\ln \left(\frac{\sum_{m,n} K_0(x_m, x_n, 2^{k+1})}{\sum_{m,n} K_0(x_m, x_n, 2^k)} \right) \right] \tag{3.74}$$

Reference [137] suggests setting ε_0 by using the k where the right-hand-side attains its maximum. In practice we find this can be overly aggressive, so we subsequently multiply ε_0 by 2. We then construct the Diffusion map kernel matrix as

$$K(x_m, x_n) = \exp\left(\frac{||x - y||^2}{\varepsilon q(x_m)^{-1/d} q(x_n)^{-1/d}} \right) \tag{3.75}$$

where we select the ε using the same procedure as before.

Out-of-sample Extension for the Diffusion-Map Basis

To predict the values of the quantities in Section 3.3 at new datapoints, we will need to extend the diffusion-map basis and guess functions to new configurations. Initially, one might attempt this by constructing a new diffusion map matrix that contains both the old and the new points and recomputing the guess and eigenvectors. However, not only would this procedure be expensive, it would change the values of the basis and guess functions on the old points. Consequently, the estimates of a_j would be incorrect, and the entire DGA scheme would need to be repeated. We therefore seek a method for extending the basis and guess functions to new points that leave their values on older points unchanged.

Let x_ν be a new point added to the dataset. To extend the basis functions to x_ν, we can use the established method of Nyström extension [157, 156]. Let φ_i be an eigenvector of the submatrix discussed in Section 3.5, and let κ_i be the associated eigenvalue. The estimate of the basis function on x_ν is given by

$$\varphi_i(x_\nu) = \frac{1}{\kappa_i} \frac{\sum_m K_\varepsilon(x_m, x_\nu) \varphi_i(x_m)}{\sum_m K_\varepsilon(x_m, x_\nu)} \tag{3.76}$$

To extend the guess function to new configurations, we introduce a new method based on the Jacobi method [158]. We first consider $\hat{P}$, a new diffusion map matrix built using both the old datapoints $x_{1..N}$ and the new datapoint x_ν. The guess function associated with $\hat{P}$ would then solve the problem

$$\left(\hat{P} - I\right) g = h \tag{3.77}$$

for all of the points in D. We will construct our estimate of g at the new point by considering a single iteration of the Jacobi method for solving (3.77). Our initial vector takes values of g_m on x_m and 0 on x_ν. This gives us the following out-of-sample extension formula

$$g_\nu = \frac{1}{\hat{P}_{\nu\nu} - 1} \left(h_\nu - \sum_{m=1}^{N} \hat{P}_{\nu m} g_m \right) \tag{3.78}$$

where the sum runs only over points in the original dataset. This can be further simplified using the definition of $\hat{P}$ to

$$g_\nu = \frac{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right) g_m}{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right)} - h_\nu \left(1 + \frac{K_\varepsilon\left(x_\nu, x_\nu\right)}{\sum_{m=1}^{N} K_\varepsilon\left(x_\nu, x_m\right)} \right). \tag{3.79}$$

3.9.3 Derivation of Transition Path Theory Reactive Flux and Rate in Discrete Time

Transition path theory was originally formulated for diffusion processes [48] and was extended to finite-state Markov jump processes [120]. Here, we derive analogous equations for discrete-time Markov chains on arbitrary state spaces. The derivation closely follows reference [48]. Let $x(t)$ be a single trajectory ergodically sampling the stationary measure. We will extend the trajectory both forwards and backwards in time so the time index t takes values from

$-\infty$ to ∞. For all t, let

$$t^+_{AB}(t) = \min\left\{t' | t' \geq t, x(t') \in A \cup B\right\} \tag{3.80}$$

$$t^-_{AB}(t) = \max\left\{t' | t' \leq t, x(t') \in A \cup B\right\} \tag{3.81}$$

be the next time the system entered A or B and the most recent time the system left A or B, respectively. Now let C be as in (3.28). The total reactive current is defined as

$$I_{B\to A} = \lim_{T\to\infty} \frac{1}{2T} \sum_{t\in[-T,T]} \left[\mathbb{1}_C\left(x(t)\right)\mathbb{1}_{C^c}\left(x(t+\Delta t)\right) - \mathbb{1}_{C^c}\left(x(t)\right)\mathbb{1}_C\left(x(t+\Delta t)\right)\right]$$
$$\left[\mathbb{1}_A\left(x(t^-_{AB}(t)\right)\mathbb{1}_B\left(x(t^+_{AB}(t+\Delta t)\right)\right]$$

Using ergodicity and the strong Markov property, we can rewrite this as an average against $\rho_{\Delta t}$.

$$I_{B\to A} = \int \mathbb{1}_{C^c}(y) q_+(y) q_-(x) \mathbb{1}_C(x) \pi(x) \rho_{\Delta t}(dx, dy) - \int \mathbb{1}_C(y) q_+(y) q_-(x) \mathbb{1}_{C^c}(x) \pi(x) \rho_{\Delta t}(dx, dy) \tag{3.82}$$

This is the discrete-time equivalent of equation (30) in reference [48]. Applying the definition of the generator and observing that that $\mathbb{1}_C(x)\mathbb{1}_{C^c}(x) = 0$ everywhere gives (3.28). We then arrive at (3.29) in our work by the same arguments as in reference [159].

3.9.4 Grid-Based Reference Scheme

Here we discuss the scheme used to calculate the reference values for our test system in Sections 3.5 and 3.6. Instead of considering the discrete time process directly, we will attempt to approximate the dynamics of the continuous-time Brownian dynamics on the test potential. To this end, we define a Markov hopping process on a grid that converges to the continuous time dynamics as the grid becomes finer. Specifically, we allow nearest neighbor hops on a

square grid with spacing ϵ. The hopping probabilities are given by

$$\begin{aligned}
P(x+\epsilon,y) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x+\epsilon,y)-U(x,y)\right]}\right) \\
P(x-\epsilon,y) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x-\epsilon,y)-U(x,y)\right]}\right) \\
P(x,y+\epsilon) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x,y+\epsilon)-U(x,y)\right]}\right) \\
P(x,y-\epsilon) &= \left(\frac{1}{4}\right)\left(\frac{1}{1+\exp\left[U(x,y-\epsilon)-U(x,y)\right]}\right) \\
P(x,y) =& 1 - P(x+\epsilon,y) - P(x-\epsilon,y) \\
& - P(x,y+\epsilon) - P(x,y-\varepsilon).
\end{aligned} \tag{3.83}$$

Here $P(x\pm\epsilon,y)$ is the probability of hopping one grid point to the right or left, $P(x,y\pm\epsilon)$ is the probability of hopping up or down the grid, and $P(x,y)$ is the probability of remaining in place.

We will not give a full proof of convergence. Instead we merely demonstrate that as $\epsilon \to 0$, we approximate the infinitesimal generator $\mathcal{L}^{\text{brwn}}$ for Brownian Dynamics. Let P be the transition matrix associated with the transition probabilities given by (3.83), f be a three-times continuously differentiable function, and the vector $\vec{f}$ the values of f evaluated at each grid point. In the limit of $\epsilon \to 0$,

$$\frac{16(P-I)\vec{f}}{\epsilon^2}(x,y) = \mathcal{L}^{\text{brwn}} f(x,y) + \mathcal{O}(\epsilon) \tag{3.84}$$

where $\mathcal{L}^{\text{brwn}}$ is the infinitesimal generator for Brownian dynamics with isotropic diffusion constant,

$$\mathcal{L}^{\text{brwn}} f(x,y) = -\partial_x U(x,y)\partial_x f(x,y) - \partial_y U(x,y)\partial_y f(x,y) + \partial_x^2 f(x,y) + \partial_y^2 f(x,y). \tag{3.85}$$

To demonstrate this, we write $(P-I)\vec{f}$ explicitly as

$$\begin{aligned}(P-I)f(x,y) =& P(x+\epsilon,y)f(x+\epsilon,y) + P(x-\epsilon,y)f(x-\epsilon,y)\\ &+ P(x,y+\epsilon)f(x,y+\epsilon) + P(x,y-\epsilon)f(x,y-\epsilon)\\ &+ P(x,y)f(x,y) + \mathcal{O}(\epsilon^3)\end{aligned}$$

If we expand f to second order around (x,y), the zeroth order term cancels, leaving

$$\begin{aligned}(P-I)f(x,y) =& P(x+\epsilon,y)\left(\epsilon\partial_x f + \frac{1}{2}\epsilon^2\partial_x^2 f\right) - P(x-\epsilon,y)\left(\epsilon\partial_x f - \frac{1}{2}\epsilon^2\partial_x^2 f\right)\\ &+ P(x,y+\epsilon r)\left(\epsilon\partial_y f + \frac{1}{2}(\epsilon r)^2\partial_y^2 f\right) - P(x,y-\epsilon r)\left(\epsilon\partial_y f - \frac{1}{2}(\epsilon r)^2\partial_y^2 f\right)\\ &+ \mathcal{O}(\epsilon^3)\end{aligned}$$

We then expand the transition probabilities to first order, giving

$$\begin{aligned}P(x\pm\epsilon,y) &= \frac{1}{8}\left(1 \mp \frac{1}{2}\partial_x U(x,y)\epsilon\right) + \mathcal{O}(\epsilon^2)\\ P(x,y\pm\epsilon r) &= \frac{1}{8}\left(1 \mp \frac{1}{2}\partial_x U(x,y)\epsilon\right) + \mathcal{O}(\epsilon^2).\end{aligned}$$

Substituting, simplifying, and multiplying by $16/\epsilon^2$ gives (3.84).

To estimate the reference quantities for our test system, we constructed a square grid on the interval $-2.5 \le x \le 1.5$ and $-1.5 \le y \le 2.5$ with grid spacing of 0.005. We then construct the transition rate matrix $16(P-I)/\epsilon^2$, and estimate the dynamical quantities using the corresponding formulas in Section 3.3.

3.9.5 Basis Size Choice for the Müller-Brown model

In Figure 3.6, we show the dependence of the root-mean-square error in the committor on basis size for the Müller-Brown model. While using 1000 basis functions gives a slightly better result at higher dimensions, it is not enough to appreciably change the trends depicted in Figure 3.2. However, choosing 1000 or more basis functions gives worse results for the two-dimensional system. We therefore chose to use 500 dimensions to avoid giving the impression that the diffusion-map basis outperforms the MSM basis at low dimensions.

3.9.6 Numerical Effect of Enforcing Detailed Balance

To test the effect of enforcing detailed balance in MSMs through a maximum likelihood procedure, we returned to our two-dimensional test potential without any additional nuisance degrees of freedom. Using the clusterings described in Section 3.5.1, we constructed MSMs in PyEMMA both with and without the reversible option set to True. We then estimated the mean first-passage time from state B into state A, using the states depicted in Figure 3.1A. As before, we repeated this procedure over thirty replicates. Moreover, we also varied the number of short trajectories included in the dataset to observe trends in statistical convergence.

Our results are given in Figure 3.7. The mean first-passage time calculated using reversible MSMs, depicted in panel A, grows unboundedly with increasing basis size. To demonstrate that this not due to the nature of the data, we repeated the calculation on a long equilibrium trajectory of commensurate length. Our results, shown in panel B, exhibit the same phenomenon. We also varied the error tolerance for convergence, as well as the minimum count required for connectivity. Neither affected the results. Moreover, an in-house code for the iteration described in reference [82] gave the same results as PyEMMA. Rather, we see that the bias decays with increasing dataset sizes, suggesting that it is statistical in nature.

In panels C and D, we show estimates constructed without enforcing reversibility, which we term the naive estimator. The naive estimator does not have the same bias. This suggests that the maximum likelihood iteration introduces a large, slowly decaying statistical error.

To ensure that this is not an artifact of the clustering procedure, we also constructed MSMs by applying k-means globally to the data, without regard to boundary conditions. We then estimated the dominant implied timescale for both clustering schemes, which we plot in Figure 3.8. We see the same trends as in the mean first-passage time: for reversible MSMs, the implied timescale grows unboundedly with the number of basis functions for both clustering methods. In contrast, both clustering methods converge equally well when using the naive estimator.

3.9.7 Supplementary Plots for Delay Embedding on the Müller-Brown model

In Figure 3.9, we give implied timescales for the MSMs constructed in Section 3.6. To test the effect of trajectory length on the one-dimensional, delay-embedded data, we repeated the calculation for three additional datasets. The total number of points in each dataset is fixed, but each nonequilibrium trajectory is of different length. We plot the resulting curves in Figure 3.10. In all cases, we see an anomalous behavior when the delay length or lag time approaches the total length of the trajectory.

