Path Optimization in Free Energy Calculations

by

Ryan Muraglia

Graduate Program in Computational Biology & Bioinformatics
Duke University

Date: ________________________
Approved:

____
Scott Schmidler, Supervisor

____
Patrick Charbonneau

____
Paul Magwene

Thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in the Graduate Program in
Computational Biology & Bioinformatics in the Graduate School
of Duke University
2016

Abstract

Path Optimization in Free Energy Calculations

by

Ryan Muraglia

Graduate Program in Computational Biology & Bioinformatics
Duke University

Date: ____________________
Approved:

____
Scott Schmidler, Supervisor

____
Patrick Charbonneau

____
Paul Magwene

An abstract of a thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science in the Graduate Program in
Computational Biology & Bioinformatics in the Graduate School
of Duke University
2016

Abstract

Free energy calculations are a computational method for determining thermodynamic quantities, such as free energies of binding, via simulation. Currently, due to computational and algorithmic limitations, free energy calculations are limited in scope. In this work, we propose two methods for improving the efficiency of free energy calculations. First, we expand the state space of alchemical intermediates, and show that this expansion enables us to calculate free energies along lower variance paths. We use Q-learning, a reinforcement learning technique, to discover and optimize paths at low computational cost. Second, we reduce the cost of sampling along a given path by using sequential Monte Carlo samplers. We develop a new free energy estimator, pCrooks (pairwise Crooks), a variant on the Crooks fluctuation theorem (CFT), which enables decomposition of the variance of the free energy estimate for discrete paths, while retaining beneficial characteristics of CFT. Combining these two advancements, we show that for some test models, optimal expanded-space paths have a nearly 80% reduction in variance relative to the standard path. Additionally, our free energy estimator converges at a more consistent rate and on average 1.8 times faster when we enable path searching, even when the cost of path discovery and refinement is considered.

Contents

List of Figures

1
Introduction

The free energy difference between two states is a highly sought after quantity, as it determines their macroscopic behavior. The states can be constructed with flexibility, such that we can obtain information about various phenomena including, but not limited to, protein binding and protein folding. Given current methods and computational resources, free energy calculations for biologically relevant macromolecules remain impractical[1,2].

This impracticality arises due to the heavy cost of sampling conformations for large molecular systems. An ideal sampler for free energy calculations will address the two main challenges current samplers face. First, it must sample the entire configuration space efficiently. The sampler must be able to move across regions of low density to sample a multitude of potential wells. Second, it must collect draws from not only the two systems of interest, but also a series of intermediate distributions which provide a sequence of maximum overlap, connecting the two systems of interest.

The objective of this research is to develop an efficient sampler that will negotiate these challenges, making reliable free energy calculations for macromolecules possible.

To meet this goal, three avenues are explored. The first aim of this research is to explore alternative, multivariate parameterizations of the potential function that are suitable for polypeptide systems. I demonstrate this idea by adding a temperature parameter to the classic λ-scaling intermediate distribution generation scheme. This increased dimensionality makes the sampler more flexible, at the cost of an increase in the difficulty of determining good sequences of intermediate distributions. For several simple models of increasing complexity, an exhaustive graph search over the space of intermediate distributions is used to determine the best set of bridging densities (hereafter, the optimal path), revealing the benefits of this more flexible scheme.

The second aim of this research is to develop a low cost sampling method compatible with the path searching paradigm introduced in the first aim. Crooks[3] and Jarzynski[4] have previously explored the application of Sequential Monte Carlo (SMC) samplers[5,6] to free energy estimation, and generated renewed interest in cost efficient nonequilibrium sampling. We present a new sampling and estimation algorithm, pCrooks, a pairwise extension of the Crooks Fluctation Theorem (CFT), which maintains the computational benefits of SMC samplers while providing detailed information on each transition in the alchemical path, allowing for interfacing with path optimization algorithms.

In order for the expanded state space from the first aim to be practical, the cost of finding an improved path must be less than the gains afforded by its use. Computational effort must be allocated between the competing tasks of drawing samples for free energy estimation and drawing samples for path space exploration. In the third aim, we apply reinforcement learning techniques, such as Q-learning[7], to efficiently optimize the path without incurring a large sampling burden and computational cost for the exploration phase.

2

Background

2.1 Free energy calculations

Biological systems of fixed connectivity can be represented as a set of conformations that are defined by a molecular topology, with varying bond lengths, bond angles, torsional angles and atomic positions[1,2,8]. In the canonical ensemble, these conformations are Boltzmann distributed, meaning they follow the distribution $p(\boldsymbol{x}) = \exp(-\beta U(\boldsymbol{x}))/Z$, where β is the inverse temperature of the system, $\boldsymbol{x}$ represents a conformation, $U(\boldsymbol{x})$ represents the potential energy of a conformation $\boldsymbol{x}$, and Z represents the partition function, defined as $Z = \int_\Omega \exp(-\beta U(\boldsymbol{x}))d\boldsymbol{x}$, where Ω represents the complete set of conformations. Hereafter, the terms partition function and normalizing constant will be used interchangeably, and $\exp(-\beta U(\boldsymbol{x}))$ will be referred to as the Boltzmann weight, unnormalized density or $q(\boldsymbol{x})$.

The free energy of a system is given by $G = -\beta^{-1}\log(Z)$, and the free energy difference between two systems is given by:

$$\Delta G_{1\to 2} = -\beta^{-1}\log(Z_2/Z_1) \tag{2.1}$$

These systems are frequently selected to define the bound and unbound states of a protein-ligand pair, making the free energy difference a ΔG_{bind}, which is representative of the likelihood of the binding reaction. Furthermore, the difference between the ΔG_{bind} of two different protein-ligand pairs (their $\Delta\Delta G$) informs which pair binds better to each other. Computational prediction of $\Delta\Delta G$ for minor ligand modifications could significantly speed up lead optimization steps in a drug development context, by focusing the search on promising modifications selected by a simulation-based prescreening step.

For biological systems of interest, namely those describing interactions between a macromolecule and a small molecule or protein ligand, the partition function is unfeasible to calculate directly due to the size of the conformation space and the impossibility of writing out the partition function analytically[8]. The goal of a free energy calculation is to estimate partition functions, or more commonly, ratios thereof.

Free energy calculations involve two components[1,8]: conformational sampling and estimation. Because the distributions of macromolecules are complex, multimodal and high dimensional[1,8], with large regions of low density, sampling requires specialized methods. The most widely used method for conformational sampling is molecular dynamics (MD)[8], which generates conformations by simulating the time varying movement of a macromolecule by numerical integration of the equations of motion. For large macromolecular systems, this is a very computationally costly process. With typical current computational resources, the µs time scale is the current limit of timescales accessible via MD[9,10]. Relevant biological movements for macromolecules such as allosteric modulation and molecular recognition are thought to occur on the µs to ms time scale[11]. This disparity means that direct simulation of binding events and calculation of ΔG_{bind} is impossible.

However, free energy is a state function, and as a result, ΔG_{bind} can be expressed as the sum of an alternate set of free energies along a different path, as illustrated in the thermodynamic cycle in figure 2.1. This alternate pathway involves the compu-

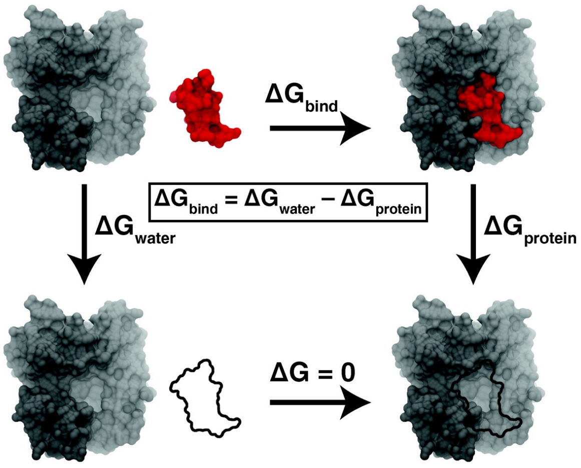

FIGURE 2.1: The thermodynamic cycle, adapted from Durrant and McCammon[12]

tation of two free energies of solvation, ΔG_{water} and $\Delta G_{protein}$, both of which depend only on small conformational changes accessible to MD, such as side chain rotations, which occur on the ps to ns scale[11].

2.2 Bridge sampling, the BAR estimator and beyond

The problem of directly calculating ΔG_{bind} has been recast as the problem of indirectly calculating ΔG_{bind} via the thermodynamic cycle. Given a collection of sampled conformations, what remains is the second phase of free energy calculation: estimation. A wide variety of estimators exist[1,2,8], but it has been shown that asymptotically, estimators derived from bridge sampling methods, notably the Bennett acceptance ratio (BAR)[13], are unbiased and have the lowest variance of known estimators[1,14,15]. Additionally, in empirical tests, BAR has been shown to outperform competing algorithms, specifically exponential averaging (EXP) and thermodynamic integration (TI)[16].

Bridge sampling is a generalized form of BAR from the statistics literature, used for estimating ratios of normalizing constants[17–19]. The bridge sampling identity is:

$$r \equiv \frac{c_1}{c_2} = \frac{E_2[q_1(w)\alpha(w)]}{E_1[q_2(w)\alpha(w)]} \tag{2.2}$$

where c_i is the normalizing constant, $\alpha(w)$ is an arbitrary function, and E_i denotes the expectation with respect to p_i. Given a set of draws from p_1 and p_2, the ratio estimate is:

$$\hat{r} = \frac{\frac{1}{n_2}\sum_{j=1}^{n_2} q_1(w_{2j})\alpha(w_{2j})}{\frac{1}{n_1}\sum_{j=1}^{n_1} q_2(w_{1j})\alpha(w_{1j})} \tag{2.3}$$

where $\{w_{i1}, ..., w_{in}\}$ are draws from p_i, $i = 1, 2$.

Equation (2.3) defines the bridge sampling class of estimators for a range of α functions. BAR is defined by the choice of α as:

$$\alpha \propto (s_1 q_1 + s_2 r q_2)^{-1} \tag{2.4}$$

where $s_i = n_i/(n_1 + n_2)$. This α minimizes the asymptotic variance of $\log(\hat{r})$ as well as the asymptotic relative variance, $E(\hat{r} - r)^2/r^2$, when the draws are independent[17]. As α depends on the unknown ratio, iterative methods are used to estimate r[19].

From equation (2.2), it is apparent that when the phase space overlap between p_1 and p_2 is small, the variance of $\hat{r}$ will be large: the expectations are dominated by rare sampling events in the small overlap region. For macromolecules, small phase space overlap is the rule, rather than the exception[20,21]. To remedy this, a series of intermediate distributions can be introduced to increase the overlap between adjacent states[1]. These intermediates are termed "alchemical intermediates," as these distributions define non-physical entities: fictitious molecules whose potential functions are defined as a mixture between those of the two real, physical endpoints. A collection of alchemical intermediates connecting two distributions of interest is called an alchemical path. The most widespread alchemical intermediate generation scheme is λ-scaling[1,8]:

$$U_\lambda(\boldsymbol{x}, \lambda) = (1-\lambda)U_1(\boldsymbol{x}) + \lambda U_2(\boldsymbol{x}) \tag{2.5}$$

λ varies between 0 and 1, allowing for a range of potential functions from those defining the real systems U_1 or U_2 for λ equals 0 and 1 respectively, to some intermediate, alchemical system when λ takes any other value. The unnormalized density is given by:

$$q_\lambda(\boldsymbol{x}, \lambda) = \exp(-\beta U_\lambda(\boldsymbol{x}, \lambda)) \tag{2.6}$$

The introduction of intermediate distributions provides a way to approach ratio estimation with a divide and conquer strategy. Until the phase space overlap between adjacent states is sufficient for reliable ratio estimation, alchemical intermediates can be introduced to further improve overlap and simplify the estimation problem. The result of a single division is illustrated below:

$$\begin{aligned}\Delta G = G_2 - G_1 &= (G_2 - G_i) + (G_i - G_1)\\ &= -\beta^{-1}\log(Z_2/Z_i) - \beta^{-1}\log(Z_i/Z_1)\\ &= -\beta^{-1}\log\left(\frac{Z_2}{Z_i} * \frac{Z_i}{Z_1}\right)\\ &= -\beta^{-1}\log\left(\frac{Z_2}{Z_1}\right)\end{aligned} \tag{2.7}$$

where G_i and Z_i respectively represent the free energy and partition function of an alchemical intermediate, i. For an arbitrary number of intermediates, the telescoping product form in the penultimate line of (2.7) remains equivalent to the desired ratio in the final line.

2.3 Alchemical intermediate selection

Careful alchemical intermediate selection has been the focus of several research efforts in the chemical physics literature[22–25], but a widely accepted and used method

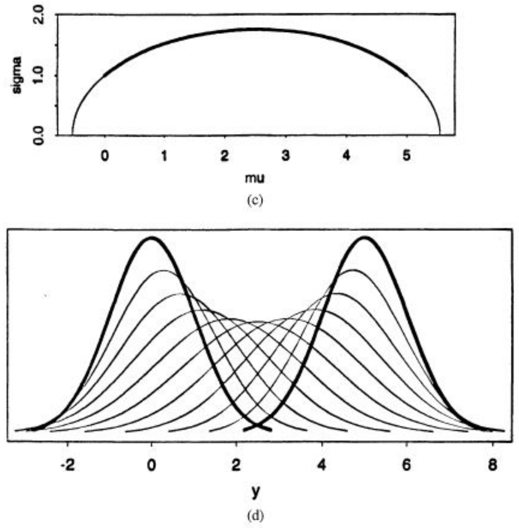

FIGURE 2.2: Figure from Gelman and Meng (1998). Optimal path from $\mathcal{N}(0,1)$ to $\mathcal{N}(5,1)$ in (μ,σ) space. Top panel: Parameter representation. Bottom panel: Density representation for selected intermediates.

for intermediate selection is yet to emerge[21]. For the alchemical intermediate generation scheme defined in equation (2.5), there are two fundamental choices to be made: how many alchemical intermediates to generate, and which λ values they will take. Suggested methods for intermediate selection range from specific Gaussian quadrature rules[26], to loose guidelines recommending a two step process[1], to dynamic λ variation in slow growth methods[27]. For one dimensional path sampling, Gelman and Meng[18] derive an expression for the optimal prior density.

In the same study, Gelman and Meng reported an even more striking result, shown in figure 2.2. For a transition between normal distributions with continuous paths, by expanding the distribution space to allow both the mean and variance to vary, as opposed to only the mean, a new minimum variance path was found that outperformed the optimal direct one dimensional solution. For alchemical intermediate selection, multivariate representations of the potential are analogous to this expansion of the distribution space. The intermediate selection problem becomes more difficult, but with a larger potential payoff.

3

Optimal Path Determination by Exhaustive Search

3.1 Defining an augmented λ space

In order to ascertain the impact of Gelman and Meng's result on alchemical intermediate selection for free energy calculations, we must first define in which way to expand the alchemical state space defined by equation (2.5). Numerous options exist, including separation of energy terms (Jiang & Roux[28]) and the addition of a biasing potential (Darve et al.[29]), but the most natural choice is to expand the alchemical space using a temperature parameter (Sugita et al.[30]).

Using temperature as a second dimension in the alchemical state space confers numerous and predictable benefits, both practically and technically. Varying temperature in the alchemical space confers similar benefits to replica exchange molecular dynamics (REMD), an expanded ensemble sampling method designed to overcome energetic barriers[30], and because temperature is a standard tunable parameter in molecular simulation packages, sampling states in this expanded space comes at no additional cost, implementation-wise.

As defined, the potential energy is not dependent on temperature, but the unnormalized density is affected:

$$q_{\lambda,\beta}(\boldsymbol{x},\lambda,\beta) = \exp(-\beta U_{\lambda}(\boldsymbol{x},\lambda)) \tag{3.1}$$

Consequently, the partition function $Z_{\lambda,\beta}$ will also depend on the temperature.

While temperature is a convenient thermodynamic parameter for our purposes, letting temperature vary introduces some issues as well. Free energy differences between two states can be expressed as the log ratio of partition functions as in (2.1) and (2.7) only when the two states are at the same temperature. This imposes the constraint that the end points of our alchemical path, the real physical systems of interest, must be at the same temperature. What must be verified is that intermediate alchemical states can pass through regions of varying temperature while still recovering the telescoping product in (2.7). Revisiting (2.7) when the temperature of the intermediate is given by β_i:

$$\Delta G = (G_2 - G_i) + (G_i - G_1) = [-\beta^{-1}\log(Z_1) + \beta_i^{-1}\log(Z_i)] + [-\beta_i^{-1}\log(Z_i) + \beta^{-1}\log(Z_0)] \tag{3.2}$$

It clear that we cannot estimate this free energy difference using BAR type methods if $\beta_i \neq \beta$. To resolve this problem, we can define G_i^* as scaled free energies for intermediate states:

$$G_i^* = \beta_i/\beta * G_i = \beta_i/\beta * (-\beta_i^{-1}\log(Z_i)) = -\beta^{-1}\log(Z_i) \tag{3.3}$$

Replacing G_i with G_i^* in (3.2) will recover the telescoping product, as in (2.7).

It is therefore important to note that in this expanded alchemical space, we are not computing sums of "correct" free energies along the alchemical path, but "improper" free energies. Only the sum of free energies along a complete path is physically meaningful. Nevertheless, this expanded space has been shown to be well suited to relative free energy calculations using BAR, since these improper free

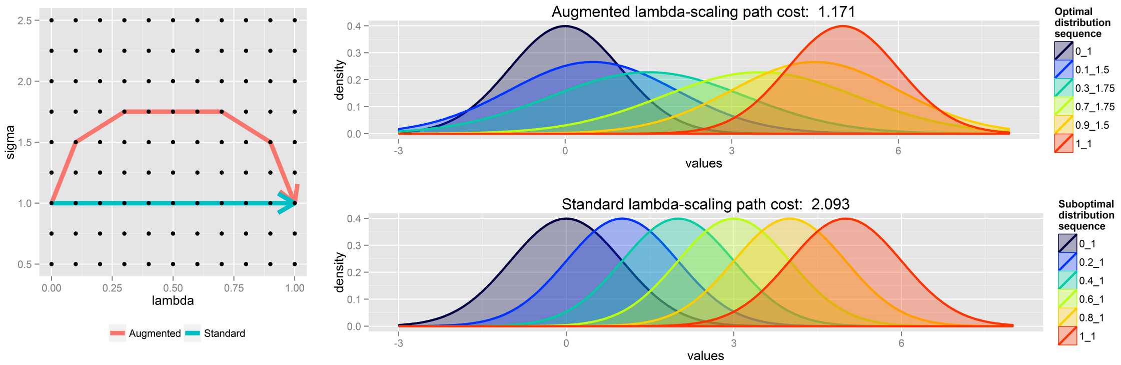

FIGURE 3.1: Dijkstra search on total variation distance for offset harmonic wells. Left panel: Parameter representation of paths. Optimal path for temperature augmented λ-space in red. Optimal path for standard, one dimensional, λ space in blue. Right panel: Density representation of paths. Top panel: distributions along optimal path flatten with increased temperature. Path cost of 1.171. Bottom: distributions along standard path simply mean shift. Path cost of 2.093.

energies allow for the estimation of a telescoping product of normalizing constants, which are unaffected by the scaling in (3.3).