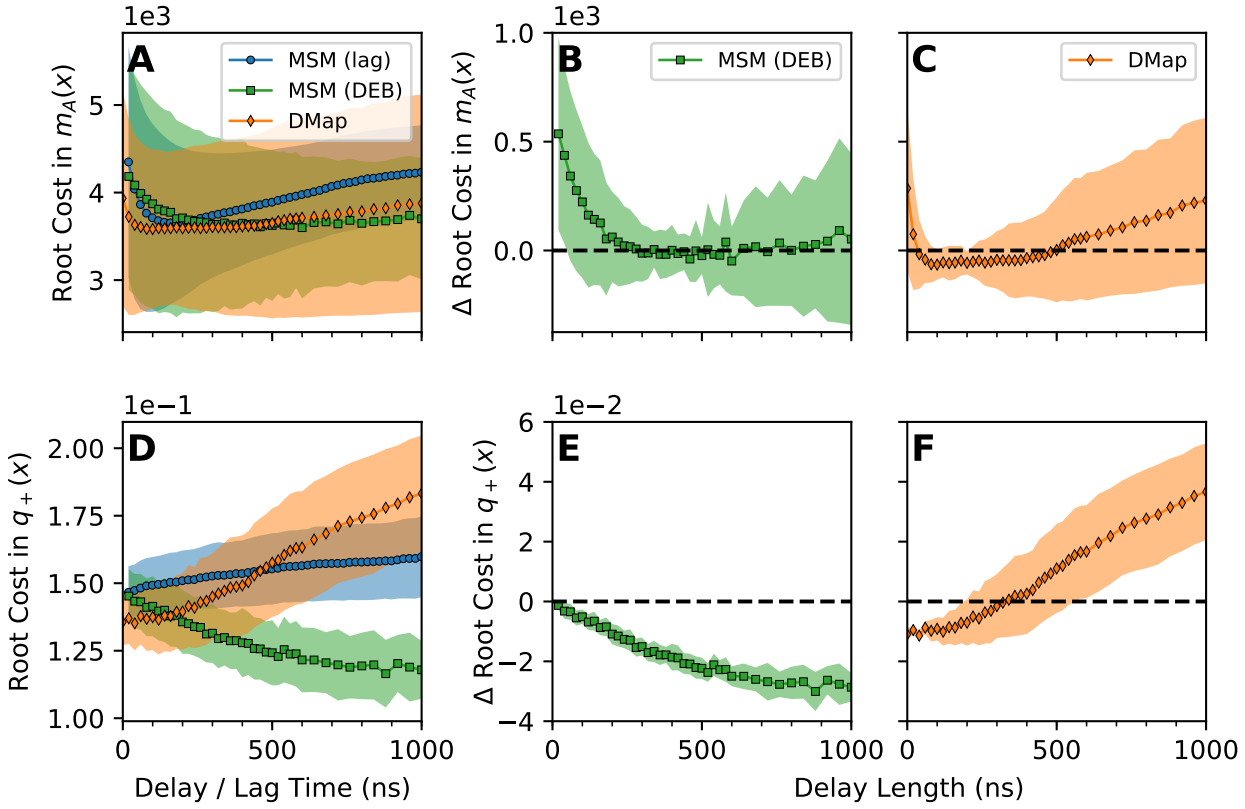


Figure 3.5: Results from a DGA calculation on a dataset of six long folding and unfolding trajectories of the Fip35 WW domain. (A,D) The root cost in the mean first-passage time and forward committor respectively, calculated using an MSM basis with increasing lag time, an MSM basis with delay embedding, and diffusion map basis with delay embedding, averaged over all test/train splits. (B,C,E,F) Difference in root cost relative to the best parameter choice for the estimate constructed using the MSM basis with increasing lag time. Negative values are better. (B) Difference in cost for the mean first-passage time estimated with an MSM basis with delay embedding. (C) The same as in (B) but with the diffusion map basis instead. (E) Difference in cost for the committor estimated with an MSM basis with delay embedding. (F) The same as in (E) but with the diffusion map basis instead. In all plots the symbols are the average over test/train splits, and the shading indicates the standard deviation across test/train splits.

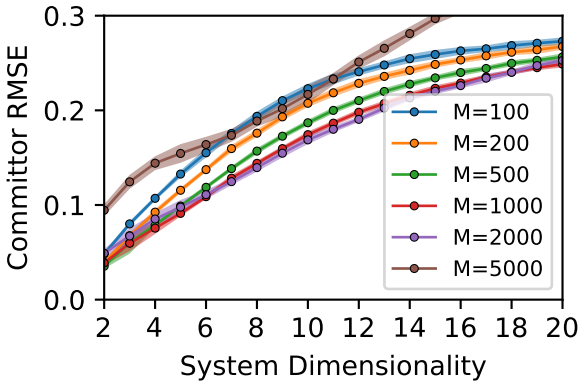


Figure 3.6: Dependence of the MSM committor root-mean-square error (RMSE) on the number of clusters. Different curves correspond to different numbers of Markov states.

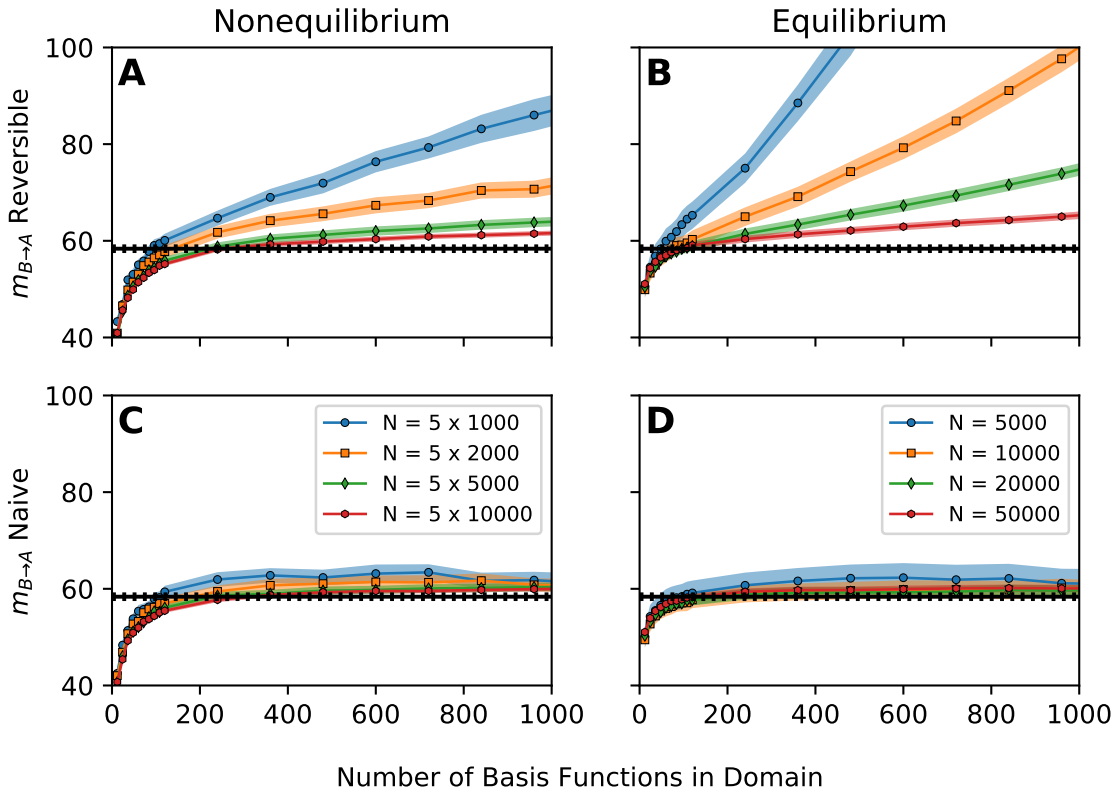


Figure 3.7: Effect of enforcing MSM reversibility on the estimated mean first-passage time from state B to state A on the scaled Müller-Brown potential. Estimates in the top row are constructed using the reversible MSM estimator and estimates in the bottom row are not. The columns correspond to two different datasets: the left column shows estimates constructed from the nonequilibrium dataset detailed in section 3.5, and the right column shows estimates constructed from a long equilibrium trajectory. Different curves correspond to MSMs constructed from datasets of different sizes.

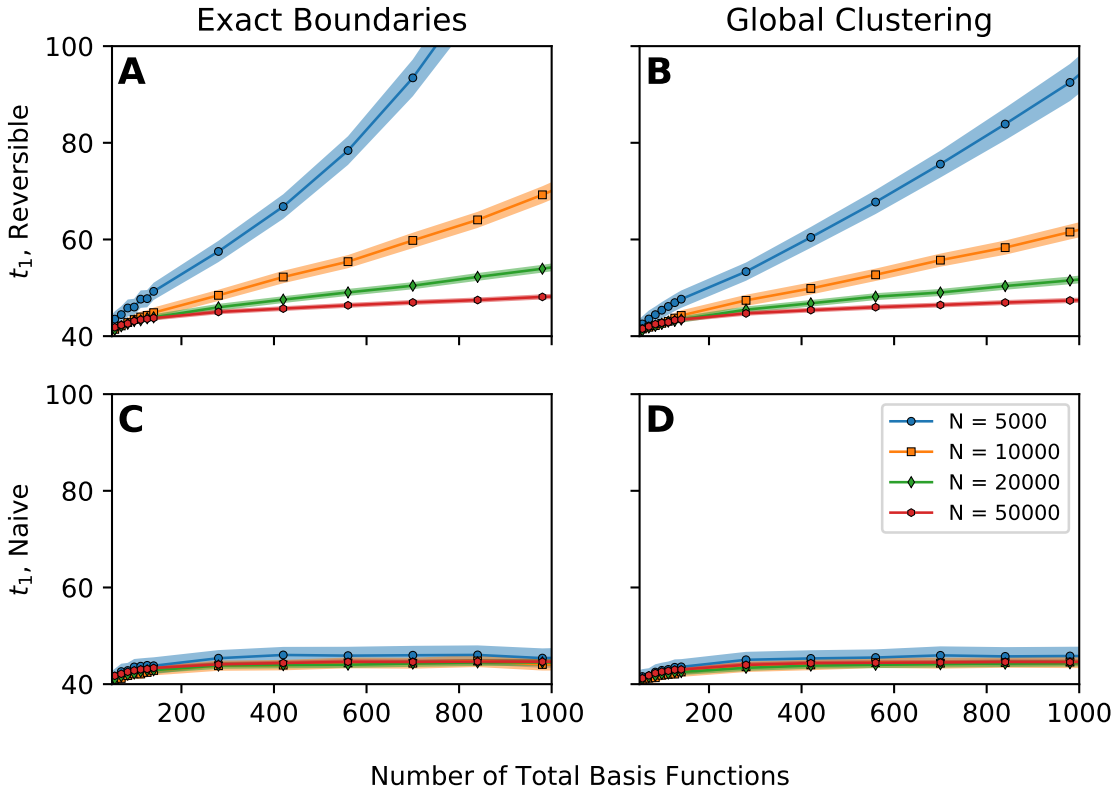


Figure 3.8: Dominant implied timescale for MSMs constructed on a long equilibrium trajectory on the scaled Müller-Brown potential. Estimates in the top row are constructed using the reversible MSM estimator and estimates in the bottom row use the naive estimator. Columns correspond two different clustering schemes. The left column gives estimates constructed using the clustering described in Section 3.5.1, and the right column gives estimates obtained by clustering the data without regard for the boundary conditions (i.e., globally). Different curves correspond to MSMs constructed on different size datasets.

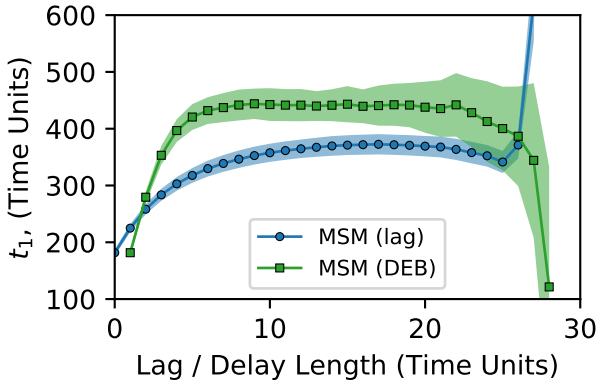


Figure 3.9: Implied timescales for the delay-embedded MSM and lagged MSMs in Section 3.6.

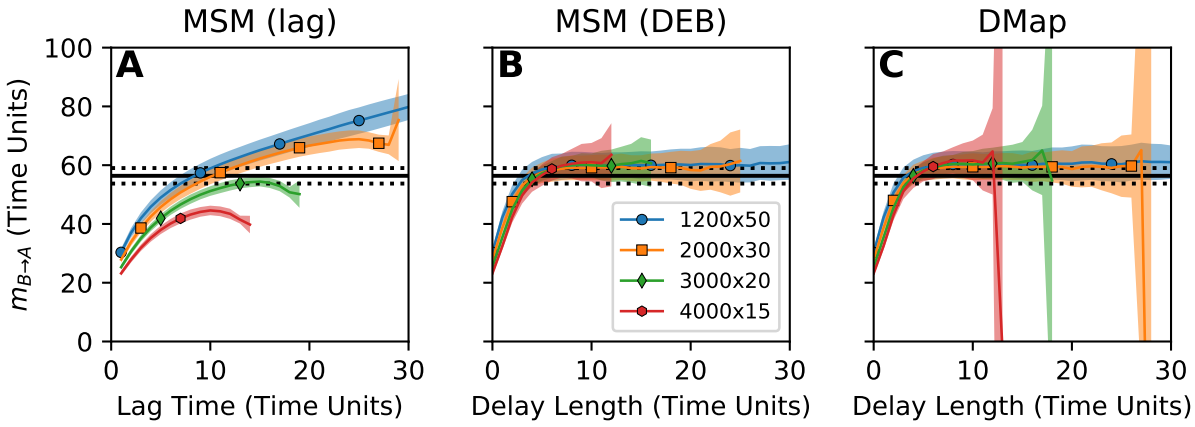


Figure 3.10: Comparison of methods for controlling the projection error in an incomplete CV space. Plots are as in Figure 3.4, with the addition of three new datasets. The curves correspond to datasets consisting of 1200 trajectories, each 50 time units long (blue circles), 2000 points, each 30 time units long (orange squares, the same data as pictured in Figure 3.4), 3000 trajectories, each 20 units long (green diamonds), and 4000 trajectories, each 15 units long (red hexagons).

CHAPTER 4

ERROR ANALYSIS FOR THE VARIATIONAL APPROACH TO CONFORMATIONAL DYNAMICS

4.1 Introduction

Molecular dynamics simulations allow the motions of chemical systems to be studied at atomistic detail. However, their complexity can belie easy interpretation. To aid in interpretation, is often useful to construct reduced descriptions of the dynamics. One approach to constructing these descriptions is to exploit the fact that many chemical systems exhibit a strong separation of timescales. While fast processes such as atomic vibrations or water rearrangements relax on timescales of picoseconds or faster, protein conformational changes typically occur on timescales of microseconds or longer. Extracting these slow processes from simulated data gives a simpler, more interpretable description of the dynamics.