3.2 Revisiting Gelman and Meng in discrete state spaces

To demonstrate the existence of better paths in our augmented (λ, T) space, we sought out to recreate Gelman and Meng's result in our discrete state space. We can define a normal distribution in our scheme by creating a harmonic well, with $U(x) = (x - a)^2$, where $a = \mu$, and the temperature and variance are related as $T \propto \sigma^2$. We set our endpoint states to be $p_0 \sim \mathcal{N}(0, 1)$ and $p_1 \sim \mathcal{N}(5, 1)$.

By using Dijkstra's algorithm[31], we can exactly determine the minimum cost path connecting our λ_0 and λ_1 states. In accordance with the qualitative interpretation of (2.2), total variation distance was used as a measure of the overlap between two distributions. The optimal path maximizes the overlap between adjacent distributions, and minimizes the sum of total variation distances along the path.

The result of the search are shown in figure 3.1. We show that in discrete spaces,

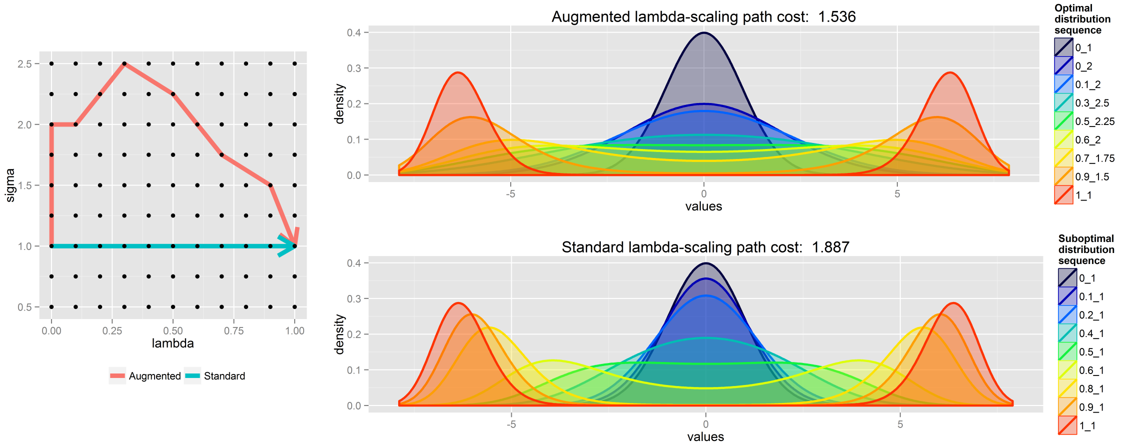

FIGURE 3.2: Dijkstra search on total variance distance for the normal to bimodal transformation. Panels mirror those from figure 3.1. Optimal path cost: 1.536. Standard path cost: 1.887.

the Gelman and Meng result holds. An ellipsoid path which raises the temperature for intermediate distributions is the overall optimal path. These results confirm that not only is the standard path suboptimal, but that it is direly so. The path cost of the best standard solution is 2.093, which is nearly a two-fold increase from the globally optimal solution of 1.171.

To confirm the general utility of the augmented λ space, we repeated the above search for a different target distribution, representing an increase in complexity. We set $U_1(x) \propto (ax^4 - bx^2)$, where a and b are arbitrary scalars, to create a bimodal distribution. The result of this search, shown in figure 3.2, is consistent with that from figure 3.1. The general shape of the optimal path appears to be somewhat conserved: high temperature regions are leveraged to increase distributional overlap.

3.3 Moving towards free energy calculations

3.3.1 A proper distance metric - varBAR

In the previous section, we used total variation distance as edge weights in the graph search. While informative, this does not accurately reflect the type of information

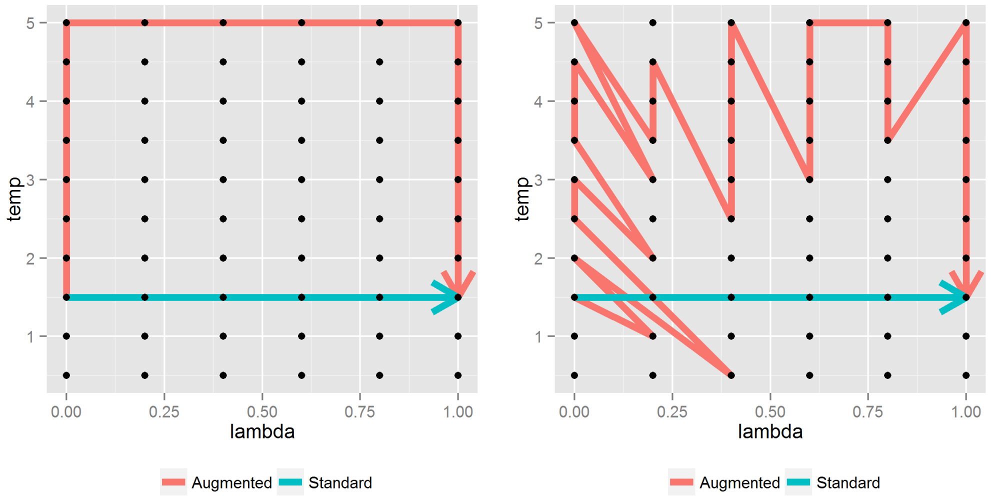

FIGURE 3.3: Parameter representation of paths for Dijkstra search on varBAR distance for offset harmonic wells (left) and normal to bimodal transformation (right). See text for path costs.

available for macromolecular systems. As the goal of a free energy calculation is to minimize the variance of the free energy estimate, a more appropriate distance metric is exactly that: the variance of the BAR estimate between adjacent states (hereafter the varBAR distance). For the BAR estimator, an expression for the asymptotic variance of the free energy difference (or equivalently, the relative variance of the ratio estimate) was derived by Shirts and his colleagues[14,15]:

$$\mathrm{var}(\Delta\hat{f}) = \frac{\mathrm{var}(\hat{r})}{r^2} = \frac{1}{N}\left[\left\langle \frac{1}{2 + 2\cosh(\Delta\hat{f} - \Delta u(x) - M)} \right\rangle^{-1} - \left(\frac{N}{N_2} + \frac{N}{N_1}\right)\right] \tag{3.4}$$

where $M = \ln N_1/N_2$. Empirical tests show that this expression returns accurate variance estimates even for limited sample sizes.

Figure 3.3 summarizes the search results using this new objective function for the previously examined test cases. For the normal to normal case, the optimal path cost is 5.13E-4, and the standard path cost is 1.58E-3. For the normal to bimodal case, the optimal path cost is 3.93E-4, and the standard path cost is 1.81E-3. While the variance is substantially reduced in both cases (approximately 70% and 80% reduction in variance respectively), we note that both optimal paths contain many intermediate states. In a molecular simulation context, these paths would incur a much higher sampling cost, due to the number of intermediate states requiring simulation. Indeed, if we add intermediates to the standard path to match the number of intermediates in the optimal paths, the benefit of the augmented space becomes unclear. The standard path costs for the normal to normal and normal to bimodal cases become 3.88E-4 and 2.18E-4, respectively.

3.3.2 Cost adjusted paths

In order to make sound comparisons, we must put a constraint on the total amount of computation, or equivalently, fix the total number of drawn samples across all paths being compared. In an equilibrium sampling context, depending on the total number of edges in a path, the number of available draws on each constituent state will vary, and consequently the cost of an edge will vary, due to variance scaling with N, as seen in (3.4). This dependence on global properties of the path on individual edge costs breaks the dynamic programming Dijktra's algorithm is built on.

To properly optimize these equal cost paths, we use a modified version of Dijkstra's algorithm, which constrains paths to a fixed length, then carry out a search for a range of possible path lengths. By prespecifying the number of edges in the solution path, we can determine the number of samples per state, and scale all edge costs appropriately for a recursive dynamic programming search.

Figure 3.4 shows the optimal paths when varBAR costs account for the total number of edges in the path. For the offset harmonic well, the standard path cost is 5.07E-4, and the augmented path cost is 3.18E-4. For the unimodal to bimodal case, the standard path cost is 7.03E-5, and the augmented path cost is 5.93E-5. We note

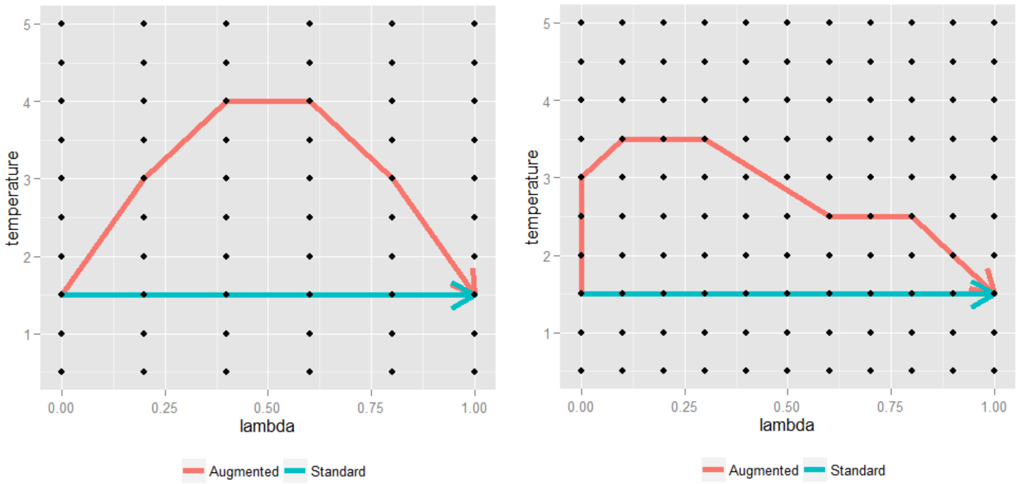

FIGURE 3.4: Parameter representation of paths for cost-adjusted Dijkstra search on varBAR distance for offset harmonic wells (left) and normal to bimodal transformation (right). See text for path costs.

that these corrected paths regain some of the smoothness that we had observed in figures 3.1 and 3.2.

3.4 Scalability of Dijkstra's algorithm to molecular systems

The results from this chapter serve as a proof of concept and establish the validity and theoretical utility of the temperature augmented state space for free energy calculations, however Dijkstra's algorithm is not suitable for path optimization going forward. Dijkstra's algorithm requires knowledge of every edge weight in the graph. Consequently, in order to apply Dijkstra's algorithm, a sufficient number of samples must be drawn from each alchemical state. For molecular systems, this type of exhaustive search is cost prohibitive, and a path searching methodology which balances exploration and exploitation to discover and refine paths online is necessary. We will revisit our options for path selection in chapter 5.

4

Sequential Monte Carlo for Free Energy Estimation

Path optimization in free energy calculations comes in several forms. In chapter 3, we explored the possibility of optimizing the intermediates that make up the path, in order to minimize the variance and decrease the required number of samples needed to reach a desired precision. In this chapter, we will approach optimization from a complementary angle. Given a fixed alchemical path, can we reduce the computational cost per sample, and can we construct an efficient estimator with the minimum variance properties of BAR?

There is considerable overlap between chemical physics studies on free energy calculations and research in statistics for estimating ratios of normalizing constants. Indeed, as we pointed out with the relationship between BAR and bridge sampling, methods from the statistics literature have previously been adapted for use in biophysical problems.

Nonequilibrium methods, such as Jarzynski's method[4], have gone largely underappreciated in the chemical physics literature despite their statistical equivalent, sequential Monte Carlo[5,6] (SMC), enjoying widespread use. A key benefit of SMC methods that we would like to leverage in molecular simulation is the enormous reduction in computational cost for sampling. For equilibrium sampling, assuming a system relaxation time of τ and k alchemical states in the path, to obtain n draws would require $\mathcal{O}(\tau k n)$ computation. On the other hand, for nonequilibrium sampling, we only require $\mathcal{O}(\tau n)$ time to generate initial configurations from one equilibrium simulation, then through a series of cheap nonequilibrium moves, we generate weighted particle trajectories in time $\mathcal{O}((k-1)\tau' n)$, for a total cost of $\mathcal{O}((\tau + (k-1)\tau')n)$, where $\tau >> \tau'$.

4.1 Combining bridge sampling and SMC

The primary downside of Jarzynski's method and its statistical analogue, annealed important sampling[32] (AIS), is that while the sample collection step is rapid, the ratio estimation step is inefficient. This is because these are one-sided estimators, which only use draws from one target state, of which the most well known is Zwanzig's exponential averaging[33]:

$$\Delta G_{0\to 1} = -\beta^{-1} \log E_0[\exp(-\beta(U_1(x) - U_0(x))] \tag{4.1}$$

To remedy this statistical inefficiency, we develop a modified version of BAR, that we call sequential BAR (seqBAR, or sBAR), which uses the BAR estimation step with rapidly generated SMC particles. This fusion of SMC and bridge sampling methods is enabled by a resampling step[34], which transforms weighted SMC particles into a collection of draws approximating the equilibrium distribution for every intermediate state. These resampled particles can then be used directly in BAR estimation as in equation (2.3).

Figure 4.1 demonstrates the gains in statistical efficiency when using seqBAR over AIS. For equal computation, seqBAR is, on average, closer to the true ratio value, and has more consistent ratio estimates, from replicate to replicate. To match the

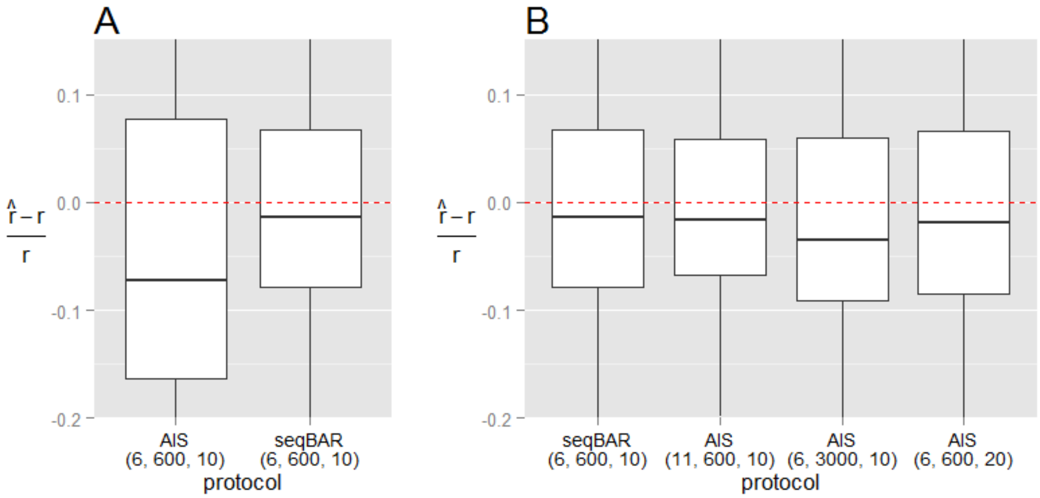

FIGURE 4.1: Comparison of seqBAR and AIS performance for the normal to bimodal transformation. Performance is judged by the empirical accuracy and precision of the relative error of the ratio estimate for 500 independent realizations. Protocols are defined by an estimation method (AIS or seqBAR), followed by a trio of parameters, indicating respectively, the number of distributions in the path, the number of sample trajectories, and the number of Metropolis mixing steps per transition. Left panel: Comparison of AIS and seqBAR performance for equal computation time. Right panel: Required computation for AIS to match seqBAR performance.

performance of seqBAR, AIS requires either a doubling of the number of distributions in the path, a five-fold increase in the number of samples, or a doubling in the cost of each nonequilibrium transition, here a Metropolis[35] mixing step.

4.2 SMC in the augmented (λ, T) space

SMC methods are also sensitive to path choice. For the normal to normal example, five paths in the (λ, T) space were compared using AIS. Ratio estimates, squared errors and estimated variances are shown in figure 4.2. We note that there appears to be a tradeoff with using the higher temperatures. As temperature increases up to a maximum value of 3 (a path height of 2), AIS performance improves. Once temperature increases beyond that point (path heights 3 and 4), the increased temperature introduces too much noise, and the variance of the ratio estimate increases,

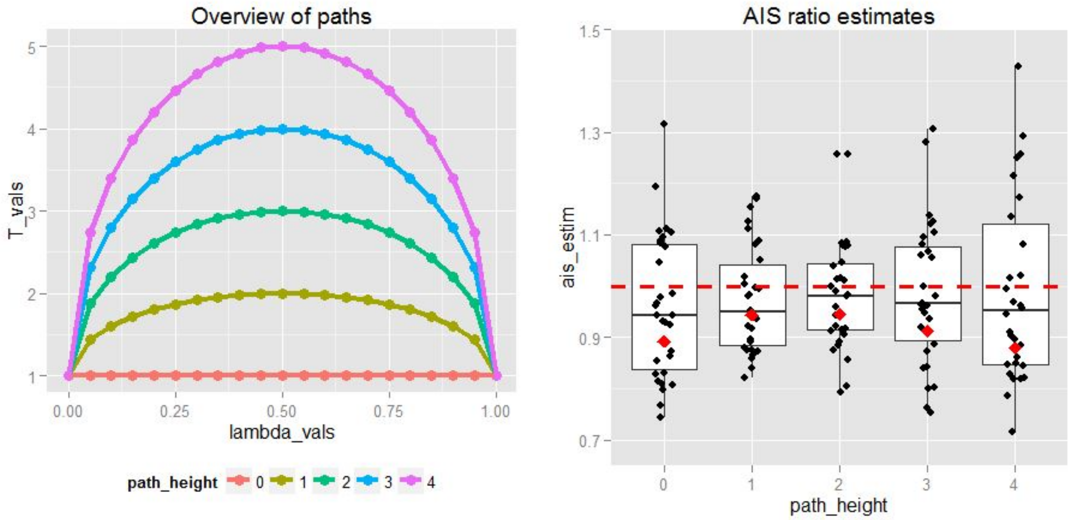

Figure 4.2: Comparison of AIS paths for offset harmonic wells. Left: parameter space representation of paths. Right: ratio estimates indexed by path. The dotted red line denotes the true ratio. Red diamonds represent average values for each path height.

even beyond that of an AIS run using the standard, fixed temperature path.