Considerable work has focused on data-driven techniques that statistically extract these degrees of freedom by diagonalizing matrices whose entries are of time correlations of a set of basis functions. Theoretically, these schemes approximate the spectral properties of an operator associated with the system's dynamics known as the *transition operator*. The transition operator completely defines the dynamics on a distributional level, and the eigenfunctions with the largest eigenvalues govern the system's long-time dynamics. More precisely, the subspace spanned by these eigenfunctions contains the functions that decay most slowly to their equilibrium averages.

Such schemes have been proposed independently in different fields. Presumably because the problems of interest have been different, different sets of basis functions have been used [105, 106, 107, 70, 71, 160, 78, 109, 110, 111]. Initial approaches such as Time-lagged Independent Component Analysis (TICA) and Relaxation Mode Analysis (RMA) used linear functions on the system's coordinates as basis functions [105, 106, 107], although it was noted early on one could employ nonlinear functions as well [107]. This was soon followed by

Markov State Models, which use a basis set of indicator functions on disjoint sets [70, 71, 161]. Subsequent work connected these techniques under the name of the Variational Approach to Conformational Dynamics (VAC) [78, 162, 88], and introduced new sets of basis functions [162, 113, 112]. Moreover, the existence of variational principles associated with the dynamics [107, 78, 162, 99] allowed basis sets to be optimized variationally. Not only does this allow basis sets to be constructed using complex, nonlinear models such as neural networks [115, 98, 163], these variational principles provide a systematic way of comparing families of basis sets.

In this work, we study how the interplay of basis set choice, lag time, and sampling error affects the accuracy of VAC schemes. Our theoretical and numerical analyses suggest that the best choice of lag time depends strongly on both the size of the slow subspace one wishes to capture and the amount of data available. For this reason, we introduce new heuristics for choosing the lag time informed by our theoretical results.

4.2 The Dynamical Operators and VAC Theory

Dynamical operators play a key role in the theory underlying VAC algorithms. Here, we review these operators and show that the subspace spanned by their eigenfunctions give the long-time behavior of the system. We then introduce VAC theory, and show how VAC attempts to approximate this subspace.

4.2.1 Dynamical Operators

To begin our analysis, we assume that the dynamics of the system are given by a continuous-time Markov process X_t in a phase space Ω. The transition operator at t is defined as

$$\mathcal{K}_t f(x) = \mathbf{E}\left[f\left(X_t\right) | X_0 = x\right], \tag{4.1}$$

where f is a function of Ω. The transition operator is also called the Markov or (stochastic) Koopman operator [80, 4]. We use the term transition operator as it is well-established in the mathematical literature, and emphasizes the connection with the transition matrix for a finite-state Markov chain: for a finite-state Markov chain, f is a vector and K_1 is a row-stochastic transition matrix. Similarly, the condition that the rows of a transition matrix sum to one can be written more generally as

$$\mathcal{K}_t 1 = \mathbf{E}\left[1 | X_0 = x\right] = 1. \tag{4.2}$$

Just as the transition operator generalizes the transition matrix for a given lag time, the operator

$$\mathcal{L}f(x) = \lim_{s \to 0} \frac{K_s f(x) - f(x)}{s}, \tag{4.3}$$

generalizes the rate matrix. This operator is known as the *infinitesimal generator*, hereafter we refer to $\mathcal{L}$ as simply the generator for brevity. Equation 4.2 immediately implies $\mathcal{L}1 = 0$, which generalizes the requirement that the rows of a transition rate matrix sum to zero. Moreover, just as the transition matrix is the matrix exponential of the rate matrix, the transition operator can be written as

$$\mathcal{K}_t f(x) = e^{\mathcal{L}t}. \tag{4.4}$$

where the exponential of an operator is defined analogously to the matrix exponential,

$$e^{\mathcal{L}t} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathcal{L}^k \tag{4.5}$$

The infinitesimal generator for many Markov processes can be written down in closed form. For instance, for a stochastic differential equation of the form

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t \tag{4.6}$$

where W_t is a Wiener process and where b and σ are vector and matrix valued functions respectively, the generator is given by the differential operator

$$\mathcal{L}f(x) = \sum_i b_i(X_t) \cdot \nabla f(x) + \frac{1}{2} \operatorname{Tr}\left(\sigma(x)\sigma(x)^T H_f\right) \tag{4.7}$$

where H_f is the Hessian of f.

The eigenfunctions η_i of $\mathcal{L}$ are defined as the functions obeying

$$\mathcal{L}\eta_i(x) = \lambda_i \eta_i(x). \tag{4.8}$$

for some eigenvalue λ_i. Equation (4.2) shows that the constant function is an eigenfunction of $\mathcal{L}$ with eigenvalue 0. However, further analyzing the other eigenfunctions and eigenvalues requires additional assumptions. We first assume that the dynamics obey detailed balance with respect to a unique stationary probability measure μ. This is defined as the probability measure such that

$$\mathbf{E}_\mu \left[\mathcal{K}_t f\right] = \mathbf{E}_\mu \left[f\right] \tag{4.9}$$

for all t and f, or equivalently that

$$\mathbf{E}_\mu \left[\mathcal{L} f\right] = 0. \tag{4.10}$$

If X_t is stationary at thermal equilibrium and μ has a density against the Lebesgue measure, we can write

$$\mathbf{E}_\mu \left[f(x)\right] = \frac{\int f(x) e^{-H(x)/k_B T} dx}{\int e^{-H(x)/k_B T} dx}$$

Note that we can conveniently express time correlation functions using expectations against μ and the transition operator,

$$\operatorname{corr}\left[f(X_0), g(X_t)\right] = \mathbf{E}_\mu \left[f(x) \left(\mathcal{K}_t g\right)(x)\right]. \tag{4.11}$$

The existence of μ allows us to define the inner product

$$\langle f, g \rangle = \mathbf{E}_\mu \left[\bar{f}(x) g(x) \right] \tag{4.12}$$

Here $\bar{f}$ is the complex conjugate of f.

Associated with this inner product is the Hilbert space of functions that are square-integrable against μ, which we denote as $L^2(\Omega, \mu)$. We will also assume that the $\mathcal{L}$ is an essentially self adjoint operator in this Hilbert space; this condition is also known as *detailed balance*. This implies that its eigenfunctions form an orthonormal basis of $L^2(\Omega, \mu)$. Moreover, the eigenvalues are all real and can be ordered such that

$$\lambda_0 > \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \to -\infty \text{ and } 0 = \lambda_0 \tag{4.13}$$

Indeed, the fact that the eigenvalues must be real can be seen from the fact that self-adjointness implies

$$\langle f, \mathcal{L} g \rangle = \langle \mathcal{L} f, g \rangle \text{ and } \langle f, \mathcal{K}_t g \rangle = \langle \mathcal{K}_t f, g \rangle \tag{4.14}$$

and setting both g and f to η_i.

4.2.2 Slow subspaces for Markov Chains

Equation (4.4) and the definition of the operator exponential imply

$$\mathcal{K}_t \eta_i(x) = e^{\lambda_i t} \eta_i(x) \tag{4.15}$$

holds for all t. Since we have assumed that the eigenfunctions form an orthonormal basis, the action of $\mathcal{K}_t$ can be written as

$$\mathcal{K}_t f(x) = \sum_{i=0}^{\infty} e^{\lambda_i t} \eta_i(x) \langle \eta_i, f \rangle \tag{4.16}$$

As the eigenvalues are all negative, each term in the sum decays exponentially. Moreover, (4.13) suggests that at long lag times the action of $\mathcal{K}_t$ is dominated by its behavior on the subspace spanned the few slowest eigenfunctions.

This can be formalized by considering the behavior of correlation functions in the long time limit. Let S_K denote the linear subspace of $L^2(\Omega, \mu)$ spanned by the top K eigenfunctions $\eta_1, \eta_2, \ldots, \eta_K$. For any function $f \in S_K$, the correlation functions of f decay slowly.

$$e^{\lambda_K t} \leq \text{corr}\left[f(X_0), f(X_t)\right] \leq e^{\lambda_1 t}$$

This follows directly from (4.16) and (4.11). In contrast, a function that is orthogonal to S_K has correlation functions that decay relatively quickly.

$$0 \leq \text{corr}\left[g(X_0), g(X_t)\right] \leq e^{\lambda_{K+1} t}$$

4.2.3 Variational Approach to Conformational Dynamics

In practice, $\mathcal{L}$ and $\mathcal{K}_t$ are generally too complicated to allow the eigenfunctions S_K, to be determined directly. We will therefore approximate S_K using a set of basis functions $\{\phi_i\}$. Specifically, let F be the subspace spanned by our basis functions. We will attempt to find S_K^F, the subspace of F spanned by the orthogonal projections $\eta_1 \ldots \eta_K$ onto F. Note that the linearity of projection implies that the orthogonal projection of any element of S_K is contained in S_K^F. For simplicity, we will assume that none of the eigenfunctions $\eta_1, \ldots, \eta_K$ are orthogonal to F.

If the operator

$$\sum_{k=1}^{K} e^{\lambda_k t} \eta_k(x) \langle \eta_k, f \rangle \tag{4.17}$$

was known for some $t > 0$, it would be possible to find the coefficients for the functions spanning S_K^F.

Theorem 4.2.1. *Let $F \subseteq L^2(\Omega, \mu)$ be an m-dimensional linear subspace spanned by a set of*

basis functions $\phi_1, \phi_2, \ldots, \phi_m$ *with* $\mathbf{E}_\mu [\phi_i (X_0)] = 0$ *for* $1 \leq i \leq m$*. Then, for each* $K < m$*, there exists a* K*-dimensional linear subspace* $S_K^F \subseteq F$*, which contains the projection of all the eigenfunctions* $\eta_1, \ldots, \eta_K$ *onto* F*. The subspace* S_K^F *can then be found exactly by solving the generalized eigenvalue problem*

$$\omega C(0)\beta = \sum_{k=1}^{K} e^{\lambda_k t} \left\langle \eta_k, \vec{\phi} \right\rangle \left\langle \eta_k, \vec{\phi} \right\rangle^T \beta \tag{4.18}$$

where $\left\langle \eta_k, \vec{\phi} \right\rangle$ *is the vector with elements* $\langle \eta_k, \phi_i \rangle$ *for* $1 \leq i \leq m$*, and* $t > 0$ *is an arbitrary lag time. Let* $\beta^1, \beta^2, \ldots, \beta^m$ *denote the generalized eigenvectors associated with eigenvalues* $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_m \geq 0$*. Then the linear subspace spanned by the basis elements*

$$\sum_{i=1}^{m} \beta_i^k \phi_i(x), \qquad 1 \leq k \leq K$$

is S_K^F*.*

For a proof, we refer to Reference [164].

Unfortunately, (4.18) can also not be evaluated in practice, as we have no way of evaluating the required sum unless the eigenfunctions are known explicitly. Here, we will consider VAC algorithms as a strategy for solving for S_K^F. Instead of solving (4.18) directly, (linear) VAC seeks to solve the eigenproblem

$$\gamma C(0)\vec{a} = C(t)\vec{a} \tag{4.19}$$

where $C(t)$ is the matrix

$$C_{ij}(t) = \left\langle \phi_i, \mathcal{K}_t \phi_j \right\rangle = \operatorname{corr}\left[\phi_i(X_0), \phi_j(X_t)\right] \tag{4.20}$$

$$= \sum_{k=1}^{\infty} e^{\lambda_k t} \langle \phi_i, \eta_k \rangle \left\langle \eta_k, \phi_j \right\rangle^T. \tag{4.21}$$

The correlation functions required in (4.20) must generally be estimated by taking sample means. Specifically, let x_n be a collection of points drawn from μ, and let y_n be the corresponding points collected by propagating the dynamics forwards for time t. We can then estimate (4.20) as

$$\hat{C}_{ij}(t) = \frac{1}{N} \sum_{n=1}^{N} \phi_i(x_n)\phi_j(y_n). \tag{4.22}$$

Substituting these equations into (4.19) allows us to compute approximate coefficients $\hat{a}$ and eigenvalues $\hat{\gamma}_i$. Note that in practice, the initial points x_n do not have to be drawn independently from μ. One could collect the data by choosing pairs of points separated by t from one or more longer equilibrium trajectories. Moreover, one could relax our assumptions that the points are drawn from μ, and instead draw them from a different probability distribution. Averages against μ could then be evaluated by reweighting with the appropriate change of measure.

While detailed balance implies that the true $C(t)$ matrix is always symmetric, with finite data $\hat{C}(t)$ may not be symmetric due to sampling error. For this reason, one commonly symmetrizes $\hat{C}(t)$ by averaging it with its transpose and solving the problem

$$\frac{1}{2}\left(\hat{C}(t) + \hat{C}(t)^T\right) a = \gamma \hat{C}(0) a \tag{4.23}$$

This ensures that the calculated eigenvalues are always real. Hereafter, we will assume that this symmetrization has been applied.

If the basis functions are close to the true eigenfunctions, we expect (4.21) to be dominated by the terms where $k \leq K$. Consequently, a should then approach β. Alternatively, for a more general basis, the space spanned by the functions $\sum_{i=1}^{m} a_i^k \phi_i$ approaches S_K^F as t goes to infinity. Both of these statements are proven in Reference [164]. However, if neither of these statements hold, there may be a substantial difference between the a and w. This difference may be further exacerbated by statistical errors in the sample means.