4.3 pCrooks: a pairwise decomposable CFT

Regardless of path, there are conditions under which AIS and seqBAR can struggle, namely when the target distributions are sufficiently different such that nonequilibrium moves are unable to create particles approximating the final distribution. One such example is the transition from a normal distribution to a Cauchy distribution (t-distribution with one degree of freedom), due to their differing tail decay rates. In the same way that BAR improves on exponential averaging by creating bridge functions and utilizing samples from both target distributions, the Crooks fluctuation theorem[3] (CFT) improves on Jarzynski's method. CFT is a a hybrid SMC-bridge sampling method, which operates on nonequilibrium work distributions generated in both sampling directions. This bidirectional propagation results in increased robustness to varying target distributions for ratio estimation, as seen in figure 4.3. Note

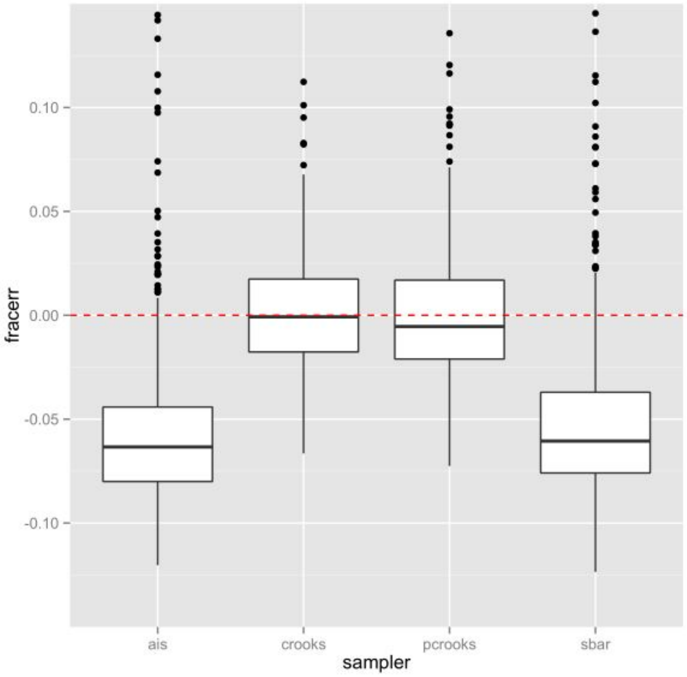

Figure 4.3: Comparison of AIS, CFT, pCrooks and seqBAR ratio estimation performance for the normal to Cauchy distribution transformation.

that both AIS and seqBAR are highly variable, and frequently under estimate the ratio, whereas CFT (crooks) is extremely accurate.

In the context of path selection however, CFT has notable shortcomings. For path selection between enumerated (and frequently non-overlapping) paths, CFT is acceptable, but when trying to optimize paths in a full grid search, when there may be significant barriers to work around, and we have no guesses as to what form an acceptable path may take, the dependence on CFT on full path work distributions hinders our ability to locally optimize transitions and build reasonable paths. To this end, we developed pairwise Crooks (pCrooks), a pairwise decomposable version of CFT. Instead of using full path work distributions to estimate the full telescoping ratio in one step, we calculate each pairwise ratio using BAR with weighted work distributions, where the weights are derived from the accumulated SMC transitions. In figure 4.3, we show that pCrooks corrects the shortcomings of seqBAR, and approaches the effectiveness of CFT. By breaking down the ratio estimation step, we can obtain information on the variance of each transition in the path, allowing us to locally optimize the path, and refine existing paths, by trying other transitions to replace high variance links.

5

Path Selection via Reinforcement Learning

Exhaustive search by Dijkstra's algorithm in chapter 3, and empirical benchmarking in section 4.2 have demonstrated the theoretical utility of an augmented state space. In practice however, the principal challenge is to efficiently navigate the augmented state space to find improved paths without offsetting the gains in efficiency afforded by those paths. We draw on machine learning techniques from the reinforcement learning literature to address this task.

5.1 SMC and the multi-armed bandit

The multi-armed bandit problem[36–38] is an optimization problem originating from probability theory. Selecting between an enumerated set of paths for free energy estimation can be cast as a sample allocation problem, where the conflicting goals of drawing samples for estimation along the putative best path and drawing samples for estimating the quality of other potentially better paths are in opposition. In figure 5.1, we examine our ability to optimize paths with SMC by comparing three bandit strategies.

The "equal" strategy is effectively the null strategy, where each path is sampled

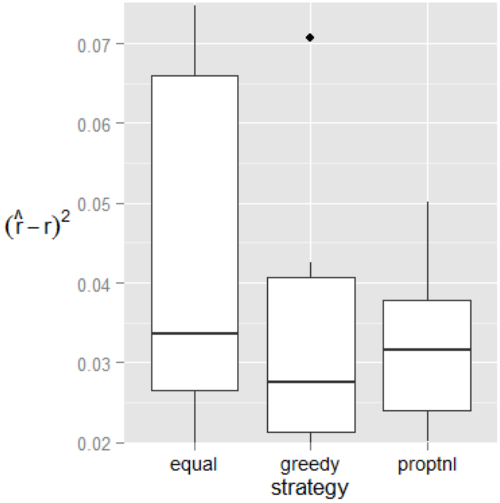

Figure 5.1: Comparison of squared error for three AIS bandit strategies for offset harmonic wells. See text for details on strategies.

equally, regardless of their estimated variances. The “greedy” strategy follows up an initial exploration period of equal sampling with a period where only the minimum estimated variance path is sampled. The “proptnl” strategy follows up an initial exploration period of equal sampling with a period where each path is sampled in quantities inversely proportional to their variances. We observe that the greedy strategy is, on average, the strategy that minimizes the squared error of the ratio estimate, however it is also interesting to note that the proportional strategy is the most consistent one.

A weakness of this form of path selection, as we alluded to in section 4.3, is that it only operates on fully predefined paths. As we saw in figure 3.4, the shape of the optimal path can vary, and as the dimensionality of the underlying distributions increases, it is possible that the optimal paths will become more irregular in order to navigate around energetic barriers. In these situations, where we have no intuition to guide path definition, selection between enumerated paths may not be sufficient for the efficiency gains we seek, and a full grid optimization is required, where a path is built edge by edge.

5.2 Full grid optimization with Q-learning

Q-learning[7] is a reinforcement learning technique for finite state Markov decision processes, typically used to derive an optimal action selection policy for total reward maximization. Q-learning progresses through successive learning episodes, where a Q function describing the expected values of the long term rewards for each state-action pair is inferred via value iteration updates. Search agent actions, and the putative best path are both guided by the current best estimate of the Q function, which is updated as follows after each search episode:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)) \tag{5.1}$$

where $\alpha \in [0,1]$ is the learning rate, $\gamma \in [0,1]$ is a discount factor to penalize delayed rewards, $R(s,a)$ is a function that returns some reward for the action a out of state s, and the maximum is the maximum Q value for all actions out of the resulting state for the observed action.

In our application, as our goal is to minimize the variance of the path, the reward function $R(s,a)$, represents estimated variances for a given transition between states. These variance estimates are given by equation (3.4) for a low cost pCrooks run. Ratio estimates for a given transition from separate learning episodes can be combined as a mean weighted by their inverse variances.

To make Q-learning compatible with a minimization task, we make two important adjustments in order to minimize the accumulated reward. First, we take the minimum Q value over the resulting states, instead of the maximum. Second, γ now takes values of 1 or higher. This second adjustment is required to avoid situations where Q-learning will create infinite loops in the solution path to delay a required

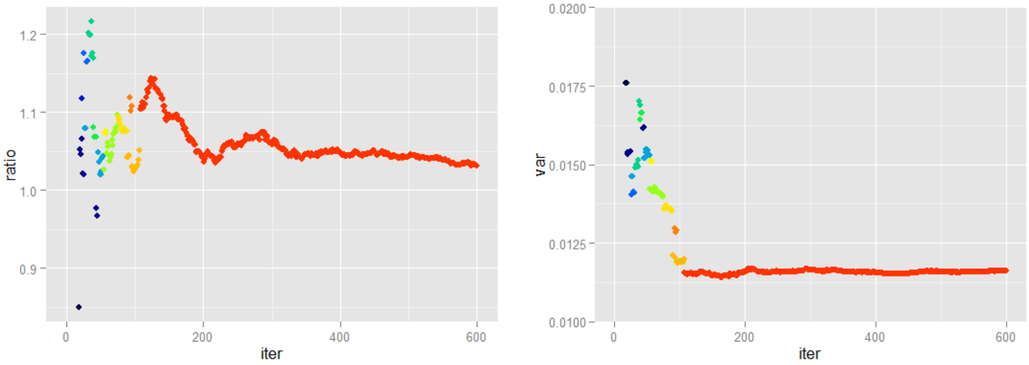

FIGURE 5.2: Ratio and variance estimates monitored for fixed duration Q-learning. Target distributions are offset harmonic wells, $\alpha = 0.8, \gamma = 1$. Changes in point color represent changes in the current best path.

large cost action. In a minimization context, when γ is bounded by 0 and 1, longer paths are favored, as distal costs become discounted. As we've previously seen, longer paths come at a larger computational cost, so we constrain γ to be 1 or larger, to favor shorter paths. γ can be tuned to reflect the relative cost of equilibrium sampling and nonequilibrium SMC transition moves, where values near one represent low cost SMC moves.

5.2.1 Fixed duration Q-learning

In this section, Q-learning is run for a preselected number of learning episodes, representing a fixed ceiling on the amount of available computation time. α is set to 0.8, and γ is set to 1. The search space is the augmented (λ, T) space for the offset harmonic wells.