VAC schemes differ primarily on the basis set used. In TICA, one uses a basis set of

functions that are linear on key degrees of freedom. The resulting approximations are often used as collective variable spaces, either for insight or for further sampling. MSMs divide the system's phase space into a finite number of disjoint sets, and defines each basis function to be one if the system is in a set and zero otherwise. In practice, these schemes are commonly used in succession. In this approach one first applies TICA to extract a lower-dimensional subspace, and then applies clustering algorithms in this subspace to define sets for the MSM. This has the added advantage that the matrix $S^{-1}K$ is a stochastic matrix, which can be used to estimate other quantities such as mean-first-passage times and committors [75, 76]. Other approaches have used basis functions constructed using tensor products of simpler basis functions, machine-learning approaches, or an implicit basis defined by a kernel [113, 112, 89]. Moreover, it has been noted that a variational principle for the eigenvalues of $\mathcal{K}_t$ can be formulated [107, 78]. This allows one to compare different sets of basis functions, as variational theory suggests that (if the averages in (4.22) have converged), larger values of γ_i correspond to more accurate representations of η_i [162]. This has also led to schemes that attempt to approximate slow subspaces by optimizing over nonlinear functions spaces, such as the outputs of a neural network [98, 163]. Here, we will only consider VAC schemes on linear subspaces. For cases where data was not collected from μ and reweighting is not feasible, or where the dynamics do not obey detailed balance, a variational principle based on the singular value decomposition has also been proposed [99]. We will leave the analysis of these schemes for future work as well.

4.2.4 Previous Theoretical Analysis of VAC

Previous work has focused on aspects of the error in VAC schemes. Work in Reference [165] focused on the effect of sampling error for a given MSM basis and choice of the lag time t, and derived expressions for the resulting distribution of the eigenvalues and eigenvectors of an MSM. In references [73, 142], the error of VAC schemes was analyzed in the limit of infinite data. In particular, upper bounds were constructed on the absolute error on the estimates

of $e^{\lambda_i t}$. These upper bounds go to zero as the lag time goes to infinity independently of the choice of basis. This establishes the consistency of VAC schemes: as the lag time increases, both $e^{\lambda_i t}$ and γ_i go to zero, as expected.

It was noted early on that the efficacy of VAC approaches may depend strongly on the choice of the lag time for $\mathcal{K}_t$ [71, 161, 166]. Choosing longer lag times causes eigenfunctions with more negative eigenvalues to contribute less to the action of $\mathcal{K}_t$, allowing (4.20) to more easily capture the contributions from the slower eigenfunctions. However, if the lag time is too long, the matrix elements in $C(t)$ may decay towards zero causing the calculation to be increasingly sensitive to sampling error. Moreover, the variational principles used to compare basis sets only hold for a given lag time, and variational principles cannot be used to compare schemes across lag times [167]. One common approach for choosing the lag time is through implied timescale analysis. The i'th implied timescale is defined as

$$ITS_i = -\frac{1}{\lambda_i} \tag{4.24}$$

and can be approximated as

$$ITS_i \approx \frac{-t}{\log \gamma_i}. \tag{4.25}$$

If the basis captured the eigenfunctions perfectly, the estimated timescales would be constant as a function of time. Consequently, one attempts to choose the lag time by finding a region where (4.25) is approximately flat.

In this work, we analyze the interplay between sampling error, basis set projection error, and the choice of lag time for VAC calculations. Our theoretical and numerical results allow us to assess the efficacy of various VAC schemes, as well as various strategies for hyperparameter selection. They also suggest new heuristics for choosing lag times for VAC schemes.

4.2.5 TICA and VAC at Short Times

To motivate our analysis, we consider the behavior of a specific VAC scheme, TICA, applied to Brownian motion at a small lag time. For now, we consider the limit of infinite data, and assume all expectations can be evaluated exactly. This highlights the importance of lag time selection, even when basis functions may be relatively well aligned with the dominant eigenfunctions.

Using (4.4), and the fact that the dynamics obey detailed balance, we can rewrite (4.20) as

$$\begin{aligned} C_{ij}(t) &= \left\langle \phi_i, e^{\mathcal{L}t}\phi_j \right\rangle \\ &= \langle \phi_k, \phi_l \rangle + t \left\langle \phi_i, \mathcal{L}\phi_j \right\rangle + \mathbf{O}\left(t^2\right) \end{aligned} \tag{4.26}$$

For a Brownian motion, a simple computation using (4.7) shows that

$$\mathcal{L}(fg)(x) - f(x)\mathcal{L}g(x) - g(x)\mathcal{L}f(x) \tag{4.27}$$

$$= \nabla f(x)^T \sigma(x)\sigma(x)^T \nabla g(x) \tag{4.28}$$

For more general Markov processes, the left-hand-side is known as the *carré du champ*. Averaging against μ and applying (4.10) and (4.14) gives

$$\begin{aligned} &\mathbf{E}_\mu \left[\nabla f(x)^T \sigma(x)\sigma(x)^T \nabla g(x) b \right] \\ &= -2\mathbf{E}_\mu \left[f(x)\mathcal{L}g(x) \right] \end{aligned} \tag{4.29}$$

Moreover, since the basis functions are simply the coordinate functions $\phi_i(x) = x_i$, we have

$$\nabla\phi_i(x)^T \sigma(x)\sigma(x)^T \nabla\phi_j(x) = \left(\sigma(x)\sigma(x)^T\right)_{ij}. \tag{4.30}$$

Combining these results shows that applying TICA with a short lag time to a Brownian

motion is equivalent to solving the eigenproblem

$$\mathbf{E}\left[\left(\sigma(x)\sigma(x)^T\right)\right]a = (1-2\gamma)C(0)a + \mathbf{O}\left(t^2\right). \tag{4.31}$$

Up to the $\mathbf{O}\left(t^2\right)$ term, this is the generalized eigenproblem associated with the average diffusion coefficient. In general, this should not be expected to contain any information about S_K.

To demonstrate this numerically, we constructed a two-dimensional double well potential of the form

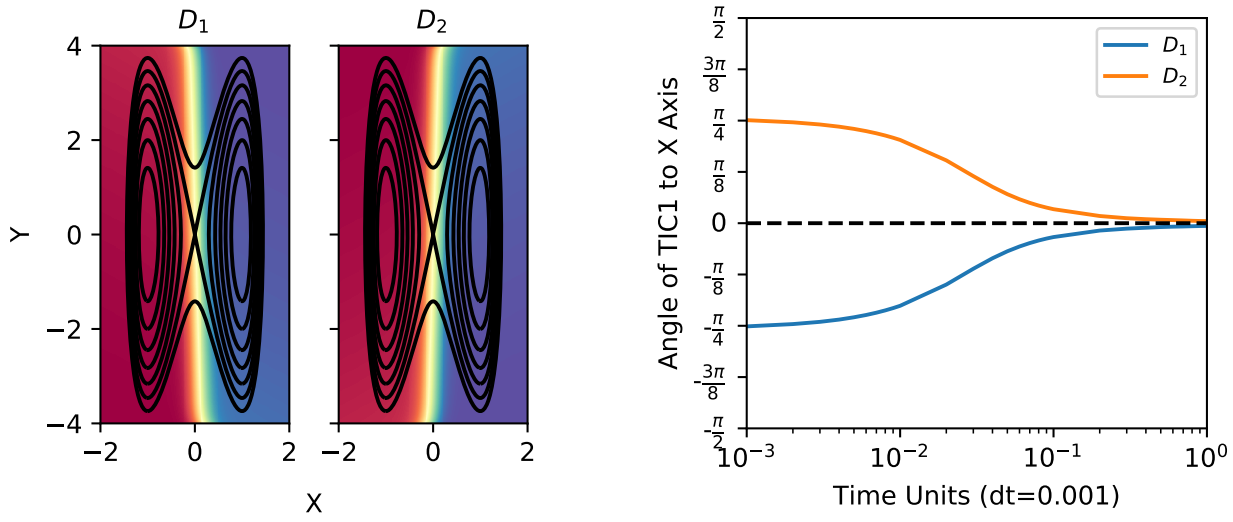
$$U(x,y) = 6(x^4 - 2x^2) + 0.5y^2. \tag{4.32}$$

This was chosen so that the variance of both x and y against π was approximately 1. We then considered a Brownian process on the potential with constant diffusion coefficients

$$D_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}. \tag{4.33}$$

We note that the dominant eigenvector of D_1 is $[1,1]$, and the dominant eigenvector of D_2 is $[1,-1]$, which point at angles of 45° and -45° from the x axis, respectively. In contrast, the dominant eigenfunction of $\mathcal{K}_t$ varies almost exclusively along the x-axis, as seen in Figure 4.1(a). Consequently, if TICA gave the coordinate that points along the slowest coordinate, we would expect the dominant TICA eigenfunction to point along the x-axis.

We simulated the Brownian process for three million steps at a timestep of 0.001, and applied TICA to the x and y coordinates. We then calculated the angle of the dominant TICA vector from the x-axis as a function of the time lag. The results are given in Figure 4.1(b) For sufficiently long lag times, we see that the TICA eigenvector does indeed point in the direction of the slowest mode of the system. However, for shorter times TICA gives a value that is almost completely dominated by the diffusion coefficient. For short lag times, the estimated TIC points almost directly towards the dominant eigenvector of D. Indeed, we



(a) (b)

Figure 4.1: Figure 4.1(a) displays the eigenfunctions of the transition operator for two diffusion coefficients. In each plot the heatmap gives the value of the subdominant eigenfunction of the transition operator. The contours give the value of the potential energy, and are spaced at values of $k_B T$. Figure 4.1(b) gives the angle of the dominant TICA vector against the x axis as a function of TICA lag time.

see that the diffusion coefficient seems to have a pronounced effect on the dynamics up to a lag time of around 100 timesteps (0.1 time units). For comparison, we observe that the average time to transition between wells is approximately 6000 timesteps(6 time units).

4.3 Perturbation of Analysis VAC

We now derive our main theoretical result, a perturbation analysis describing the error of VAC. To describe the error in perturbing a generalized eigenvalue problem, we first give the following theorem.

Theorem 4.3.1. *Consider the eigenproblem*

$$\nu I x = A x$$

where A is a rank-K matrix with top K eigenvalues $\nu_1 \geq \nu_2 \geq \cdots \geq \nu_K > 0$. Let S be the

linear subspace associated with the largest K eigenvalues. Consider also a perturbed problem with error terms E_1 and E_2:

$$\nu' (I + E_1) x' = (A + E_2) x' \tag{4.34}$$

Let $\hat{S}$ denote the subspace associated with the top K eigenvalues for this perturbed problem. The sine of the largest canonical angle between S and $\hat{S}$ can be then bounded by

$$\sin(\theta_1) \leq \|E_1\|_2 + \frac{\|E_2\|_2}{\nu_K} + \mathcal{O}\left(\|E_1\|_2 + \|E_2\|_2\right)^2 \tag{4.35}$$

A proof is given in Reference [164].

To apply this theorem to VAC, we first rewrite the scheme in terms of the "whitened" basis functions

$$g_i(x) = \sum_{j=1}^{m} C_{ij}^{-\frac{1}{2}}(0)\,\phi_j(x), \qquad 1 \leq i \leq m$$

These new basis functions also have F as their span, and the associated VAC eigenproblem is given by

$$\frac{1}{2}\left(\hat{D}(t) + \hat{D}(t)^T\right) v(i) = \gamma_i \hat{D}(0) v(i) \tag{4.36}$$

where the new terms are related to the terms in (4.23) as follows.

$$v = C^{\frac{1}{2}}(0)\, a$$
$$\hat{D}(t) = C^{-\frac{1}{2}}(0)\, \hat{C}(t)\, C^{-\frac{1}{2}}(0)$$

The eigenvalues remain unchanged through this redefinition.

At this point we are ready to apply a perturbation analysis. The covariance of the features g_i is the identity matrix, so the matrix $\hat{D}(0)$ is a perturbation of the identity matrix. Similarly, we can consider $\hat{D}(t)$ to be a perturbation of $\sum_{k=1}^{K} e^{\lambda_k t} \langle \eta_k, \vec{g} \rangle \langle \eta_k, \vec{g} \rangle^T$.

Comparing with (4.34), we can identify E_1 and E_2 as

$$E_1 = \hat{D}(0) - I \tag{4.37}$$

$$E_2 = \hat{D}(t) - \sum_{k=1}^{K} e^{\lambda_k t} \langle \eta_k, \vec{g} \rangle \langle \eta_k, \vec{g} \rangle^T . \tag{4.38}$$

Since E_1 is purely a statistical perturbation, we expect the magnitude of E_1 to scale as $1/\sqrt{N}$. To show that E_2 vanishes, we rewrite it as

$$\begin{aligned} E_2 = &\left(\hat{D}(t) - \sum_{k=1}^{\infty} e^{\lambda_k t} \langle \eta_k, \vec{g} \rangle \langle \eta_k, \vec{g} \rangle^T \right) \\ &+ \sum_{k=K+1}^{\infty} e^{\lambda_k t} \langle \eta_k, \vec{g} \rangle \langle \eta_k, \vec{g} \rangle^T \end{aligned} \tag{4.39}$$

The first term is purely statistical, and is expected to decay as $1/\sqrt{N}$ as before. The second term is a systematic error, which decays as t becomes substantially larger than λ_k. Consequently, as t and N becoming large, we expect the norm of E_1 and E_2 to vanish. We can therefore justified in applying (4.35) and have the bound

$$\sin(\theta_1) \leq \left\| \hat{D}(0) - I \right\|_2 + \frac{\left\| \hat{D}(t) - \sum_{k=1}^{K} e^{\lambda_k t} \langle \eta_k, \vec{g} \rangle \langle \eta_k, \vec{g} \rangle^T \right\|_2}{\omega_K}. \tag{4.40}$$

up to vanishing, higher order terms.

4.3.1 Heuristic for choice of lag time

By analyzing the behavior of each term in (4.40), we can attempt to derive a heuristic for the lag times. Immediately, we note that, E_1 is not affected by the choice of lag time. Moreover, for small perturbations we can approximate $\omega_K \approx \gamma_K$.

$$\omega_K \approx \gamma_K$$

Finally, we note that $\|E_2\|_2$ is at least as large as γ_{K+1}. This suggests that a reasonable strategy for minimizing the distance between subspaces would be to attempt to minimize

$$\frac{\gamma_{K+1}}{\gamma_K}$$

within a reasonable range of candidate lag times.