Figure 5.2 shows the evolution of the ratio estimates, variance estimates, and current best path as a function of the number of learning episodes that have elapsed. For the first 100 steps of the algorithm, the best guess optimal path varies rapidly, as denoted by the changing point colors. As the algorithm progresses, path variance estimates become more precise, and the low cost paths are learned by the Q function, as illustrated by the decreasing variance in the second panel, and the fixation of the

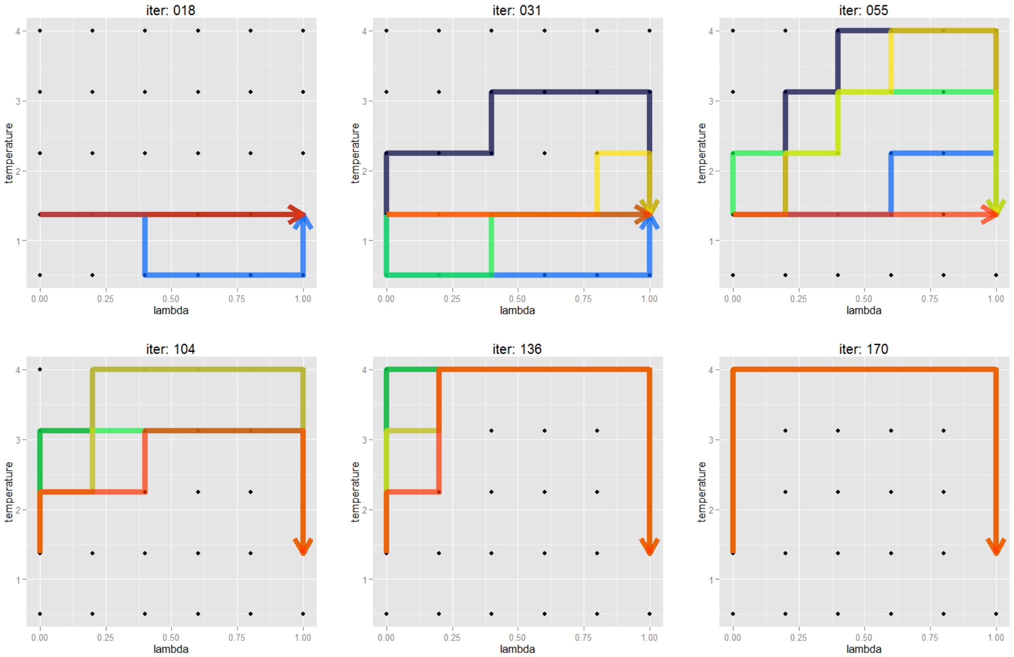

FIGURE 5.3: Time evolution of five Q-learning chains for offset harmonic wells. $\alpha = 0.8, \gamma = 1$. Each panel shows the current best path as determined by independent runs of Q-learning at different time points. Clockwise from top left: 18, 31, 55, 170, 136 and 104 iterations elapsed.

red path as the optimal path. The ratio estimation converges near the true value of 1.

Figure 5.3 depicts the evolution of the best guess path for five replicates of the Q-learning algorithm on this test case. Each color represents one chain's best guess path varying across iterations. At the 18th iteration, only some replicates have determined a complete best path. These paths are direct, and not necessarily low variance. In fact, the path in blue is a higher variance path than the standard, fixed temperature path. At the 31st iteration, all chains have found solution paths, but their forms vary wildly. Some, like the dark blue path, are promising in their use of the higher temperature regions. By the 55th iteration, almost all paths are improved relative to the standard, fixed temperature path. The 104th and 136th iterations show that the chains are beginning to converge on similar solution paths. In the

136th iteration, some have even found the true optimal path we had determined by exhaustive search in figure 3.3. By the 170th iteration, all chains have converged on the true optimal path.

5.2.2 Q-learning convergence

The goal of a free energy calculation is to obtain an estimate of maximum precision in a set amount of time, or an estimate of set precision in a minimum amount of time. A downside of Q-learning, and of path selection algorithms in general, is that it divides sampling efforts among many paths during the search phase, and not all of the generated SMC particles will participate in the ratio estimation if the distribution they approximate does not lie on the putative best path. A benefit of standard λ scaling is that all particles traverse only one path, ensuring full sample utilization, and a full reduction in variance by a factor of N.

In this section, we account for this variance reduction factor, and determine if Q-learning can converge on an improved path fast enough to offset particle waste in the early stages of the algorithm. For both an augmented and standard state space, we run three independent Q-learning chains. For the standard state space, this entails simply running a free energy estimation along the fixed temperature path, as there are no alternative paths to consider. After each iteration, we construct a composite free energy estimate based on the three chain estimates. Within a predefined tolerance level, we assess if the individual chains agree with this composite estimate, based on a chain interval derived from the chain estimate and variance. Chain concordance with the composite estimate provides a heuristic check for free energy estimate convergence.

The results of testing this convergence heuristic are shown in figure 5.4. Despite the computational overhead Q-learning in the augmented state space incurs, it still converges significantly faster and at a more consistent rate than free energy

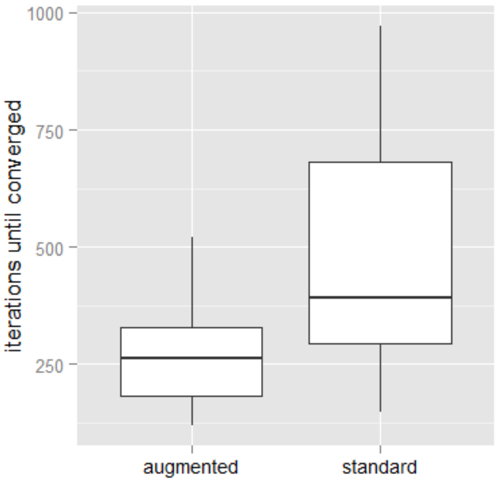

FIGURE 5.4: Rates of Q-learning convergence for offset harmonic wells. $\alpha = 0.8, \gamma = 1$. For 30 replicates each, convergence rates for Q-learning in the temperature-augmented and standard state spaces are compared. The free energy estimates converge signficantly faster with path searching enabled via Q-learning, relative to when path searching is disabled, with the standard state space.

estimation along the standard path. On average, when path searching is permitted, convergence is achieved 1.8 times faster than when free energy estimation is confined to a fixed temperature path. This result shows that path selection by Q-learning, in conjunction with free energy estimation with pCrooks is a valid approach for practical path optimization.

6

Discussion

Multivariate representations of state spaces have enjoyed great success for sampling, both in a molecular dynamics context with replica-exchange molecular dynamics[30], and in a statistical context with hybrid Monte Carlo[39]. In this work, we have motivated an analogous extension to free energy calculations. Inspired by the work of Gelman and Meng[18], we showed that the addition of a temperature parameter to the λ scaling alchemical generation scheme allows for the creation of new, reduced variance alchemical paths. Furthermore, we found that Q-learning[7], a reinforcement learning technique, could be successfully adapted for use as an algorithm for simultaneous path optimization and free energy estimation.

In parallel work, we expanded on the work of Jarzynski[4] and Crooks[3] to develop a new nonequilibrium sampler and free energy calculation method, which we call pCrooks. pCrooks is fundamentally a bridge sampling method, like CFT, but unlike CFT, pCrooks provides information on the variance for each pair of adjacent states in an alchemical path, making it a method that synergizes well with path optimization algorithms.

This document serves as a proof of concept for these methods. For simple models, where exhaustive or analytical methods are feasible, we determined reference values against which to test our methods, and showed that we were able to efficiently recover these solutions with our methodology. The combined algorithm with Q-learning for path optimization and pCrooks as a subroutine for sampling and free energy estimation represents the culmination of this work. In a head to head comparison, our path optimizing Q-learning algorithm was able to converge on a free energy estimate at significantly lower computational cost than the fixed path alternative.

This result is notable because it demonstrates the practical applicability of path optimization in free energy calculations, even in situations where little is known about the energy landscape of the alchemical state space. For large molecular systems, where the standard path may run into wide regions of poor overlap, characterized by significant energy barriers, this path optimization may provide a method for avoiding these difficult and problematic transitions.

Moving forward, to fully realize the potential of this method, further testing must be done to determine if the path searching will remain efficient for more complex distributions. Reasonable next test cases range from those where a known barrier exists, such as in a ferromagnetic Ising model[40], to well studied examples from the free energy calculation literature, such as the free energy of solvation of methane[41], or relative binding free energies to T4 lysozyme[42].

To deal with the task of sampling high dimensional energy functions, the SMC sampling protocol may require additional improvements. A bidirectional sampling method, like pCrooks, can be combined with a resampling step as is used in seqBAR to alleviate issues related to minimal density overlap. Further expansions to the alchemical state space should be considered with caution. As the dimensionality of the state space grows, the overhead associated with Q-learning's path searching will increase rapidly.

In summary, we hope that this work will represent the beginning of a paradigm shift in alchemical intermediate selection away from unprincipled, rule of thumb methods, and towards heuristic methods which take into account the properties of the underlying state space determined by low cost pilot simulations.

Appendix A

Detailed Methods

This appendix contains detailed method descriptions and parameter values required to reproduce the work shown in this thesis.
Code is available at `https://github.com/rmuraglia/Schmidler`.

A.1 Exhaustive search

A.1.1 Dijkstra's algorithm

Dijkstra's algorithm is a dynamic programming algorithm for finding minimum cost paths through graphs.

Technically, because we are only interested in the shortest path between a given pair of nodes, we can terminate the algorithm when the selected u in line 5 is the target node. To reconstruct a solution path, we can simply trace our way backwards. Starting with the target node, we simply need to note the *node.prev* value, and keep on iterating backwards until we reach the initial node, with a *node.prev* value of NA.