To demonstrate the behavior of our heuristic, we consider a VAC calculation on a simple example. Our system dynamics are given by the SDE

$$dx_k = -D_k x_k(t)dt + \sqrt{2D_k}dW_t, \tag{4.41}$$

where $D_k = 5^{k-1}$ in dimension k. This corresponds to a Brownian motion in a five-dimensional, isotropic harmonic well with an anisotropic diffusion. We choose a basis set of four functions, each defined according to the formula

$$\phi_i = x_1 + x_{i+1}. \tag{4.42}$$

Note that the function x_k is an eigenfunction of the generator with eigenvalue $-D_k$, as can be easily verified from (4.7). This means that each basis function is a sum of two eigenfunctions of $\mathcal{K}_t$. In fact, the dominant eigenfunction of $\mathcal{L}$ is the function $\eta_1 = x_1$ with eigenvalue -1. Orthogonality of the eigenfunctions implies that the assumption that none of top η_i's are orthogonal to the dominant slow subspace is incorrect. Consequently, for the rest of this section we will only consider the linear eigenfunctions. We will discard the orthogonal eigenfunctions, and relabel the remaining linear eigenfunctions and corresponding eigenvalues as

$$\eta_i = x_i \lambda_i = -D_i. \tag{4.43}$$

To study the behavior of VAC, we simulate this system for 10000 steps at a timestep of 0.001 time units. We then construct the matrices $\hat{C}(t)$ and $\hat{C}(0)$ for a variety of lag times between 1 and 5000 steps and calculate the top three eigenvalues and associated generalized eigenvectors and timescales for each lag time.

We aim to see how well the results of VAC align with the slowest degrees of freedom. To quantify the error in our VAC calculation, we first project the eigenfunctions with nonzero overlap with the basis onto the basis functions.

$$a^{\mathrm{TRUE}} = C(0)^{-1} \langle \vec{u}, \vec{x} \rangle . \tag{4.44}$$

We then quantify how well VAC captures the subspace spanned by the M slowest x_i. To do so, we construct a matrix whose columns are a^{TRUE} for the dominant M degrees of freedom, and apply a thin QR decomposition [168] to orthogonalize the coefficients:

$$A^{\mathrm{TRUE}} = Q^{\mathrm{TRUE}} R^{\mathrm{TRUE}}. \tag{4.45}$$

The columns of the resulting Q matrix gives new, orthogonalized coefficients that span the same space as the original coefficients, and the associated linear combinations of basis functions span the same space of $L^2(\Omega, \mu)$. We then apply the same procedure to the coefficients from the TICA calculation to get analogous orthogonalized coefficients, which we call Q^{VAC}. The degree to which the subspaces overlap can be measured using the projection metric [169, 170]

$$D(Q^{\mathrm{TRUE}}, Q^{\mathrm{VAC}}) = \sqrt{M - \sum_{i=1}^{M} s_i^2}, \tag{4.46}$$

where the s_i are the singular values of $(Q^{\mathrm{TRUE}})^T Q^{\mathrm{VAC}}$.

In Figure 4.2, we plot the projection metric between the true and the estimated subspaces as a function of the lag time for subspaces of dimension one, two, and three. We see that if the lag time is too short, the scheme fails to find the correct subspace due to contributions

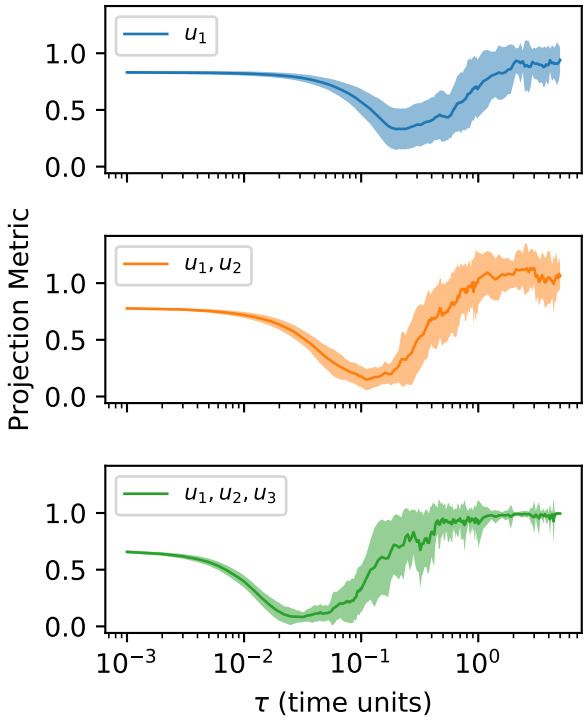


Figure 4.2: Error in choosing the slowest one-dimensional (top), two-dimension (middle), and three-dimensional (bottom) subspace on the multidimensional harmonic toy example. The solid line is the mean and the shaded region corresponds to one standard deviation from the mean, estimated over 20 replicates.

from the lower eigenfunctions. However, if the lag time is too long, the error due to sampling dominates, again causing the VAC scheme to give the incorrect result. The best answer is given by choosing a lag time that balances these factors. We note, however, that the optimal lag time depends on the size of the subspace that is desired. The best estimate for a one-dimensional subspace is achieved at a lag time of approximately 0.1 time units, whereas the best estimate for a three-dimensional subspace is achieved at a lag time that is an order of magnitude shorter. In particular, at the optimal lag time for the three-dimensional subspace, the projection metric for the one-dimensional subspace shows that the dominant eigenvector has poor overlap with the true best projection. These results suggest that, for arbitrary basis functions, one cannot expect a single lag time to give the best projection of each of the top M eigenfunctions individually. However, for a well-chosen lag time, VAC is able to

extract the subspace *spanned* by the projections of the top eigenfunctions onto the basis.

As discussed in Subsection 4.2.4, lag times for VAC schemes are currently chosen by implied timescale analysis. In Figure 4.3 we plot the implied timescales (top), and the eigenvalues on a log-log plot. While it is difficult to see where, if at all, the implied timescales

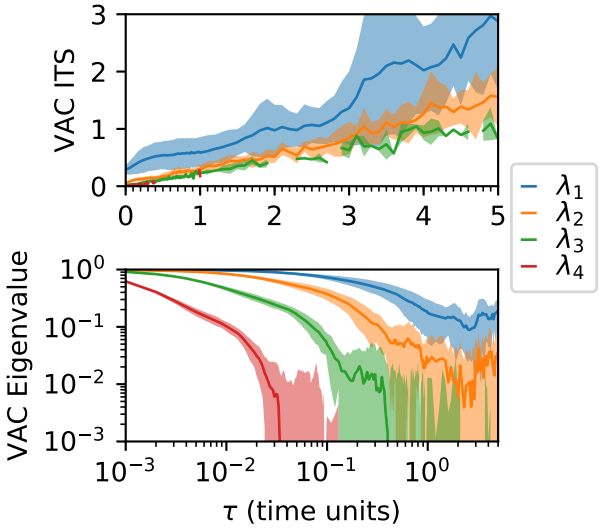


Figure 4.3: (Top) Implied timescales for the harmonic toy system, averaged over twenty replicates. (Bottom) VAC eigenvalues as a function of lag time, on a log-log scale. Note that there are missing values in both plots, as for sufficiently long lag times, sampling error may cause eigenvalues to become negative. At these points, the average is ommitted.

converge, the curve of the implied timescales seems reasonably flat at 0.5 time units. However, referring to Figure 4.2, we see that this lag time is suboptimal even for choosing the dominant subspace correctly. Moreover, the two-dimensional and three-dimensional subspaces are almost completely untrustworthy.

An alternative approach is to look at the cumulative kinetic variance [171], defined as

$$CV\left(K\right) = \frac{\sum_{i=1}^{K} \gamma_i^2}{\sum_{i=1}^{M} \gamma_i^2} \tag{4.47}$$

The cumulative kinetic variance was proposed as a method for choosing the dimensionality of S_K^F [171]. Specifically, one would specify a lag time and approximate CV as a function

of K. One would then choose the first K where the approximation rose above a given predefined threshold. While using the cumulative kinetic variance to choose the lag time was not discussed in the paper, a natural extension of this formalism would be to choose the first time when the M-dimensional subspace has a cumulative kinetic variance greater than a certain threshold.

Our results in Section 4.3 suggest a third approach: constructing a heuristic that attempts to maximize the ratio of the K'th and the $K+1$'th eigenvalue within a reasonable range. Specifically, we propose setting t according to

$$t = \arg\max_{s \in S} \left[\frac{\hat{\gamma}_i(s)}{\hat{\gamma}_{i+1}(s)} \right] \tag{4.48}$$

$$= \arg\max_{s \in S} \left[\log\left(\hat{\gamma}_i(s)\right) - \log\left(\hat{\gamma}_{i+1}(s)\right) \right] \tag{4.49}$$

where S is a set of reasonable candidate lag times (we discuss the choice S later in the section). The second line follows from monotonicity of the log function and properties of logarithms. We consider the eigenvalues on a log-scale purely for the convenience: plotting the argument of (4.49) as a function of t allows one to easily see the time where the distance between two curves is largest.

In Figure 4.3, we plot the eigenvalues on a log-log scale. We see that as the lag time increases, the gap between the log eigenvalues steadily grows. However, as the lag time increases further, we see that statistical error begins to grow, causing the gap to potentially narrow slightly. For even longer lag times, we see that the bottom eigenvalue becomes increasingly erratic. In particular, the eigenvalue can increase with t or become negative, causing the logarithm of the ratio to not be computable. One can show that both of these behaviors occur due to statistical noise, and cannot occur in the limit of infinite data [164]. Comparing with Figure 4.2, we see that if one ignores these erratic regions, the locations where the distance between successive curves in Figure 4.3 reaches its maximum corresponds well to the location of the optimal choices for lag time. Consequently, when choosing our

lag time in (4.49), we set S to be the interval ranging from 0 to the first point where the $M+1$'th eigenvalue either becomes negative or increases.

To quantify how well equation (4.49) performs, for each replicate dataset we choose t according to (4.49). For comparison, we also choose t according to the points where the cumulative variance passes 0.9 or 0.95 and according to the implied timescale analysis. In each case, we calculate the value of the projection metric against the true M dimensional subspace (Figure 4.4). For a one-dimensional subspace, we see that all three methods give comparable results. However, for the two- and three-dimensional subspaces, (4.49) gives noticeably better results.

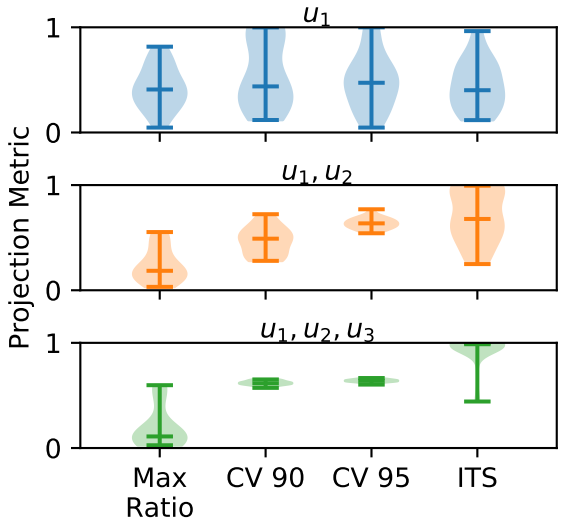


Figure 4.4: Violin plots of the projection metric, measuring the error in choosing the one, two, or three-dimensional subspace that best aligns with the true spectrum using VAC. The lag time is chosen by looking at the maximum ratio of the M and $M+1$'th eigenvalues, choosing the first lag time where the cumulative variance surpasses 90% or 95%, or by handpicking a single lag time from the plot of the implied timescales.

4.4 Lag-time selection for the alanine dipeptide

To show that our results transfer to molecular systems, we consider the task of performing dimensionality reduction on the alanine dipeptide. Our dataset consists of a long, unbiased

trajectory of thlength 150 ps. We take as our basis functions the sin and cos functions on each of the four backbone dihedral angles. We then estimated the VAC coefficients for a variety of lag times. To have a reference for the "true" eigenfunctions, we constructed a Markov state model on a separate dataset consisting of 1.5 ns. The Markov model was constructed using 1000 states on the space of all RMSD fit heavy atom positions with a lag time of 1 ps. Dominant eigenvectors were independent of the lag time used, and eigenvectors did not vary as the number of states were increased or decreased.

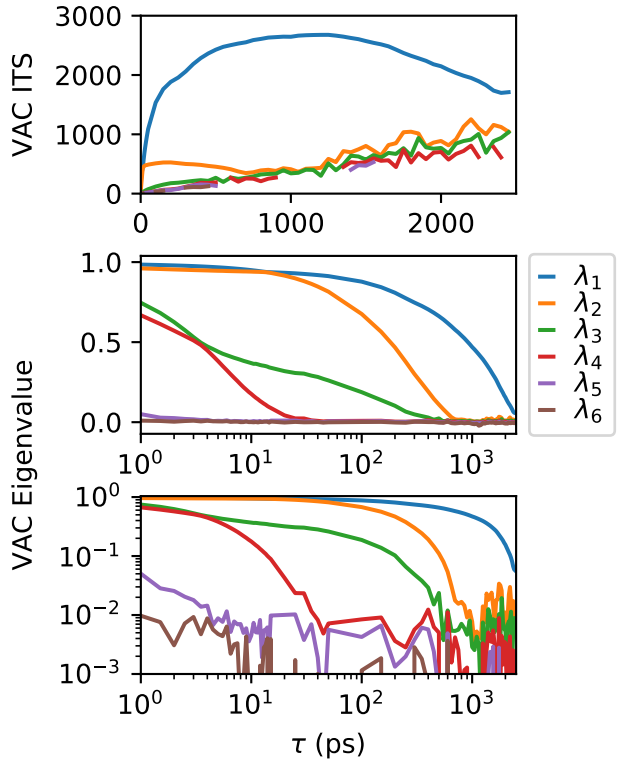


Figure 4.5: Top six VAC implied timescales (top) and eigenvalues on a linear (center) and logarithmic scale (bottom) for the alanine dipeptide using a basis of sine and cosine functions on the dihedral angles.

We show the implied timescales and the top six eigenvalues as a function of lag time in Figure 4.5. The implied timescale analysis seems to suggest a lag time of somewhere between 500 and 1500 ps. Directly plotting the eigenvalues against the lag time on a log-scale shows additional structure. In particular, we see that $e^{\lambda_1 t}$ and $e^{\lambda_2 t}$ approach each other near a lag

time of 20 ps, as do $e^{\lambda_3 t}$ and $e^{\lambda_4 t}$ near a lag time of 4 ps.

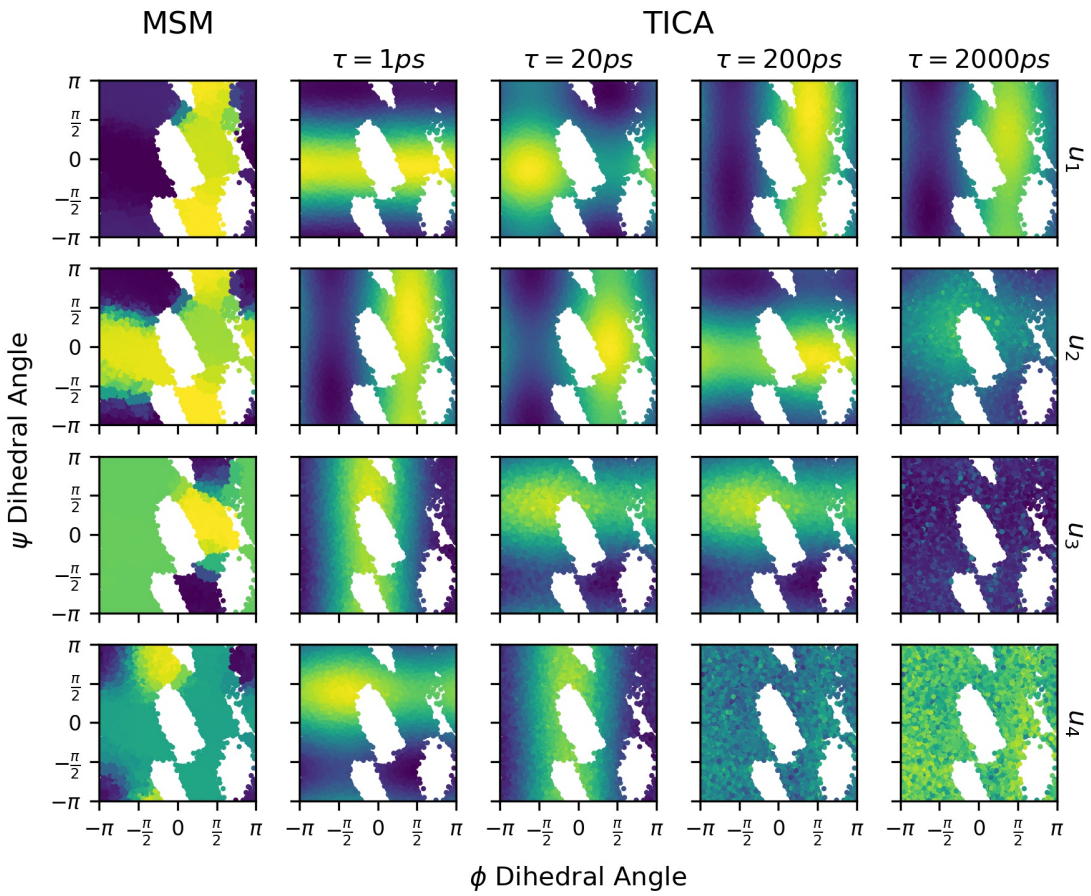


Figure 4.6: VAC results for various lag times, compared against the true dominant eigenfunctions (left-most column).

In Figure 4.6, we plot an example of the eigenfunctions projected onto the ϕ and ψ dihedral angles; all estimated eigenfunctions have negligible contribution from the other two backbone dihedrals. We see that at all lag times, the estimated eigenfunctions correlate strongly with the projections of the associated eigunfunctions. This should not be expected to hold in general, as our other numerical experiments show. At a very short lag time (1 ps), we see that the ordering of u_1 and u_2 are switched: u_1 is similar to the projection of u_2 and vice versa. Similarly, the order of the third and fourth eigenfunctions is also switched. As the time lag is increased to 20 ps, we recover the correct ordering of the third and fourth projected eigenfunctions. Note that this occurs after the point where the curves

for the third and fourth eigenvalue approach each other. As we further increase the lag to 200 ps, we recover the correct ordering for the top three projected eigenfunctions. However, at this point the fourth projected eigenfunction is no longer recoverable. Again, we can see evidence for this behavior by looking at the eigenvalues: by this time, the fourth eigenvalue has decayed to zero. Moreover, we see that the first and second eigenvalues have approached and then diverged.

Just as we saw with the harmonic example, we see that no single lag time can be used to accurately construct a collective variable space of arbitrary dimension. While choosing a long time would allow us to capture the right dominant eigenfunction, it would prevent us from accurately estimating any eigenfunctions further in the spectrum. Conversely, while choosing a short lag time allows one to resolve the *span* of the top k eigenfunctions projected onto the basis, we are not guaranteed that the dominant VAC eigenfunction corresponds to the true dominant eigenfunction.

We therefore test the performance of our heuristic for choosing the lag time. Specifically, we replicate the simulation 20 times with different random seeds. We then evaluate the VAC scheme described above at various lag times and see how well various heuristics for choosing the lag time perform. Our results suggest that our heuristic outperforms the other competing approaches.

4.5 Conclusion

In this work, we evaluated the accuracy of VAC schemes in estimating the slowest modes for a system's dynamics. We describe the short and long-time behavior of VAC schemes, and give theoretical results describing the asymptotic error of the scheme in the limit of a large number of samples. Our analysis suggests that, unless the true eigenfunctions lie very close to the span of the basis functions, the accuracy of the VAC scheme can depend heavily on the choice of lag time. For short lag times, faster modes may contribute substantially to VAC estimates of the system's slowest modes. Consequently, the dominant eigenfunctions

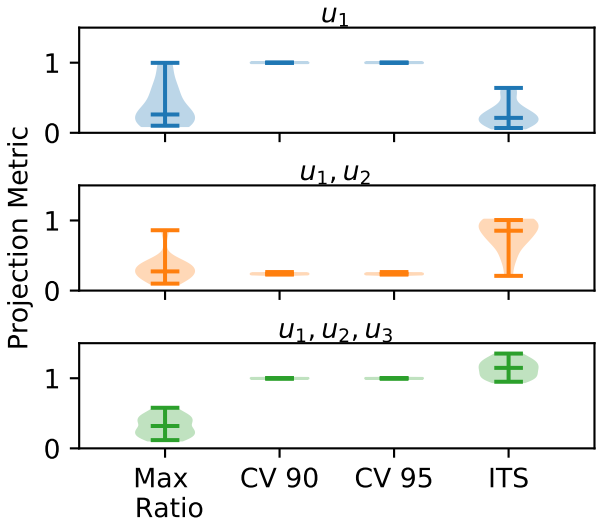


Figure 4.7: Violin plots of the projection metric for the alanine dipeptide, measuring the error in choosing the one, two, or three-dimensional subspace that best aligns with the true spectrum using VAC. The lag time is chosen by looking at the maximum ratio of the M and $M+1$'th eigenvalues, choosing the first lag time where the cumulative variance surpasses 90% or 95%, or by handpicking a single lag time from the plot of the implied timescales.

of VAC may have little to do with the optimal projection of the slow modes onto the basis, even in the limit of infinite data. This is particularly true for algorithms such as TICA, which attempt dimension reduction using a set of simple basis functions that are known to be incomplete. For the special case of TICA applied to a system undergoing Brownian dynamics, we have the added result that rather than giving the linear functions best aligned with the slow degrees of freedom, TICA diagonalizes the average diffusion coefficient. Indeed, we demonstrate on a simple double-well potential that, at short lag times, TICA may contain no information about the system's slow degrees of freedom.

At longer lag times, VAC schemes are able to recover projections of slow modes. However, as the lag time is increased, VAC schemes become increasingly sensitive to the amount of sampling error. Consequently, accurately estimating the slow mode requires setting the lag time sufficiently long that faster modes do not contribute to dynamical estimates, but not so long that sampling error dominates the estimate. Furthermore, we find that the precise

point where this occurs may be different depending on how many slow modes one wishes to retain. A lag time that gives the optimal projection of the slowest mode may be too long to capture the next slowest modes. Similarly, a lag time that accurately projects the M slowest modes may be too short to accurately distinguish between the slowest and second slowest modes. This suggests that, unless the basis is sufficiently good, at any given lag time the VAC eigenfunctions can be trusted to span an M dimensional projection of a slow subspace. However, within that subspace, the k'th eigenfunction should not be expected to correspond to the k'th slow mode. We demonstrate these results both on a simple harmonic well and on the alanine dipeptide.

Finally, since choosing the lag time correctly can be critical to getting good results with VAC schemes, we propose a novel heuristic for choosing the lag time based on the ratio of the eigenvalues. Unlike implied timescale analysis, the dominant method for choosing lag times, our heuristic can be automated and does not require human judgement. Moreover, our heuristic accounts for the dimensionality of the subspace the VAC scheme is attempting to estimate, which our numerical results suggest can drastically change the optimal choice of lag time. Our numerical results suggest that, compared with implied timescale analysis, this heuristic can give substantially improved estimates of slow subdominant processes.

CHAPTER 5

OUTLOOK

In this work, we study the effect of sampling error on the output of molecular dynamics simulations. Particular emphasis is put on understanding the performance of enhanced sampling schemes, which attempt to increase the probability of seeing key rare events in the simulated dataset.

In Chapter 2, we give a new method for recombining the data in umbrella sampling calculations. This algorithm allows one to track how errors are propagated from sampling in individual windows to the final calculated chemical quantities. Not only does this allow scaling arguments that describe the performance of the scheme to be derived, it gives quantitative error estimates that account for the effect of autocorrelation in the sampled trajectories. This comes at the expense of having reduced statistical accuracy compared to iterative schemes such as MBAR or WHAM. Consequently, future work may focus on using similar techniques to derive asymptotic variances for iterative schemes. Indeed, initial explorations suggest that it might be possible to do this for MBAR and iterative EMUS.

We then turn our attention to enhanced sampling schemes for the calculation of dynamical quantities such as reaction rates or committors. In Chapter 3, we give the Dynamical Galerkin Approximation, a scheme for estimating these quantities by solving Feynman-Kac formulae. We then approximate the solution using a basis set expansion, and estimate the resulting matrix elements using averages over short trajectories. This allows one to estimate dynamical quantities using only collections of short trajectories, avoiding the need to perform the challenging sampling task of harvesting full reactive trajectories. Our method generalizes estimates of dynamical quantities from Markov State Models. We also show that applying delay embedding to these estimates can give improved estimates in the face of missing collective variables.

While we give an initial algorithm for constructing a basis set, we believe that the construction of new basis sets could lead to further improvement in dynamical estimates. For

instance, one could attempt to construct tensor-product bases, such as those previously used in VAC schemes [115]. Development of a variational principle for the associated Feynman-Kac formulae would give further flexibility, allowing the problems to be solved using nonlinear basis sets such as the output of neural networks.

In Chapter 4, we investigate how sampling error affects the approximations of slow subspaces estimated using VAC. Our work highlights the nontrivial interplay between the choice of basis set, the amount of data available, and the choice of the lag time. In particular, we show that at small lag times VAC schemes may give results that have no information about the system's slow degrees of freedom. However, for lag times that are too long, information about subdominant slow processes is easily lost due to statistical noise. Using perturbation arguments and numerical experiments, we quantify the error in VAC estimates and argue for new heuristics for choosing the lag time. Similar arguments could potentially be extended to the SVD calculation in VAMP algorithms to further generalize this work. One could also consider the connections between our analysis and attempts to variationally optimize VAC basis sets. In particular, further analysis that accounts for the effect of sampling might allow for more robust selection across basis sets.

REFERENCES

[1] Dan Foreman-Mackey and Jonathan Goodman. ACOR 1.1.1. https://pypi.python.org/pypi/acor/1.1.1.

[2] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187, 1977.

[3] Aaron R Dinner, Jonathan C Mattingly, Jeremy OB Tempkin, Brian Van Koten, and Jonathan Weare. Trajectory stratification of stochastic dynamics. *SIAM Review*, 60(4):909–938, 2018.

[4] Stefan Klus, Feliks Nüske, Péter Koltai, Hao Wu, Ioannis Kevrekidis, Christof Schütte, and Frank Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28(3):985–1010, 2018.

[5] Yehuda Vardi. Empirical distributions in selection bias models. *The Annals of Statistics*, pages 178–203, 1985.

[6] Richard D. Gill, Yehuda Vardi, and Jon A. Wellner. Large sample theory of empirical distributions in biased sampling models. *Ann. Statist.*, 16(3):1069–1112, 09 1988.

[7] Shankar Kumar, Djamal Bouzida, Robert H. Swendsen, Peter A. Kollman, and John M Rosenberg. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[8] Michael R Shirts and John D Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *The Journal of Chemical Physics*, 129(12):124105, 2008.

[9] Edina Rosta and Gerhard Hummer. Free energies from dynamic weighted histogram analysis using unbiased Markov state model. *Journal of Chemical Theory and Computation*, 11(1):276–285, 2014.