For figure 3.1, the initial and target distributions were $\mathcal{N}(0, 1)$ and $\mathcal{N}(5, 1)$ distributions, respectively. More specifically, they were defined with the potential function

Algorithm 1 Dijkstra's algorithm

```
Initialize queue $Q$ containing all nodes of the graph.
Set node attributes as: $node.dist = \infty, node.prev =$NA, $\forall$ nodes.
For node corresponding to initial state, set $node.dist = 0$.
while $Q$ not empty do
    Pop node with minimum $node.dist$ from $Q$. Set as $u$.
    Set $V$ as the list of neighbor nodes to $u$.
    for $v$ in $V$ do
        Calculate the distance to $v$ via $u$ as $v.alt = u.dist + d(u, v)$.
        if $v.alt < v.dist$ then
            This is a new shortest path to $v$.
            Set $v.dist = v.alt$ and $v.prev = u$
        end if
    end for
end while
```

$U(x) = \frac{(x-\mu)^2}{2\sigma^2}$, and the unnormalized density $q(x) = \exp(-\beta U(x))$. The initial and target state had (μ, σ) pairs $(0, 1), (5, 1)$, respectively. Intermediate distributions were defined by λ-scaling, as shown in equation (2.5).

For figure 3.2, the initial distribution was the same standard normal, but the target distribution was a bimodal distribution, defined by a quartic potential $U(x) = (ax^4 - bx^2)/(\sigma^2 c)$. Here, $a = 1, b = 81, c = 150, \sigma = 1$.

For both cases, the distance metric was the total variation distance, as defined by Roberts and Rosenthal[43]: $d_{TV}(p_1, p_2) = E_1 \left[\min\left(1, \frac{p_2(x)}{p_1(x)}\right)\right]$. The search parameters used were 1000 draws per node, and a *move.jump* coefficient of 4.

For figure 3.3, the distributions were defined as they were for figures 3.1 and 3.2. Here temperature varied in the grid instead of σ. σ was held constant at 1, but β, the inverse temperature varied according to the grid coordinates. The distance metric was the asymptotic variance of the BAR estimator, as defined in equation (3.4). In both cases, there were 1000 draws per node and a *move.jump* coefficient of 5.

A.1.2 Fixed path length search

In equation (3.4), there is a leading $1/N$ variance reduction term based on the number of samples drawn at each node. As a result, when 1000 draws are sampled from each state, paths with more edges benefit from this leading term more. In other words, depending on path length, some paths can be thought of as more computationally costly. In figure 3.4 we compare equal computation paths by fixing the total number of samples drawn per path. To do this, we carry out a dynamic programming search, similar to Dijkstra's algorithm, which returns the shortest path of specified length.

Algorithm 2 Fixed path length search

```
Initialize $C$ matrix values to $\infty$, where $C[i,j]$ represents the cost of the cheapest
  path from the initial state to state $i$ in $j$ steps.
Initialize $C[init, 1] = 0$.
Initialize $P$ matrix values to NA, where $P[i,j]$ represents the previous state
  corresponding to the move that gave rise to $C[i,j]$.
for $k$ in $2 : maxlength$ do
    Set $Q'$ as the states with a complete path in $k-1$ steps ($C[,k-1] \neq \infty$)
    Set $Q$ as the neighbors of $Q'$. These are the nodes to be queried.
    for $q$ in $Q$ do
        set $V$ as the list of neighbor nodes to $q$.
        for $v$ in $V$ do
            Calculate the distance to $q$ via $v$ in $k$ steps as $q.alt = C[v,k-1] + d(v,q)$.

            if $q.alt < C[q,k]$ then
                This is a new shorter path to $q$.
                Set $C[q,k] = q.alt$ and $P[q,k] = v$.
            end if
        end for
    end for
end for
```

At the completion of the algorithm, we can then directly read off the unscaled path costs, constrained to a path length, which can then be corrected to account for the number of edges in the path. Obtaining the solution path is done in the same way as for Dijkstra's algorithm. We simply need to trace back values in the P matrix.

For the harmonic wells, the end point distributions were the same as previously. Each state was sampled from 5000 times to obtain the unscaled edge cost, which was then scaled with an effective sample size. The *move.jump* coefficient was 4.

For the unimodal to bimodal case, each state was sampled from 1000 times, and the *move.jump* coefficient was 10. The initial normal distribution state was the same as before, but the bimodal distribution now has coefficients: $a = 0.5, b = 14, c = 64$.

A.2 Sequential Monte Carlo

In this document, we presented four sequential Monte Carlo based free energy estimators: annealed importance sampling (AIS, equivalent to Jarzynski's method), sequential BAR (seqBAR, or sBAR), Crooks (an estimator which follows from the Crooks fluctuation theorem), and pairwise Crooks (pCrooks). Each method can be decomposed into two steps: nonequilibrium sampling and ratio estimation. When transitions are carried out as symmetric Metropolis moves, the nonequilibrium sampling protocol is as follows:

Algorithm 3 A nonequilibrium sampler with Metropolis transitions

```
Define a series of distributions $\{p_1, ..., p_n\}$
Generate $N$ independent draws $\{x_1^{(i)}\}$ from $p_1$
Initialize weights: $\{w_1^{(i)}\} = 1/N$
for $j = 2$ to $n$ do
    Calculate incremental weights: $\{\tilde{w}_{j-1}\} = q_j(x_{j-1})/q_{j-1}(x_{j-1})$
    if RESAMPLE then
        Generate $\{\tilde{x}_{j-1}\}$ by sampling $\{x_{j-1}\}$ with replacement with probabilities $\{\tilde{w}_{j-1}\}$
        Reset weights: $\{w_{j-1}\} = 1/N$
    else
        Set $\{\tilde{x}_{j-1}\}$ to $\{x_{j-1}\}$
    end if
    Normalize weights: $w_j = (w_{j-1} * \tilde{w}_{j-1})/(\sum w_{j-1} * \tilde{w}_{j-1})$
    Carry out a nonequilibrium transition
    for $i = 1$ to $N$ do
        Propagate each particle to the next distribution
        $x_j^{(i)} \sim Metropolis(\tilde{x}_{j-1}^{(i)}, q_j)$
    end for
end for
```

In this work, the Metropolis transitions consisted of accept/reject steps for trial configurations generated by drawing from a Gaussian centered on the current state with scale 0.5. The "RESAMPLE" flag is always set to "FALSE" for AIS, Crooks and pCrooks. For seqBAR, the flag is always set to "TRUE." In general usage, we would resample when the effective sample size criterion dips below a prespecified threshold, such as suggested by Del Moral, Doucet and Jasra[5]. For AIS, we only run the sampler in one trajectory generation direction. For seqBAR, we may run it either only on direction, or we can combine information from both propagation directions. For Crooks and pCrooks, the sampler must be run in both directions (1 to n and n to 1).

For AIS, the ratio estimation step is simple. First, calculate full trajectory particle weights by multiplying incremental weights for each particle, then obtain a ratio estimate by taking the mean of the full trajectory particle weights.

For seqBAR, because we resample at each step, our draws approximate the equilibrium distributions. We can then directly apply BAR estimation to these draws as described in equation (2.3) and Meng and Schilling[17].

As previously noted, there is significant overlap in these methods. For the Crooks estimator, we directly apply equation (19) from Crooks[3] with the same type of iterative scheme as we do for BAR. Converting between work and ratios of densities, or free energies and ratios of normalizing constants is trivial and left to the reader.

Lastly, the pCrooks estimator is nearly identical to the BAR estimator. When precomputing l and s terms, as described in Meng and Schilling's equation, we must take care to use the correct draws, such that when weighted, they approximate the desired distribution. In general, to approximate the "forward" draws for the jth distribution, use $w_j x_{j-1}$, and for the "reverse" draws for the jth distribution, use $w_j x_{j+1}$. For the first and last distribution edge cases, simply use x_1 and x_n weighted by uninformative weight of $1/N$. In plain terms, we use the unweighted draws specified in the previous sentence to calculate l and s, then use the normalized weights as a leading correction term in the sum.

A.2.1 Specific parameter values

To create the figures, the following run parameters for the SMC algorithms were used:

For figure 4.1, the initial distribution was $\mathcal{N}(0, 1)$, and the target distribution was a bimodal with parameters $a = 0.5, b = 14, c = 64$. All trajectories were propagated in the normal to bimodal direction. The temperature was held fixed.

For figure 4.2, the initial and target distributions were $\mathcal{N}(0, 1)$ and $\mathcal{N}(5, 1)$ distributions, respectively. Each Metropolis transition has 25 accept/reject steps. Each path had 21 λ points, which were spaced uniformly along the λ coordinate. Their temperature values were chosen to lie along a half-ellipse defined as: $T(\lambda) = \sqrt{h^2 - h^2/c^2 * (\lambda - c)^2} + 1$, where c indicates the center of the ellipse (here $c = 0.5$), and h indicates the maximum height of the path.

For figure 4.3, the initial distribution was $\mathcal{N}(0, 1)$, and the target distribution was a t-distribution with one degree of freedom (a Cauchy distribution). Each algorithm was run with six distributions in the path, 600 particles per ratio estimate, and ten mixing steps per transition. For Crooks, pCrooks and sBAR, trajectories were evenly split between forward and reverse propagation. For AIS all trajectories were run in the forward direction. The temperature was held fixed for all paths.