[10] Antonia SJS Mey, Hao Wu, and Frank Noé. xTRAM: Estimating equilibrium expectations from time-correlated simulation data at multiple thermodynamic states. *Physical Review X*, 4(4):041018, 2014.

[11] Zhiqiang Tan, Emilio Gallicchio, Mauro Lapelosa, and Ronald M Levy. Theory of binless multi-state free energy estimation with applications to protein-ligand binding. *The Journal of Chemical Physics*, 136(14):144102, 2012.

[12] Tony Lelièvre, Gabriel Stoltz, and Mathias Rousset. *Free Energy Computations: A Mathematical Perspective*. World Scientific, 2010.

[13] David DL Minh and John D Chodera. Optimal estimators and asymptotic variances for nonequilibrium path-ensemble averages. *The Journal of Chemical Physics*, 131(13):134110, 2009.

[14] Fangqiang Zhu and Gerhard Hummer. Convergence and error estimation in free energy calculations using the weighted histogram analysis method. *Journal of Computational Chemistry*, 33(4):453–465, 2012.

[15] Erik H Thiede, Brian Van Koten, Jonathan Weare, and Aaron R Dinner. Eigenvector method for umbrella sampling enables error analysis. *The Journal of Chemical Physics*, 145(8):084115, 2016.

[16] Aaron R Dinner, Erik H Thiede, Brian Van Koten, and Jonathan Weare. Stratification of Markov chain Monte Carlo. *arXiv preprint arXiv:1705.08445*, 2017.

[17] Gene H Golub and Carl D Meyer, Jr. Using the QR factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains. *SIAM Journal on Algebraic Discrete Methods*, 7(2):273–281, 1986.

[18] Hans Schneider. The concepts of irreducibility and full indecomposability of a matrix in the works of Frobenius, König and Markov. *Linear Algebra and its applications*, 18(2):139–162, 1977.

[19] Zhiqiang Tan. On a likelihood approach for Monte Carlo integration. *Journal of the American Statistical Association*, 99(468):1027–1036, 2004.

[20] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*, volume 1. Academic press, 2001.

[21] Charles J Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, pages 473–483, 1992.

[22] Himanshu Paliwal and Michael R Shirts. Using multistate reweighting to rapidly and efficiently explore molecular simulation parameters space for nonbonded interactions. *Journal of Chemical Theory and Computation*, 9(11):4700–4717, 2013.

[23] Michael R Shirts, David L Mobley, and John D Chodera. Alchemical free energy calculations: Ready for prime time. *Ann Rep Comput Chem*, 3(4):41–59, 2007.

[24] Mark James Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.

[25] Gareth A Tribello, Massimiliano Bonomi, Davide Branduardi, Carlo Camilloni, and Giovanni Bussi. PLUMED 2: New feathers for an old bird. *Computer Physics Communications*, 185(2):604–613, 2014.

[26] Alexander D MacKerell, Nilesh Banavali, and Nicolas Foloppe. Development and current status of the CHARMM force field for nucleic acids. *Biopolymers*, 56(4):257–265, 2000.

[27] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman JC Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of Computational Physics*, 23(3):327–341, 1977.

[28] Alan Grossfield. WHAM: the weighted histogram analysis method (version 2.0.9), 2013.

[29] Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

[30] David Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, 1987.

[31] Christophe Chipot and Andrew Pohorille. *Free Energy Calculations*. Springer, 2007.

[32] JS van Duijneveldt and D Frenkel. Computer simulation study of free energy barriers in crystal nucleation. *The Journal of Chemical Physics*, 96(6):4655–4668, 1992.

[33] Jeffrey Comer, James C Gumbart, Jrme Hnin, Tony Lelivre, Andrew Pohorille, and Christophe Chipot. The adaptive biasing force method: everything you always wanted to know but were afraid to ask. *The Journal of Physical Chemistry B*, 119(3):1129–1151, 2014.

[34] Peter Virnau and Marcus Müller. Calculation of free energy through successive umbrella sampling. *The Journal of Chemical Physics*, 120(23):10925–10930, 2004.

[35] Frank P Kelly. *Reversibility and Stochastic Networks*. Cambridge University Press, 2011.

[36] Tony Lelièvre, Gabriel Stoltz, and Mathias Rousset. *Free Energy Computations: A Mathematical Perspective*. Imperial College Press, Hackensack, N.J., 2010.

[37] Martin Bilodeau and David Brenner. *Theory of Multivariate Statistics*. Springer Science & Business Media, 2008.

[38] Erik Thiede, Brian Van Koten, and Jonathan Weare. Sharp entrywise perturbation bounds for Markov chains. *SIAM Journal on Matrix Analysis and Applications*, 36(3):917–941, 2015.

[39] A. W. van der Vaart. *Asymptotic Statistics*. Cambridge series on statistical and probabilistic mathematics. Cambridge University Press, Cambridge ; New York, 1998.

[40] Sylvia. Richardson and Peter J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

[41] A. Jasra, C. C. Holmes, and D. A. Stephens. Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statist. Sci.*, 20(1):50–67, 02 2005.

[42] Nicolas Chopin, Tony Lelièvre, and Gabriel Stoltz. Free energy methods for Bayesian inference: efficient exploration of univariate Gaussian mixture posteriors. *Statistics and Computing*, 22(4):897–916, 2012.

[43] Murray Aitkin. Likelihood and Bayesian analysis of mixtures. *Statistical Modelling*, 1(4):287–304, 2001.

[44] Alan J. Izenman and Charles J. Sommer. Philatelic mixtures and multimodal densities. *Journal of the American Statistical Association*, 83(404):941–953, 1988.

[45] Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.

[46] Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

[47] Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: fifty years after Kramers. *Reviews of Modern Physics*, 62(2):251, 1990.

[48] Eric Vanden-Eijnden. Transition path theory. In *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, pages 453–493. Springer, 2006.

[49] Alexander M Berezhkovskii, Attila Szabo, Nicholas Greives, and Huan-Xiang Zhou. Multidimensional reaction rate theory with anisotropic diffusion. *The Journal of Chemical Physics*, 141(20):11B616_1, 2014.

[50] Ao Ma, Ambarish Nag, and Aaron R Dinner. Dynamic coupling between coordinates in a model for biomolecular isomerization. *The Journal of chemical physics*, 124(14):144911, 2006.

[51] Victor Ovchinnikov, Kwangho Nam, and Martin Karplus. A simple and accurate method to calculate free energy profiles and reaction rates from restrained molecular simulations of diffusive processes. *The Journal of Physical Chemistry B*, 120(33):8457–8472, 2016.

[52] An Ghysels, Richard M Venable, Richard W Pastor, and Gerhard Hummer. Position-dependent diffusion tensors in anisotropic media from simulation: oxygen transport in and through membranes. *Journal of chemical theory and computation*, 13(6):2962–2976, 2017.

[53] Aaron R Dinner and Martin Karplus. The thermodynamics and kinetics of protein folding: A lattice model analysis of multiple pathways with intermediates. *The Journal of Physical Chemistry B*, 103(37):7976–7994, 1999.

[54] Aaron R Dinner, Andrej Šali, Lorna J Smith, Christopher M Dobson, and Martin Karplus. Understanding protein folding via free-energy surfaces from theory and experiment. *Trends in Biochemical Sciences*, 25(7):331–339, 2000.

[55] Christoph Dellago, Peter G Bolhuis, Félix S Csajka, and David Chandler. Transition path sampling and the calculation of rate constants. *The Journal of Chemical Physics*, 108(5):1964–1977, 1998.

[56] Peter G Bolhuis, David Chandler, Christoph Dellago, and Phillip L Geissler. Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual Review of Physical Chemistry*, 53(1):291–318, 2002.

[57] Ao Ma and Aaron R Dinner. Automatic method for identifying reaction coordinates in complex systems. *The Journal of Physical Chemistry B*, 109(14):6769–6779, 2005.

[58] Jie Hu, Ao Ma, and Aaron R Dinner. A two-step nucleotide-flipping mechanism enables kinetic discrimination of DNA lesions by AGT. *Proceedings of the National Academy of Sciences*, 105(12):4615–4620, 2008.

[59] Michael Grünwald, Christoph Dellago, and Phillip L Geissler. Precision shooting: Sampling long transition pathways. *The Journal of Chemical Physics*, 129(19):194101, 2008.

[60] Todd R Gingrich and Phillip L Geissler. Preserving correlations between trajectories for efficient path sampling. *The Journal of Chemical Physics*, 142(23):06B614_1, 2015.

[61] Gary A Huber and Sangtae Kim. Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophysical Journal*, 70(1):97, 1996.

[62] Titus S van Erp, Daniele Moroni, and Peter G Bolhuis. A novel path sampling method for the calculation of rate constants. *The Journal of Chemical Physics*, 118(17):7762–7774, 2003.

[63] Anton K Faradjian and Ron Elber. Computing time scales from reaction coordinates by milestoning. *The Journal of Chemical Physics*, 120(23):10880–10889, 2004.

[64] Rosalind J Allen, Daan Frenkel, and Pieter Rein ten Wolde. Simulating rare events in equilibrium or nonequilibrium stochastic systems. *The Journal of chemical physics*, 124(2):024102, 2006.

[65] Aryeh Warmflash, Prabhakar Bhimalapuram, and Aaron R Dinner. Umbrella sampling for nonequilibrium processes. *The Journal of Chemical Physics*, 127(15):114109, 2007.

[66] Eric Vanden-Eijnden and Maddalena Venturoli. Exact rate calculations by trajectory parallelization and tilting. *The Journal of chemical physics*, 131(4):044120, 2009.

[67] Alex Dickson, Aryeh Warmflash, and Aaron R Dinner. Separating forward and backward pathways in nonequilibrium umbrella sampling. *The Journal of Chemical Physics*, 131(15):154104, 2009.

[68] Nicholas Guttenberg, Aaron R Dinner, and Jonathan Weare. Steered transition path sampling. *The Journal of Chemical Physics*, 136(23):06B609, 2012.

[69] Juan M Bello-Rivas and Ron Elber. Exact milestoning. *The Journal of Chemical Physics*, 142(9):03B602_1, 2015.

[70] Christof Schütte, Alexander Fischer, Wilhelm Huisinga, and Peter Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *Journal of Computational Physics*, 151(1):146–168, 1999.

[71] William C Swope, Jed W Pitera, and Frank Suits. Describing protein folding kinetics by molecular dynamics simulations. 1. Theory. *The Journal of Physical Chemistry B*, 108(21):6571–6581, 2004.

[72] Vijay S Pande, Kyle Beauchamp, and Gregory R Bowman. Everything you wanted to know about Markov state models but were afraid to ask. *Methods*, 52(1):99–105, 2010.

[73] Marco Sarich, Frank Noé, and Christof Schütte. On the approximation quality of Markov state models. *Multiscale Modeling & Simulation*, 8(4):1154–1177, 2010.

[74] Frank Noé and Stefan Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Current Opinion in Structural Biology*, 18(2):154–162, 2008.

[75] Frank Noé and Jan-Hendrik Prinz. Analysis of Markov models. In Gregory R Bowman, Vijay S Pande, and Frank Noé, editors, *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation, vol. 797 of Advances in Experimental Medicine and Biology*, chapter 6. Springer, 2014.

[76] Bettina G Keller, Stevan Aleksic, and Luca Donati. Markov state models in drug design. In Francesco L Gervasio and Vojtech Spiwok, editors, *Biomolecular Simulations in Drug Discovery*, chapter 4. Wiley-VCH, 2019.

[77] Marcus Weber. *Meshless Methods in Conformation Dynamics*. PhD thesis, Freie Universität Berlin, 2006.

[78] Frank Noé and Feliks Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.

[79] Erik H Thiede, Dimitrios Giannakis, Aaron R Dinner, and Jonathan Weare. Galerkin approximation of dynamical quantities using trajectory data. *arXiv preprint arXiv:1810.01841*, 2018.

[80] Tanja Eisner, Bálint Farkas, Markus Haase, and Rainer Nagel. *Operator Theoretic Aspects of Ergodic Theory*, volume 272. Springer, 2015.

[81] Patrick Billingsley. *Probability and Measure*. John Wiley & Sons, 2008.

[82] Gregory R Bowman, Kyle A Beauchamp, George Boxer, and Vijay S Pande. Progress and challenges in the automated construction of Markov state models for full protein systems. *The Journal of Chemical Physics*, 131(12):124101, 2009.

[83] Jan-Hendrik Prinz, Hao Wu, Marco Sarich, Bettina Keller, Martin Senne, Martin Held, John D Chodera, Christof Schütte, and Frank Noé. Markov models of molecular kinetics: Generation and validation. *The Journal of Chemical Physics*, 134(17):174105, 2011.

[84] Peter Deuflhard, Wilhelm Huisinga, Alexander Fischer, and Ch Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315(1-3):39–59, 2000.

[85] Susanna Röblitz and Marcus Weber. Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification. *Advances in Data Analysis and Classification*, 7(2):147–179, 2013.

[86] Frank Noé, Christof Schütte, Eric Vanden-Eijnden, Lothar Reich, and Thomas R Weikl. Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106(45):19011–19016, 2009.

[87] Christian R Schwantes and Vijay S Pande. Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9. *Journal of Chemical Theory and Computation*, 9(4):2000–2009, 2013.

[88] Guillermo Pérez-Hernández, Fabian Paul, Toni Giorgino, Gianni De Fabritiis, and Frank Noé. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics*, 139(1):07B604_1, 2013.

[89] Christian R Schwantes, Robert T McGibbon, and Vijay S Pande. Perspective: Markov models for long-timescale biomolecular dynamics. *The Journal of Chemical Physics*, 141(9):09B201, 2014.

[90] Ch Schütte and Marco Sarich. A critical appraisal of Markov state models. *The European Physical Journal Special Topics*, 224(12):2445–2462, 2015.

[91] Diwakar Shukla, Carlos X Hernandez, Jeffrey K Weber, and Vijay S Pande. Markov state models provide insights into dynamic modulation of protein function. *Accounts of Chemical Research*, 48(2):414–422, 2015.