A.3 Reinforcement learning

A.3.1 AIS bandit

For figure 5.1, each AIS run had 200 particles, with 25 mixing steps per transition. The bandit strategy was selecting between three paths, each with 26 distributions per path. The three paths were defined with the same semi-ellipse scheme as above, with maximum heights of 0, 2 and 4. All particles were propagated from the $\mathcal{N}(0, 1)$ to the $\mathcal{N}(5, 1)$ distribution. Each strategy ran 90 separate ratio estimates. Ten were allocated to each path to get an estimate of the path quality (the variance of the ratio estimates). The remaining 60 were allocated as described in the main body of the text.

A.3.2 Q-learning

Here we present complete Q-learning pseudocode to illustrate our method combined with pCrooks and stochastic edge weight evaluation.

Algorithm 4 Q-learning pseudocode

```
Initialize all $Q(s, a)$ values to 1
while Stop condition not met do
    Run a Q-learning episode
    Propagate particles forward
    Set $s$ to initial distribution
    Generate draws from $s$
    while $s \neq$ target state do
        Select an outbound action ($a$) from state ($s$)
        Propagate particles from $s$ to resultant state of action $a$
        Obtain SMC weights associated with transition
        Set $s$ as resultant state of action $a$
    end while
    Propagate particles backwards along same path
    Set $s$ to target distribution
    Generate draws from $s$
    while $s \neq$ initial state do
        Propagate particles from $s$ to previous state in path
        Obtain SMC weights
        Set $s$ as previous state in path
    end while
    Calculate pCrooks ratio and var(ratio) based on forward draws, forward
    weights, reverse draws and reverse weights
    Add var(ratio) to reward table entry for each (state, action) pair in path
    Update Q map according to equation (5.1)
end while
```

The action selection move and stopping criteria vary, depending on the type of Q-learning run. Different options are discussed below. To get the current best-path, we simply trace the minimum outbound Q value, starting from the initial state.

For figure 5.2, additional parameter values not discussed in the main text are as follows. The initial and target distributions are $\mathcal{N}(0, 0.7^2)$ and $\mathcal{N}(5, 0.7^2)$, respectively. The *move.jump* coefficient for the search was 1. Each search agent calculated ratios with pCrooks with 100 particles. At each transition, there were 25 Metropolis mixing steps. The Q map values were initialized to a value of 1. Next state selection was based on two factors: an ϵ value representing the impact of randomness on the search, and the Q values for potential next moves. With probability ϵ, we simply selected the minimum Q move, otherwise we selected from the next possible moves randomly. We set ϵ to vary linearly, starting from a value of 0, and ending with a value of 1, once 85% of the learning episodes were complete. The stopping criteria was when 600 Q-learning episodes had elapsed, for this fixed duration Q-learning.

For figure 5.4, here we present additional details on the convergence method. We run three separate Q-learning chains, which maintain their own independent Q and reward maps. For a *min.episode* number of episodes (here 30), they run independently, searching randomly ($\epsilon = 0$). Following this initial learning period, after every episode, we assess the convergence of each chain.

To do this, we construct a composite estimate, based on a mean of the three chain ratio estimates, weighted inversely by their variances. We then construct a window around this composite estimate as $estimate \pm estimate * conv.tol$ (here $conv.tol = 0.025$). For each chain, we construct an interval as the chain estimate ± the chain standard deviation. If the entire chain interval is contained within the composite window, we claim that chain to be converged. When all three chains are converged, the Q-learning algorithm is deemed converged.

After the initial learning period, the ϵ values are tuned starting from a value of 0.3, according to each chain's converged status. If the chain is converged, we increase ϵ by $\Delta\epsilon$ (here set to 0.005), with an ϵ cap of 1. If the chain is not converged, we aim for a steady state ϵ value of 0.8. One way to accomplish this stochastically is provided in the code.

Bibliography

[1] Michael R Shirts, David L Mobley, and John D Chodera. Alchemical free energy calculations: ready for prime time. *Annual Reports in Computational Chemistry*, 3:41–59, 2007.

[2] Andrew Pohorille, Christopher Jarzynski, and Christophe Chipot. Good practices in free-energy calculations. *The Journal of Physical Chemistry B*, 114(32):10235–10253, 2010.

[3] Gavin E Crooks. Path-ensemble averages in systems driven far from equilibrium. *Physical review E*, 61(3):2361, 2000.

[4] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690, 1997.

[5] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[6] Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[7] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[8] Clara D Christ, Alan E Mark, and Wilfred F van Gunsteren. Basic ingredients of free energy calculations: a review. *Journal of computational chemistry*, 31(8):1569–1582, 2010.

[9] K Vandivort, JC Phillips, E Villa, PL Freddolino, J Gumbart, LG Trabuco, DE Chandler, J Hsin, CB Harrison, L Kale, et al. Long time and large size molecular dynamics simulations made feasible through new TeraGrid hardware and software. In *Proceedings of the 2008 TeraGrid Conference*, 2008.

[10] Romelia Salomon-Ferrer, Andreas W Gotz, Duncan Poole, Scott Le Grand, and Ross C Walker. Routine microsecond molecular dynamics simulations with

Amber on GPUs. 2. Explicit solvent particle mesh Ewald. *Journal of Chemical Theory and Computation*, 9(9):3878–3888, 2013.

[11] Matthew J Harvey and Gianni De Fabritiis. High-throughput molecular dynamics: the powerful new tool for drug discovery. *Drug discovery today*, 17(19):1059–1062, 2012.

[12] Jacob D Durrant and J Andrew McCammon. Molecular dynamics simulations and drug discovery. *BMC biology*, 9(1):71, 2011.

[13] Charles H Bennett. Efficient estimation of free energy differences from Monte Carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976.

[14] Michael R Shirts, Eric Bair, Giles Hooker, and Vijay S Pande. Equilibrium free energies from nonequilibrium measurements using maximum-likelihood methods. *Physical review letters*, 91(14):140601, 2003.

[15] 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.

[16] Michael R Shirts and Vijay S Pande. Comparison of efficiency and bias of free energies computed by exponential averaging, the Bennett acceptance ratio, and thermodynamic integration. *The Journal of chemical physics*, 122(14):144107, 2005.

[17] Xiao-Li Meng and Stephen Schilling. Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586, 2002.

[18] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.

[19] Xiao-Li Meng and Wing Hung Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, 6(4):831–860, 1996.

[20] Di Wu and David A Kofke. Phase-space overlap measures. I. Fail-safe bias detection in free energies calculated by molecular simulation. *The Journal of chemical physics*, 123(5):054103, 2005.

[21] Di Wu and David A Kofke. Phase-space overlap measures. II. Design and implementation of staging methods for free-energy calculations. *The Journal of chemical physics*, 123(8):084109, 2005.

[22] Tri T Pham and Michael R Shirts. Identifying low variance pathways for free energy calculations of molecular transformations in solution phase. *The Journal of chemical physics*, 135(3):034114, 2011.

[23] Nandou Lu and David A Kofke. Optimal intermediates in staged free energy calculations. *The Journal of chemical physics*, 111(10):4414–4423, 1999.

[24] Arnaud Blondel. Ensemble variance in free energy calculations by thermodynamic integration: theory, optimal alchemical path, and practical solutions. *Journal of computational chemistry*, 25(7):985–993, 2004.

[25] Haluk Resat and Mihaly Mezei. Studies on free energy calculations. I. Thermodynamic integration using a polynomial path. *The Journal of chemical physics*, 99(8):6052–6061, 1993.

[26] *AMBER 14 reference manual*, 2014.

[27] David A Pearlman and Peter A Kollman. A new method for carrying out free energy perturbation calculations: dynamically modified windows. *The Journal of Chemical Physics*, 90(4):2460–2470, 1989.

[28] Wei Jiang and Benoît Roux. Free energy perturbation hamiltonian replica-exchange molecular dynamics (FEP/H-REMD) for absolute ligand binding free energy calculations. *Journal of chemical theory and computation*, 6(9):2559–2565, 2010.

[29] Eric Darve, David Rodríguez-Gómez, and Andrew Pohorille. Adaptive biasing force method for scalar and vector free energy calculations. *The Journal of chemical physics*, 128(14):144120, 2008.

[30] Yuji Sugita, Akio Kitao, and Yuko Okamoto. Multidimensional replica-exchange method for free-energy calculations. *The Journal of Chemical Physics*, 113(15):6042–6051, 2000.

[31] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[32] Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[33] Robert W Zwanzig. High-temperature equation of state by a perturbation method. i. nonpolar gases. *The Journal of Chemical Physics*, 22(8):1420–1426, 1954.

[34] Lawrence Murray. Gpu acceleration of the particle filter: the metropolis resampler. *arXiv preprint arXiv:1202.6163*, 2012.

[35] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[36] Michel Tokic and Günther Palm. Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In *KI 2011: Advances in Artificial Intelligence*, pages 335–346. Springer, 2011.

[37] Steven L Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.

[38] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, pages 437–448. Springer, 2005.

[39] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[40] Thierry Mora, Aleksandra M Walczak, and Francesco Zamponi. Transition path sampling algorithm for discrete many-body systems. *Physical Review E*, 85(3):036710, 2012.

[41] Himanshu Paliwal and Michael R Shirts. A benchmark test set for alchemical free energy transformations and its use to quantify error in common free energy methods. *Journal of Chemical Theory and Computation*, 7(12):4115–4134, 2011.

[42] Sarah E Boyce, David L Mobley, Gabriel J Rocklin, Alan P Graves, Ken A Dill, and Brian K Shoichet. Predicting ligand binding affinity with alchemical free energy methods in a polar model binding site. *Journal of molecular biology*, 394(4):747–763, 2009.

[43] Gareth O Roberts and Jeffrey S Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71, 2004.