[92] Ganna Berezovska, Diego Prada-Gracia, and Francesco Rao. Consensus for the Fip35 folding mechanism? *The Journal of Chemical Physics*, 139(3):07B608_1, 2013.

[93] Fu Kit Sheong, Daniel-Adriano Silva, Luming Meng, Yutong Zhao, and Xuhui Huang. Automatic state partitioning for multibody systems (APM): an efficient algorithm for constructing Markov state models to elucidate conformational dynamics of multibody systems. *Journal of Chemical Theory and Computation*, 11(1):17–27, 2014.

[94] Yan Li and Zigang Dong. Effect of clustering algorithm on establishing Markov state model for molecular dynamics simulations. *Journal of Chemical Information and Modeling*, 56(6):1205–1215, 2016.

[95] Brooke E Husic and Vijay S Pande. Ward clustering improves cross-validated Markov state models of protein folding. *Journal of Chemical Theory and Computation*, 13(3):963–967, 2017.

[96] Brooke E Husic, Keri A McKiernan, Hannah K Wayment-Steele, Mohammad M Sultan, and Vijay S Pande. A minimum variance clustering approach produces robust and interpretable coarse-grained models. *Journal of Chemical Theory and Computation*, 14(2):1071–1082, 2018.

[97] Jan-Hendrik Prinz, John D Chodera, and Frank Noé. Spectral rate theory for two-state kinetics. *Physical Review X*, 4(1):011020, 2014.

[98] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5, 2018.

[99] Hao Wu and Frank Noé. Variational approach for learning Markov processes from time series data. *arXiv preprint arXiv:1707.04659*, 2017.

[100] Wei Wang, Siqin Cao, Lizhe Zhu, and Xuhui Huang. Constructing Markov state models to elucidate the functional conformational changes of complex biomolecules. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 8(1):e1343, 2018.

[101] Brooke E Husic and Vijay S Pande. Markov state models: From an art to a science. *Journal of the American Chemical Society*, 140(7):2386–2396, 2018.

[102] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pages 273–309. Springer, 2004.

[103] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[104] Martin K Scherer, Benjamin Trendelkamp-Schroer, Fabian Paul, Guillermo Prez-Hernndez, Moritz Hoffmann, Nuria Plattner, Christoph Wehmeyer, Jan-Hendrik Prinz, and Frank No. PyEMMA 2: a software package for estimation, validation, and analysis of Markov models. *Journal of Chemical Theory and Computation*, 11(11):5525–5542, 2015.

[105] Lutz Molgedey and Heinz Georg Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72(23):3634, 1994.

[106] Hiroshi Takano and Seiji Miyashita. Relaxation modes in random spin systems. *Journal of the Physical Society of Japan*, 64(10):3688–3698, 1995.

[107] Hidetomo Hirao, Sachiko Koseki, and Hiroshi Takano. Molecular dynamics study of relaxation modes of a single polymer chain. *Journal of the Physical Society of Japan*, 66(11):3399–3405, 1997.

[108] Christof Schütte, Frank Noé, Jianfeng Lu, Marco Sarich, and Eric Vanden-Eijnden. Markov state models based on milestoning. *The Journal of Chemical Physics*, 134(20):05B609, 2011.

[109] Dimitrios Giannakis, Joanna Slawinska, and Zhizhen Zhao. Spatiotemporal feature extraction with data-driven Koopman operators. In *Feature Extraction: Modern Questions and Challenges*, pages 103–115, 2015.

[110] Dimitrios Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Applied and Computational Harmonic Analysis*, 2017.

[111] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[112] Lorenzo Boninsegna, Gianpaolo Gobbo, Frank Noé, and Cecilia Clementi. Investigating molecular kinetics by variationally optimized diffusion maps. *Journal of Chemical Theory and Computation*, 11(12):5947–5960, 2015.

[113] F Vitalini, F Noé, and BG Keller. A basis set for peptides for the variational approach to conformational kinetics. *Journal of Chemical Theory and Computation*, 11(9):3992–4004, 2015.

[114] Feliks Nüske, Bettina G Keller, Guillermo Pérez-Hernández, Antonia SJS Mey, and Frank Noé. Variational approach to molecular kinetics. *Journal of Chemical Theory and Computation*, 10(4):1739–1752, 2014.

[115] Feliks Nüske, Reinhold Schneider, Francesca Vitalini, and Frank Noé. Variational tensor approach for approximating the rare-event kinetics of macromolecular systems. *The Journal of Chemical Physics*, 144(5):054105, 2016.

[116] Pierre Del Moral. *Feynman-Kac Formulae*. Springer, 2004.

[117] Ioannis Karatzas and Steven Shreve. *Brownian Motion and Stochastic Calculus*, volume 113. Springer Science & Business Media, 2012.

[118] Rose Du, Vijay S Pande, Alexander Yu Grosberg, Toyoichi Tanaka, and Eugene S Shakhnovich. On the transition coordinate for protein folding. *The Journal of Chemical Physics*, 108(1):334–350, 1998.

[119] Peter G Bolhuis, Christoph Dellago, and David Chandler. Reaction coordinates of biomolecular isomerization. *Proceedings of the National Academy of Sciences*, 97(11):5877–5882, 2000.

[120] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. Transition path theory for Markov jump processes. *Multiscale Modeling & Simulation*, 7(3):1192–1219, 2009.

[121] Kôsaku Yosida. Functional Analysis. *Spring-Verlag, New York/Berlin*, 1980.

[122] Mauro Lapelosa and Cameron F Abrams. Transition-path theory calculations on non-uniform meshes in two and three dimensions using finite elements. *Computer Physics Communications*, 184(10):2310–2315, 2013.

[123] Rongjie Lai and Jianfeng Lu. Point cloud discretization of Fokker–Planck operators for committor functions. *Multiscale Modeling & Simulation*, 16(2):710–726, 2018.

[124] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1, 2019.

[125] L.C. Evans. *Partial Differential Equations*. Orient Longman, 1998.

[126] Erik Thiede. PyEDGAR. https://github.com/ehthiede/PyEDGAR/, 2018.

[127] Hao Wu, Feliks Nüske, Fabian Paul, Stefan Klus, Péter Koltai, and Frank Noé. Variational Koopman models: slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of Chemical Physics*, 146(15):154104, 2017.

[128] Kevin K Chen, Jonathan H Tu, and Clarence W Rowley. Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses. *Journal of nonlinear science*, 22(6):887–915, 2012.

[129] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

[130] Tyrus Berry and John Harlim. Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 40(1):68–96, 2016.

[131] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Pablo G Debenedetti, and Ioannis G Kevrekidis. Systematic determination of order parameters for chain dynamics using diffusion maps. *Proceedings of the National Academy of Sciences*, 107(31):13597–13602, 2010.

[132] Mary A Rohrdanz, Wenwei Zheng, Mauro Maggioni, and Cecilia Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(12):03B624, 2011.

[133] Wenwei Zheng, Bo Qi, Mary A Rohrdanz, Amedeo Caflisch, Aaron R Dinner, and Cecilia Clementi. Delineation of folding pathways of a β-sheet miniprotein. *The Journal of Physical Chemistry B*, 115(44):13065–13074, 2011.

[134] Andrew L Ferguson, Athanassios Z Panagiotopoulos, Ioannis G Kevrekidis, and Pablo G Debenedetti. Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach. *Chemical Physics Letters*, 509(1):1–11, 2011.

[135] Andrew W Long and Andrew L Ferguson. Nonlinear machine learning of patchy colloid self-assembly pathways and mechanisms. *The Journal of Physical Chemistry B*, 118(15):4228–4244, 2014.

[136] Sang Beom Kim, Carmeline J Dsilva, Ioannis G Kevrekidis, and Pablo G Debenedetti. Systematic characterization of protein folding pathways using diffusion maps: Application to Trp-cage miniprotein. *The Journal of Chemical Physics*, 142(8):02B613_1, 2015.

[137] Tyrus Berry, Dimitrios Giannakis, and John Harlim. Nonparametric forecasting of low-dimensional dynamical systems. *Physical Review E*, 91(3):032915, 2015.

[138] Klaus Müller and Leo D Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theoretica Chimica Acta*, 53(1):75–93, 1979.

[139] Benedict Leimkuhler and Charles Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013(1):34–56, 2012.

[140] Kyle A Beauchamp, Gregory R Bowman, Thomas J Lane, Lutz Maibaum, Imran S Haque, and Vijay S Pande. MSMBuilder2: Modeling conformational dynamics on the picosecond to millisecond scale. *Journal of Chemical Theory and Computation*, 7(10):3412–3419, 2011.

[141] Ernesto Suárez, Joshua L Adelman, and Daniel M Zuckerman. Accurate estimation of protein folding and unfolding times: beyond Markov state models. *Journal of Chemical Theory and Computation*, 12(8):3473–3481, 2016.

[142] Natasa Djurdjevac, Marco Sarich, and Christof Schütte. Estimating the eigenvalue error of Markov state models. *Multiscale Modeling & Simulation*, 10(1):61–81, 2012.

[143] Robert Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001.

[144] Floris Takens. Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, 898(1):366–381, 1981.

[145] Dirk Aeyels. Generic observability of differentiable systems. *SIAM Journal on Control and Optimization*, 19(5):595–603, 1981.

[146] Mark R Muldoon, David S Broomhead, Jeremy P Huke, and Rainer Hegger. Delay embedding in the presence of dynamical noise. *Dynamics and Stability of Systems*, 13(2):175–186, 1998.

[147] J Stark, DS Broomhead, ME Davies, and J Huke. Takens embedding theorems for forced and stochastic systems. *Nonlinear Analysis: Theory, Methods & Applications*, 30(8):5303–5314, 1997.

[148] Tyrus Berry, John Robert Cressman, Zrinka Greguric-Ferencek, and Timothy Sauer. Time-scale separation from diffusion-mapped delay coordinates. *SIAM Journal on Applied Dynamical Systems*, 12(2):618–649, 2013.

[149] Jiang Wang and Andrew L Ferguson. Nonlinear reconstruction of single-molecule free-energy surfaces from univariate time series. *Physical Review E*, 93(3):032412, 2016.

[150] Jiang Wang and Andrew L Ferguson. Recovery of protein folding funnels from single-molecule time series by delay embeddings and manifold learning. *The Journal of Physical Chemistry B*, 122(50):11931–11952, 2018.

[151] Russell Fung, Ataya M Hanna, Oriol Vendrell, S Ramakrishna, Tamar Seideman, Robin Santra, and Abbas Ourmazd. Dynamics from noisy data with extreme timing uncertainty. *Nature*, 532(7600):471, 2016.

[152] Ernesto Suarez, Steven Lettieri, Matthew C Zwier, Carsen A Stringer, Sundar Raman Subramanian, Lillian T Chong, and Daniel M Zuckerman. Simultaneous computation of dynamical and equilibrium information using a weighted ensemble of trajectories. *Journal of Chemical Theory and Computation*, 10(7):2658–2667, 2014.

[153] Rick Durrett. *Probability: Theory and Examples*. Cambridge university press, 2010.

[154] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, and Willy Wriggers. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[155] Stefano Piana, Krishnarjun Sarkar, Kresten Lindorff-Larsen, Minghao Guo, Martin Gruebele, and David E Shaw. Computational design and experimental testing of the fastest-folding β-sheet protein. *Journal of Molecular Biology*, 405(1):43–48, 2011.

[156] Andrew W Long and Andrew L Ferguson. Landmark diffusion maps (L-dMaps): Accelerated manifold learning out-of-sample extension. *Applied and Computational Harmonic Analysis*, 2017.

[157] Christopher KI Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688, 2001.

[158] Ilya Bronshtein, Konstantin Semendyayev, Gerhard Musiol, and Heiner Muehlig. *Handbook of Mathematics*. Springer, 3 edition, 2007.

[159] Weinan E and Eric Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual Review of Physical Chemistry*, 61:391–420, 2010.

[160] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[161] William C Swope, Jed W Pitera, Frank Suits, Mike Pitman, Maria Eleftheriou, Blake G Fitch, Robert S Germain, Aleksandr Rayshubski, TJ Christopher Ward, Yuriy Zhestkov, et al. Describing protein folding kinetics by molecular dynamics simulations. 2. Example applications to alanine dipeptide and a β-hairpin peptide. *The Journal of Physical Chemistry B*, 108(21):6582–6594, 2004.

[162] Feliks Nske, Bettina G Keller, Guillermo Pérez-Hernández, Antonia SJS Mey, and Frank No. Variational approach to molecular kinetics. *Journal of Chemical Theory and Computation*, 10(4):1739–1752, 2014.

[163] Wei Chen, Hythem Sidky, and Andrew L Ferguson. Nonlinear discovery of slow molecular modes using hierarchical dynamics encoders. *arXiv preprint arXiv:1902.03336*, 2019.

[164] Erik Henning Thiede, Rob Webber, Jonathan Weare, and Aaron R Dinner. In preparation.

[165] Nina Singhal Hinrichs and Vijay S Pande. Calculation of the distribution of eigenvalues and eigenvectors in Markovian state models for molecular dynamics. *The Journal of Chemical Physics*, 126(24):244101, 2007.

[166] Yusuke Naritomi and Sotaro Fuchigami. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: the case of domain motions. *The Journal of Chemical Physics*, 134(6):02B617, 2011.

[167] Brooke E Husic and Vijay S Pande. Note: MSM lag time cannot be used for variational model selection. *The Journal of Chemical Physics*, 147(17):176101, 2017.

[168] Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU press, 2012.

[169] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[170] Liwei Wang, Xiao Wang, and Jufu Feng. Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition. *Pattern Recognition*, 39(3):456–464, 2006.

[171] Frank Noé and Cecilia Clementi. Kinetic distance and kinetic maps from molecular dynamics simulation. *Journal of Chemical Theory and Computation*, 11(10):5002–5011, 2015.